

Université de Montréal

**La Représentation Algorithmique
Des Dômes Mouqarnas :
Du Planaire Au Volumique**

par

Anis Semlali

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

Décembre 2001

© ANIS SEMLALI, 2001



NA
9000
U54
2002
V.005

Université de Montréal
Faculté des Études supérieures

Ce mémoire intitulé :

**La Représentation Algorithmique
Des Dômes Mouqarnas :
Du Planaire Au Volumique**

présenté par :

Anis Semlali

A été évalué par un jury composé des personnes suivantes :

Prof. Tidafi Temy

président du jury

Prof. Parisel Claude

directeur de recherche

Prof. Guité Manon

Codirectrice de recherche

Prof. Lachapelle Jacques

Membre du Jury

Mémoire accepté le : ..21 Mars 2002.....

Sommaire

L'objectif de la présente recherche est d'établir une méthode cohérente et structurée permettant de générer, à partir d'un tracé régulateur des mouqarnas modulaires en bois, le modèle tridimensionnel correspondant.

Pour ce faire, nous avons commencé par étudier l'origine et les différents styles et fonctions du modèle architectural objet d'étude. Cette partie nous a permis de constater l'aspect géométrique rigoureux qui caractérise l'architecture arabo – musulmane et spécialement les mouqarnas.

Ensuite nous avons procédé à une analyse géométrique des différentes pièces qui constituent l'alphabet des mouqarnas modulaires en bois. Cette partie nous a permis de dégager les différentes règles géométriques qui régissent la composition des tracés régulateurs mouqarnas et des éléments qui les constituent.

Ces règles géométriques nous ont permis de développer le volet algorithme et stratégie de traitement dans lequel nous proposons une méthode de traitement permettant de générer, de deux façons différentes, à partir d'un tracé régulateur d'un dôme mouqarnas, le modèle tridimensionnel correspondant.

Mots-clés	Voûtes mouqarnas, Tracé régulateur, Géométrie, programmation fonctionnelle
-----------	--

Abstract

The object of this research is to establish a coherent and structured method making it possible to generate a three-dimensional model from the corresponding plane projection of a mouqarnas dome.

With this intention, we began by studying the origin and the various styles and functions of the muqarnas. This part enabled us to note the rigorous geometrical aspects which characterize the arabo-muslim architecture and especially the mouqarnas.

We then carried out a geometrical analysis of the various parts which constitute the modular alphabet of the wood mouqarnas. This part enabled us to note the various geometrical rules which govern the composition of the regulating layout mouqarnas and of the elements which constitute them.

These geometrical rules enabled us to develop the algorithm and the processing strategy. In this part, we propose a method of processing allowing to generate, in two different ways, from a plane projection of a mouqarnas dome, the corresponding three-dimensional model.

Key words:	Muqarnas Vaults, Traced regulating, Geometry, Functional programming
------------	--

TABLE DES MATIÈRES :

1.	CHAPITRE 1 : INTRODUCTION	1
1.1	HYPOTHÈSE GÉNÉRALE	1
1.2	QUESTIONS DE LA RECHERCHE	2
1.3	OBJECTIF DE LA RECHERCHE	2
1.4	LIMITES DE LA RECHERCHE	2
1.5	MÉTHODOLOGIE ENVISAGÉE	3
2	CHAPITRE 2 : HISTOIRE ET ÉVOLUTION DES MOUQARNAS	4
2.1	INTRODUCTION	4
2.2	L'INTERMÉDIAIRE DE LA GÉOMÉTRIE	5
2.3	LES TROIS GENRES DE L'ART ISLAMIQUE LIÉS À L'ARCHITECTURE.....	6
2.3.1	<i>Le décor calligraphique.....</i>	<i>6</i>
2.3.2	<i>Le décor floral.....</i>	<i>7</i>
2.3.3	<i>Le décor géométrique</i>	<i>8</i>
2.4	LES MOUQARNAS	9
2.4.1	<i>Étymologie du mot</i>	<i>9</i>
2.4.2	<i>Définition.....</i>	<i>10</i>
2.4.3	<i>Mouqarnas et espace atomique</i>	<i>12</i>
2.4.4	<i>Les différents styles de mouqarnas.....</i>	<i>13</i>
2.4.5	<i>Les mouqarnas au Maroc</i>	<i>18</i>
2.4.6	<i>Quelques applications.....</i>	<i>18</i>
3	CHAPITRE3 : GÉOMÉTRIE ET RÈGLES DE COMPOSITIONS DES MOUQARNAS	22
3.1	INTRODUCTION	22
3.2	DÉFINITION GÉOMÉTRIQUE GÉNÉRALE DES MOUQARNAS PAR AL-KASHI	23
3.3	LA GÉOMÉTRIE DES PIÈCES.....	25

3.3.1	<i>Le module mouqarnas</i>	25
3.3.2	<i>Les pièces à base de triangle isocèle</i>	26
3.3.3	<i>Les pièces à base de losange</i>	29
3.3.4	<i>Les pièces à base de rectangle</i>	30
3.4	LE PLAN ET LE PAVAGE « CARRÉS-LOSANGES ».....	31
3.5	LES RÈGLES D'ASSEMBLAGE DES MOUQARNAS.....	31
4	CHAPITRE 4 : ÉTAT DES CONNAISSANCES	38
4.1	ARTICLE1: « MUQARNAS VISUALISATION AT THE IWR ».....	38
4.1.1	<i>Conclusion</i>	39
4.2	ARTICLE 2: "MOUQARNAS : SECRETS DES ORIENTAUX, UN COMPOSANT INNOVATEUR AVEC RYTHMIQUES ET LUMIÈRE ».....	40
4.2.1	<i>Conclusion</i>	40
5	CHAPITRE 5 : LA PARTIE INFORMATIQUE	41
5.1	OBJECTIF.....	41
5.2	CHOIX DE L'OUTIL.....	41
5.3	STRATÉGIE GÉNÉRALE DE TRAITEMENT :.....	41
5.3.1	<i>Analyse des données</i> :.....	42
5.3.2	<i>Les points qui codent les pièces</i> :.....	42
5.4	STRATÉGIE DE TRAITEMENT :.....	44
5.4.1	<i>Étape 1 : la représentation du tracé</i>	46
5.4.2	<i>Étape 2 saisie des données</i>	49
5.4.3	<i>Étape 3 : La reconnaissance des formes</i>	49
5.4.4	<i>Étape 4 : La construction des pièces</i> :.....	58
5.4.5	<i>Le traitement du tracé : Le système de couches</i>	65
6	CHAPITRE 6 : EXPÉRIMENTATION ET DISCUSSION	69
6.1	PRÉSENTATION DES MÉTHODES :.....	69
6.1.1	<i>Méthode de traitement globale des tracés</i> :.....	69

6.1.2	<i>Méthode de traitement couche par couche des tracés :</i>	69
6.2	LA MÉTHODE DE TRAITEMENT GLOBALE	70
6.2.1	<i>1^{er} cas d'étude :</i>	70
6.2.2	<i>2^{ème} cas d'étude :</i>	75
6.3	LA MÉTHODE DE TRAITEMENT COUCHE PAR COUCHE :	78
6.3.1	<i>Le traitement préliminaire :</i>	78
6.3.2	<i>Principe du traitement couche par couche :</i>	79
6.3.3	<i>Conclusion du chapitre :</i>	86
7	CHAPITRE 7 : CONCLUSION GÉNÉRALE	89
7.1	AUTRES APPLICATIONS POSSIBLES :	90
7.2	TRAVAIL À COMPLÉTER:	90
7.2.1	<i>Généralisation du traitement :</i>	90
7.2.2	<i>Étude des tracés à base de carré-losange :</i>	90
	BIBLIOGRAPHIE	92

LISTE DES FIGURES :

Figure 1	<i>Dôme de Alhambra de Grenade en Espagne.....</i>	2
Figure 2	<i>Modèles de calligraphies coufiques relevées sur divers monument du Caire</i>	7
Figure 3	<i>Gebs extérieur au palais « la Bahia » à Marrakech.....</i>	8
Figure 4	<i>Étoile à 16 de « Alhambra » en « gebs » inscrite dans un carré aux écoinçons floraux</i>	9
Figure 5	<i>Exemple de coupole mouqarnas revêtue de miroirs en Syrie mais de style iranien.....</i>	13
Figure 6	<i>Série de dessins tirés d'un manuel iranien montrant la manière de procéder, par niveaux successifs, pour bâtir une demi-coupole à mouqarnas en Iran.....</i>	14
Figure 7	<i>Tracé pour la taille de pierre dessiné par Michel Ecochard.....</i>	15
Figure 8	<i>Coupole en stuc.....</i>	16
Figure 9	<i>Alphabet des pièces mouqarnas les plus utilisées avec leur représentation 2d</i>	17
Figure 10	<i>Frise à Marrakech, Palmeraie Golf Palace.....</i>	19
Figure 11	<i>Arcade en Gebs, Marrakech, medersa Ben Youssef</i>	20
Figure 12	<i>Détail d'une alcôve du Palais Royal de Mekhnès.....</i>	20
Figure 13	<i>Une des deux impressionnantes coupoles à mouqarnas en bois au palais « Dar Essalem »</i>	21
Figure 14	<i>Photos de Al-Kashi.....</i>	23
Figure 15	<i>Code de représentation 2d de la pièce Dembouq.</i>	26
Figure 16	<i>Photos du modèle 3d de la pièce : Dembouq en bois de cèdre.....</i>	26
Figure 17	<i>Code de représentation 2d de la pièce Serwaliya.....</i>	26
Figure 18	<i>Photos de la pièce Serwaliya réalisé en bois de cèdre</i>	26
Figure 19	<i>Code de la représentation 2d de la pièce « Serwaliya Sguira »</i>	27
Figure 20	<i>Photos de la pièce « Serwaliya Sguira » en bois de cèdre</i>	27
Figure 21	<i>Code de la représentation 2d de la pièce « Loza1 ».....</i>	27
Figure 22	<i>Modèle 3d de la pièce « loza1 ».....</i>	27
Figure 23	<i>Code de représentation 2d de la pièce « Loza2 ».....</i>	28
Figure 24	<i>Modèle 3d de la pièce « Loza2 ».....</i>	28
Figure 25	<i>Exemple de combinaison de pièces.....</i>	28

Figure 26	Code de représentation 2d de la pièce « Ktaf ».....	29
Figure 27	Photos de la pièce « Ktaf » réalisé en bois de cèdre	29
Figure 28	Code de représentation 2d de la pièce « Charbiya ».....	29
Figure 29	Photos de la pièce « Charbiya » réalisé en bois de cèdre	29
Figure 30	Code de représentation 2d de la pièce « T'stiya Masdouda ».....	30
Figure 31	Photos de la pièce « T'stiya Masdouda » en bois de cèdre	30
Figure 32	Code de représentation 2d de la pièce « T'stiya masdouda ».....	30
Figure 33	Photos de la pièce « T'stiya Maftouha » en bois de cèdre	30
Figure 34	Toute composition mouqarnas peut être ramenée à un pavage carrés-losanges.	31
Figure 35	Orientation des arrêtes.	32
Figure 36	Exemples d'assemblage des pièces.....	33
Figure 37	Construction d'une structure mouqarnas à partir d'un pavage carré-losange quelconque.	33
Figure 38	Exemple d'orientation sur des pièces réelles.....	33
Figure 39	La pièce « rhombus » en 2d et en 3d.....	38
Figure 40	Modèle 3d de la voûte de « Takht-i Süleyman »	39
Figure 41	Modèle 3d de la voûte étudiée et transférée en VRML	40
Figure 42	Logique générale de traitement algorithmique d'un tracé régulateur mouqarnas.....	45
Figure 43	Tracé régulateur original d'une voûte mouqarnas en bois utilisé encore par les artisans au Maroc.	47
Figure 44	Organisation de la liste des entités sélectionnées.....	48
Figure 45	Saisie des données.....	49
Figure 46	Reconnaissance de la forme générale.....	50
Figure 47	Traitement des triangles	55
Figure 48	Traitement des quadrilatères	58
Figure 49	Stratégie des pièces simples : Exemple de la pièce « Dembouq »....	62
Figure 50	Stratégie des pièces semi-complexes : Exemple de la pièce « Ktaf »	63
Figure 51	Stratégie des pièces complexes : Exemple de la pièce « Charbiya».	64
Figure 52	Tracé original	70
Figure 53	Représentation informatique du tracé.....	70
Figure 54	Modèle numérique de la voûte.....	74
Figure 55	Tracé régulateur mouqarnas et sa représentation à l'écran.....	75

<i>Figure 56</i>	<i>Modèle numérique de la voûte</i>	77
<i>Figure 57</i>	<i>Première couche</i>	80
<i>Figure 58</i>	<i>Deuxième couche</i>	80
<i>Figure 59</i>	<i>Troisième couche</i>	81
<i>Figure 60</i>	<i>Quatrième couche</i>	81
<i>Figure 61</i>	<i>Cinquième couche</i>	82
<i>Figure 62</i>	<i>Sixième couche</i>	82
<i>Figure 63</i>	<i>Première rangée</i>	83
<i>Figure 64</i>	<i>Deuxième rangée</i>	83
<i>Figure 65</i>	<i>Troisième rangée</i>	84
<i>Figure 66</i>	<i>Quatrième rangée</i>	84
<i>Figure 67</i>	<i>Cinquième rangée</i>	85
<i>Figure 68</i>	<i>Sixième rangée</i>	85
<i>Figure 69</i>	<i>Septième rangée</i>	86

À mes parents ...

À mon frère Faouzi et sa femme Nadia ...

À mes frères et sœurs Mourad, Abir, Douaa et Ahmed ...

À Sniwty-el-galia, ma fiancée et future femme...

Remerciements

Mes remerciements vont tout d'abord à vous monsieur Claude Parisel. J'ai eu la chance de vous avoir comme directeur de recherche et vous avez su, avec votre grande expérience, faire ressortir ce qu'il y a de mieux en moi grâce à votre simplicité, votre encouragement, votre compréhension, votre objectivité et votre grand sens pédagogique.

Je voudrais vous remercier aussi monsieur Temy Tidafi pour votre aide, votre compréhension et votre amitié...Votre tempérament agréable et ouvert a permis de maintenir l'ambiance chaleureuse et fraternelle au sein du groupe.

Mes remerciements vont aussi à madame Manon Guité, ma codirectrice de recherche et professeure agrégée à l'Université de Montréal pour tout l'aide et le support qu'elle n'a cessé de m'apporter.

Je remercie également monsieur Giovanni de Paoli, responsable de la maîtrise CMFAO et professeur agrégé à l'Université de Montréal pour son support et ses conseils.

Je remercie mes très chers amis Leila Mankai et Denis Gamache. Je suis très fier de vous avoir comme amis...

Mes remerciements vont aussi à tous mes amis membres du GRCAO, leur compagnie et leur amitié m'a beaucoup aidé à m'intégrer au sein du groupe.

Je voudrais remercier mon très cher ami Ramez Bennour pour son amitié et son soutien moral.

Finalement je remercie madame Nicole Larivière : Sa gentillesse et sa bienveillance m'ont beaucoup aidé à me familiariser à un système qui m'étais étranger.

1. Chapitre 1 : Introduction

Les tracés régulateurs, en architecture, représentent un support de figuration graphique des éléments architecturaux codifiés sous forme de signes et de symboles. Ces signes et symboles permettent d'une part la lecture de ces tracés et d'autre part une articulation entre le système de représentation de l'objet architectural et son modèle tridimensionnel.

Mais cette représentation bidimensionnelle nous donne seulement une vue de l'objet architectural en faisant abstraction de ses informations tridimensionnelles qui sont nécessaires pour sa compréhension.

Dans le cas des mouqarnas (voir définition page 10), les tracés régulateurs qui les représentent ne peuvent contenir que des informations limitées à l'organisation des pièces les unes par rapport aux autres. Les transformer en une composition tridimensionnelle est un procédé d'interprétation dans lequel l'interpréteur a besoin de compléter les informations manquantes, tâche qui n'est pas toujours évidente à cause du niveau mathématique que leur analyse demande.

Aujourd'hui, grâce au développement des outils et langages informatiques, nous constatons qu'il est possible de décrire informatiquement les lois géométriques et les règles qui régissent ces tracés afin d'offrir de nouvelles possibilités de lecture, d'interprétation et de représentation de ces tracés régulateurs et de leurs modèles tridimensionnels correspondants.

1.1 Hypothèse générale

Les relations logiques et géométriques entre le tracé régulateur des mouqarnas modulaires en bois et le modèle tridimensionnel qu'il représente peuvent conduire à la description d'un processus de traitement informatique qui permet le passage de la représentation plane vers le modèle tridimensionnel.

1.2 Questions de la recherche

- Quelle stratégie adopter pour représenter, par le biais de moyens informatiques, les lois géométriques qui régissent les tracés régulateurs afin d'identifier les éléments architecturaux qu'ils représentent ?
- Quelle stratégie adopter pour générer le modèle tridimensionnel d'une voûte mouqarnas à partir de son tracé régulateur ?

1.3 Objectif de la recherche

Le but de la recherche est d'établir une méthode cohérente et structurée permettant de générer, à partir du tracé régulateur mouqarnas, le modèle tridimensionnel correspondant.

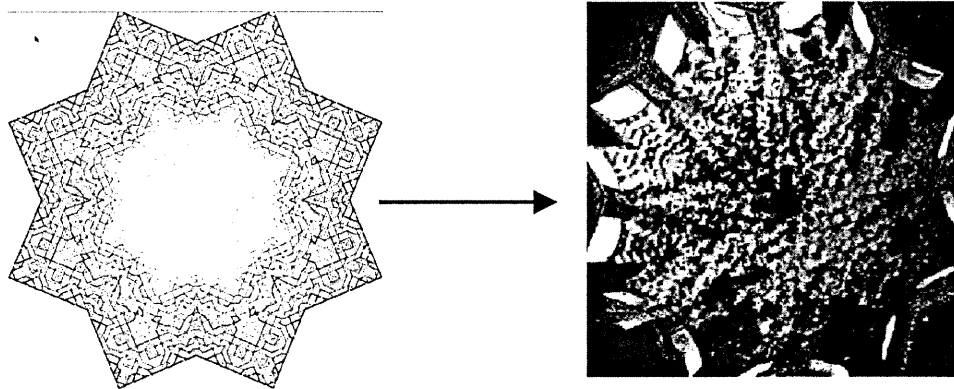


Figure 1 Dôme de Alhambra de Grenade en Espagne

1.4 Limites de la recherche

Cette recherche ne se veut pas une étude de tous les tracés régulateurs et de leurs relations avec leurs modèles tridimensionnels correspondants, mais elle sera limitée au seul cas des dômes mouqarnas au Maroc (structure modulaire en bois).

1.5 Méthodologie envisagée

La méthodologie retenue se fera en quatre étapes :

Origine et évolution des mouqarnas

Dans cette partie, nous nous proposons de traiter la question de l'origine et des différents styles et fonctions des mouqarnas.

Analyse géométrique

Cette partie traitera de l'analyse des formes géométriques de base des tracés régulateurs mouqarnas et des éléments qui les constituent.

La partie algorithmique et stratégie de traitement

Cette partie traitera de la description d'un processus de traitement qui permettra la réalisation de l'objectif de la recherche. Elle conduira à l'étape de traduction informatique de cet algorithme. Cette même partie comprendra 3 sous-parties :

- Le choix de l'outil informatique
- La partie algorithme
- La partie développement informatique

La partie expérimentation et discussion

Cette partie consistera à expérimenter la méthode proposée sur deux types de tracés régulateurs mouqarnas et ceci de deux manières différentes.

2 Chapitre 2 : Histoire et évolution des mouqarnas

Dans cette partie, nous nous proposons de situer le modèle architectural que nous allons traiter dans la présente recherche d'un point de vue historique pour arriver à préciser parmi toutes ses formes possibles, le cas que nous allons traiter et qui représentera le système combinatoire dans les dômes mouqarnas en bois au Maroc.

2.1 Introduction

Pour les premiers conquérants arabes, le mot architecture ne pouvait pas signifier vraiment le type d'espace dans lequel ils vivaient : les nomades ne bâtissaient pas, sinon quelques abris sommaires. Cependant, la liturgie de l'Islam imposera la nécessité d'une architecture : si le musulman n'est pas tenu pour ses cinq prières de se rendre en un lieu particulier, il doit, s'il veut participer à la prière collective du vendredi, se rendre à la mosquée : lieu de prière mais aussi lieu où sont discutées les affaires courantes de la communauté.

C'est bien plus tard que les Arabes commencèrent à manifester le désir d'une architecture prestigieuse, à la mesure de celle qu'ils découvraient en pays conquis.

Première affirmation de cette volonté, le dôme du rocher à Jérusalem (achevé en 692) est de conception nettement byzantine, puis, de construction plus tardive (714), la mosquée des Omeyyades à Damas, également réalisée par des artisans byzantins, est déjà, par son plan, une architecture islamique.

Cette architecture est donc, à l'origine, liée au sacré. Il en est de même pour l'art, dont les lieux d'application privilégiés sont l'architecture, par son décor, et le livre. Les artistes, suivant le penchant naturel de l'islam pour les idées abstraites, respectant l'interdiction de représenter des figures, se révélèrent très vite d'étonnants créateurs d'art non figuratif.

L'art islamique trouvait au XI siècle son langage propre. Il devait alors se répandre dans tout le monde et s'y maintenir pendant des siècles, tout en développant à foison ses potentialités.

Jean-Marc Castéra¹ a essayé de préciser les caractères généraux de cet art :

- « Un art lié au sacré..... pour le retour vers l'unité.....au moyen de l'harmonie.
- En conséquence un art de l'environnement.
- Un art abstrait (c'est normal pour l'art géométrique).
- Un art sous-tendu par une géométrie plus ou moins évidente : c'est l'aspect de la loi.
- L'art islamique a horreur du vide, d'où une profusion délirante qui contribue beaucoup à son identité formelle.
- Un art collectif, surtout dans l'architecture : dans la réalisation bien sûr, nécessitant la contribution de nombreux corps de métier, mais aussi dans la conception : le concepteur puise dans la tradition, des éléments qu'il assemble à sa manière ou à partir desquels il invente des variations qui viendront enrichir le répertoire commun. Rarement un motif de «zellige », mosaïque de terre cuite émaillée, sera revendiqué par un artiste-artisan particulier ; généralement, il ne sera jamais signé.
- Non-séparativité des genres : un même motif pourra se trouver décliné sur tous les supports, dans toutes les disciplines. »

2.2 L'intermédiaire de la géométrie

La pensée musulmane fût, dès l'origine de l'islam, l'héritière directe du savoir antique, chose qui a amené à la traduction en arabe de nombreux ouvrages de Pythagore, Platon, Aristote et autres. Ces traductions organisées à grande échelle dès le VIII^{ème} siècle(Fondation de nombreuses bibliothèques largement ouvertes au public, telle *la maison du savoir (dar el ilm)*) organisèrent le développement de la géométrie et de la science des nombres.

¹ « Arabesques : Art Décoratif Au Maroc » page 16

Ainsi, l'interdiction de la figuration dans l'Islam a orienté l'ensemble de la pensée musulmane vers l'expression géométrique. Elle s'est peu à peu imposée comme art majeur de l'Islam, visiblement parce que les tracés, dans leurs formes infinies, reflètent le fondement de la croyance, l'indivisibilité de Dieu. La complexité des tracés exprime bien aussi l'idée de structure atomique de l'univers. (Jean-Marc Castéra, 1996)

« Dans l'évolution des styles qui se sont succédés, chacun progressant naturellement vers un épanouissement, la science des tracés a contribué à maintenir le décor par une structure rigoureuse. Sans gêner son développement, elle a aboli tout risque de décadence en remplaçant, dans les périodes où les formes se compliquent à l'extrême, les répétitions et le maniérisme par une extraordinaire transcendance des compositions géométriques. »

(Jean-Marc Castéra, 1996)

2.3 Les trois genres de l'art islamique liés à l'architecture

Sur le plan formel, l'ensemble des arts plastiques musulmans liés à l'architecture comprend trois genres : Le décor calligraphique, le décor floral et le décor géométrique.

2.3.1 Le décor calligraphique

À l'exception du très abstrait style « Coufi » géométrique, tout l'art calligraphique islamique s'inscrit par-dessus un enchevêtrement de volutes d'inspiration végétale appartenant à l'univers de forme de l'arabesque florale (figure 2).

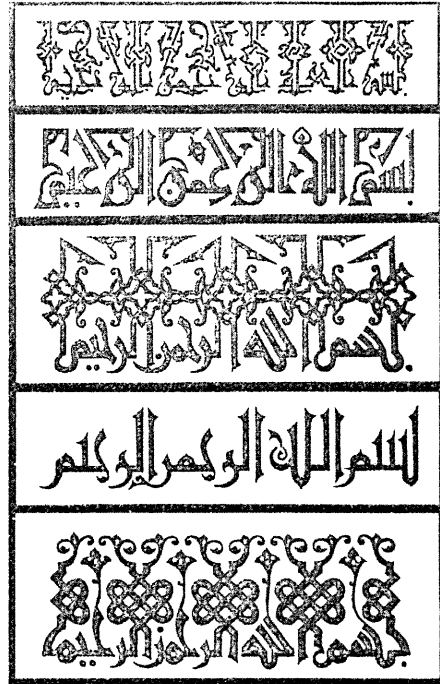


Figure 2 Modèles de calligraphies coufiques relevées sur divers monument du Caire

2.3.2 Le décor floral

En se développant à partir des modèles byzantins, l'univers décoratif floral islamique est devenu à la fois simple et complexe. Il s'est simplifié en stylisant les formes imitatives préexistantes, renouant ainsi avec les traditions artistiques primitives et il s'est enrichi en mettant en avant l'esprit de la géométrie.

L'arabesque florale peut se déployer dans toutes les dimensions du support. Mais c'est lorsqu'elle court en frises, plus ou moins larges le long des murs, sur les linteaux, ou au-dessus des panneaux de zelliges, souvent mêlée de calligraphies, qu'elle nous apparaît sous sa forme la plus originale.

Le décor floral convient parfaitement au remplissage des écoinçons des portes, ouvertures ou chemmasiats (de l'arabe « *chems* » : soleil désignant des petites fenêtres cintrées réalisées en stuc), car il peut s'adapter facilement à leurs formes irrégulières (figure 3).



Figure3 Gébs extérieur au palais « la Bahia » à Marrakech.

2.4.3 Le décor géométrique

Cet art a commencé à s'affirmer à partir du X^{ème} siècle en tant que composante majeure et indispensable de l'art musulman pour devenir prédominant plus tard en extrême Orient, au Maroc et en Andalousie. Il connaîtra un développement exceptionnel pour ensuite trouver son expression dans les réalisations les plus remarquables et les plus spectaculaires.

C'est à partir de compositions très simples, que le décor géométrique s'est peu à peu enrichi et compliqué jusqu'à devenir un art majeur, à forte cohérence interne.

Obsédé par l'idée d'unité, l'art musulman a, dans tous ses genres, accentué les rapports et les liens entre les formes. Or, si l'espace vide entre plusieurs lettres ou plusieurs éléments floraux qui se « donnent la main » prend difficilement la forme d'une lettre ou d'une feuille, le vide entre des polygones liés est aussi un polygone. C'est ainsi que disparaît la distinction fond/forme dans le décor géométrique. Il y a une parfaite réversibilité, seule la couleur permettra d'imposer au regard une interprétation plutôt qu'une autre.

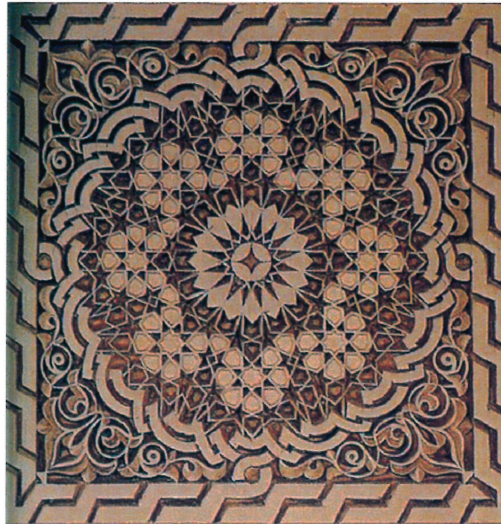


Figure4 Étoile à 16 de « Alhambra » en « gebs » inscrite dans un carré aux écoinçons floraux ²

Dans la présente étude, c'est le dernier aspect qui m'intéresse, dans le sens où il inclue à la fois le décor géométrique plan (les tracés régulateurs) et son prolongement tridimensionnel (les mouqarnas).

2.5 Les mouqarnas

2.5.1 Étymologie du mot

Les chercheurs n'ont pas réussi à trouver l'origine exacte du mot « mouqarnas », d'après Mohammed El-Assad, dans son article « The muqarnas : a géométric analysis », il existe diverses sources qui ont traitées ce sujet :

- La première prétend que le mot mouqarnas est emprunté au mot grec « Koronis ».
- Selon une autre, ce mot dérive d'un mot grec désignant les tuiles de toitures.

² Marrakech, restaurant « Narjis » au Palmeraie Golf Palace

- Si on revient aux racines, le mot mouqarnas dérive du mot arabe « qirnas » qui veut dire une épée en forme d'escalier, ou une falaise surplombante.

Une autre supposition est citée par Gulru Necipoglu dans son livre « The Topkapi scroll-géométry and ornament in Islamic architecture » où il affirme que le mot espagnol « mocarabe » serait à l'origine du mot mouqarnas en se basant sur sa prononciation (mouqarbas).

2.4.2 Définition

Le terme mouqarnas est la transcription d'un mot arabe le plus souvent traduit par « stalactites ». Il désigne des structures tridimensionnelles formées de certains éléments concaves, parfois fort nombreux, assemblés selon une géométrie complexe dans un but plus souvent ornemental que structurel.

Les mouqarnas sont utilisés dans les trompes d'angles ou dans les pendentifs des salles coupolées, dans la décoration des corniches. Encorbellement, chapiteaux et arcades, parfois en simples frises, toujours assez haut placés pour ne pas fatiguer le regard.

Leur application la plus spectaculaire concerne les plafonds et coupoles devant lesquels l'image des stalactites ne manque pas de s'imposer.

En fait, bien que témoignant toujours d'une évidente parenté formelle, ces structures seront bien différentes selon l'époque et le lieu :

- Par les matériaux de construction (briques, pierre, plâtre ou bois).
- Par le facteur d'échelle.
- Et par la conception même du tracé : stéréotomie précise pour les mouqarnas en pierre en Syrie ou en Égypte, assemblage de briques en Iran, système combinatoire opérant sur des modules préfabriqués au Maroc ou en Andalousie.

Apparues discrètement à l'aube du X^{ème} siècle, ces nouvelles formes évoluèrent très vite et se répandirent avec une étonnante rapidité jusqu'au Turkestan et en

Andalousie, tout comme les motifs géométriques plans dont elles constituent en quelque sorte le prolongement tridimensionnel.

Sans équivalent dans d'autres civilisations, les mouqarnas constituent par leur profonde originalité et leur application quasi systématique, une manifestation majeure de l'architecture islamique.

Entre le X^{ème} et le XI^{ème} siècle, les mouqarnas envahissent les lieux les plus chargés de sens de l'architecture : les « iwans » (portails monumentaux caractéristiques des mosquées de style iranien), les coupoles, et même les « mihrabs » (niche creusée dans la « qibla », le mur de la mosquée indiquant la direction de la Mecque) des mosquées. Aucune région du monde islamique n'échappe à cette mutation.

Comment expliquer un aussi fulgurant succès ? De quelle symbolique cette nouvelle forme est-elle porteuse pour justifier l'abandon à son profit des dômes lisses jusqu'à alors tant appréciés ?

Remarquons que ce mouvement est contemporain de l'extraordinaire développement du décor géométrique plan et de l'arabesque : en même temps que les murs se couvrent d'entrelacs sans fin, les coupoles se divisent en stalactites aux alvéoles souvent elles-mêmes décorées de motifs géométriques.

L'art tridimensionnel des arabesques et celui des arabesques géométriques planes participent manifestement du même esprit. Ses principales caractéristiques sont:

- Soumission des formes à la loi géométrique.
- Discontinuité de l'espace, représenté comme assemblage compact de formes élémentaires. C'est un espace atomisé, qui d'ailleurs n'existe pas sans matière, puisqu'il n'y a jamais de vides entre les formes.
- Continuités de la ligne, immatérielle, qui engendre ces formes discontinues composant le monde visible.

« La ligne est immatérielle, car elle n'est jamais que frontière entre deux pièces jointives. Et lorsqu'elle s'habille de matière, c'est par entrelacs, lesquels, témoignant d'une troisième dimension absente du plan où ils s'inscrivent (un « au-delà » de ce plan), sont clairement désignés comme l'image de la chose et non la chose elle-même. »

Jean-Marc Castéra, 1996 ³

Ainsi, la discontinuité concerne l'espace des formes, le monde des apparences. Mais ces formes ne sont pas pour autant isolées : elles sont interdépendantes, complémentaires, et peuvent être perçues comme la manifestation d'un dessin spirituel s'exprimant par le trait.

2.4.3 Mouqarnas et espace atomique

Qu'est-ce qui a poussé les musulmans, au début du XI^{ème} siècle, à abonder les coupoles demi-sphériques, lisses, avec toutes leurs valeurs structurelles, historiques et symboliques pour cette voûte fragmentée et conique ?

D'un point de vue historique, Yasser Tabbaa (1985), dans son article « The muqarnas Dome : its origin and meaning »⁴, remarque que l'origine des coupoles à stalactites, qu'il situe à Bagdad au début du XI^{ème} siècle, coïncide avec le triomphe des idées atomistes-occasionalistes de l'univers.

Selon ces théories, la matière est composée de particules indivisibles, les atomes, tenues dans une configuration donnée par la volonté de Dieu. Sans cette intervention, les formes n'auraient aucune cohésion durable, « l'univers ne serait que chaos ».

Les coupoles à mouqarnas, comme vision du monde (qui représente tout sauf Dieu), illustrent ces conceptions mieux que les voûtes lisses héritées de Byzance.

³ Arabesque : art décoratif au Maroc p :296

⁴ « Muqarnas » n :3 p :61

D'autant plus que les mouqarnas permettent de gommer les trompes, donnant au dôme cet air suspendu, sans assise terrestre, à l'image coranique du paradis.

Mathématique et architecture étaient simplement les outils choisis pour refléter et faire ressurgir ce concept théologique de la nature du monde et sa relation avec Dieu.

2.5.4 Les différents styles de mouqarnas

Dans cette partie, nous allons essayer de présenter sommairement les différents styles de mouqarnas qui existent ainsi que les méthodes de tracé qui lui sont associées de manière à mieux apprécier par la suite la particularité des techniques mises au point au Maroc pour les mouqarnas en bois et en stuc.

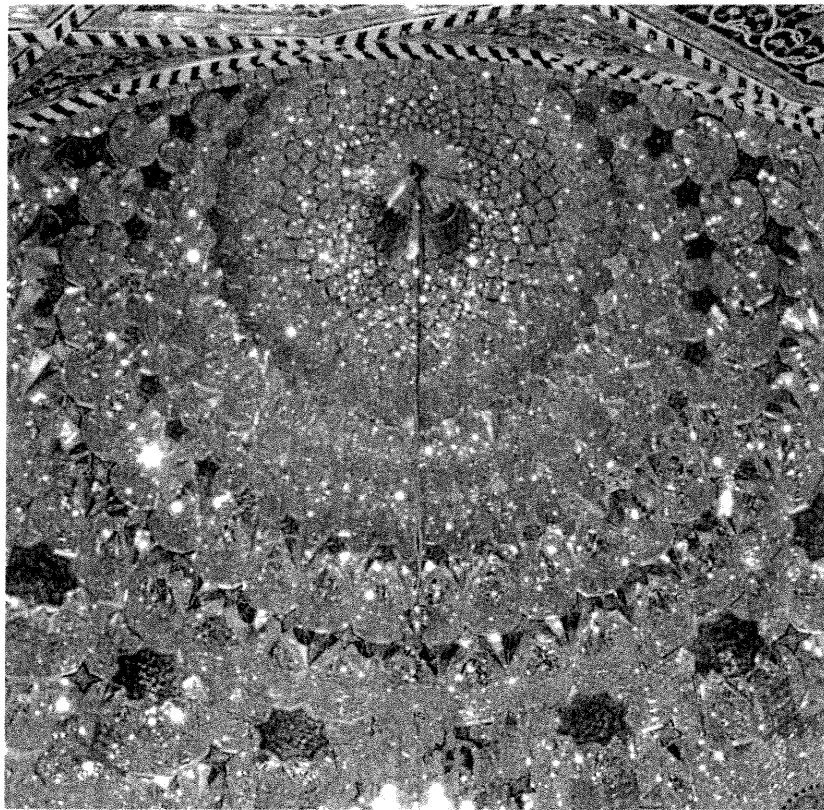


Figure5 Exemple de coupole mouqarnas revêtue de miroirs en Syrie mais de style iranien

2.5.4.1 La Brique

Cette série de dessins (Figure 6) tirés d'un manuel iranien montre la manière de procéder, par niveaux successifs, pour bâtir une demi-coupole à mouqarnas en Iran. Chaque niveau de mouqarnas sera construit en briques qu'il faudra ensuite sculpter pour adoucir les transitions.

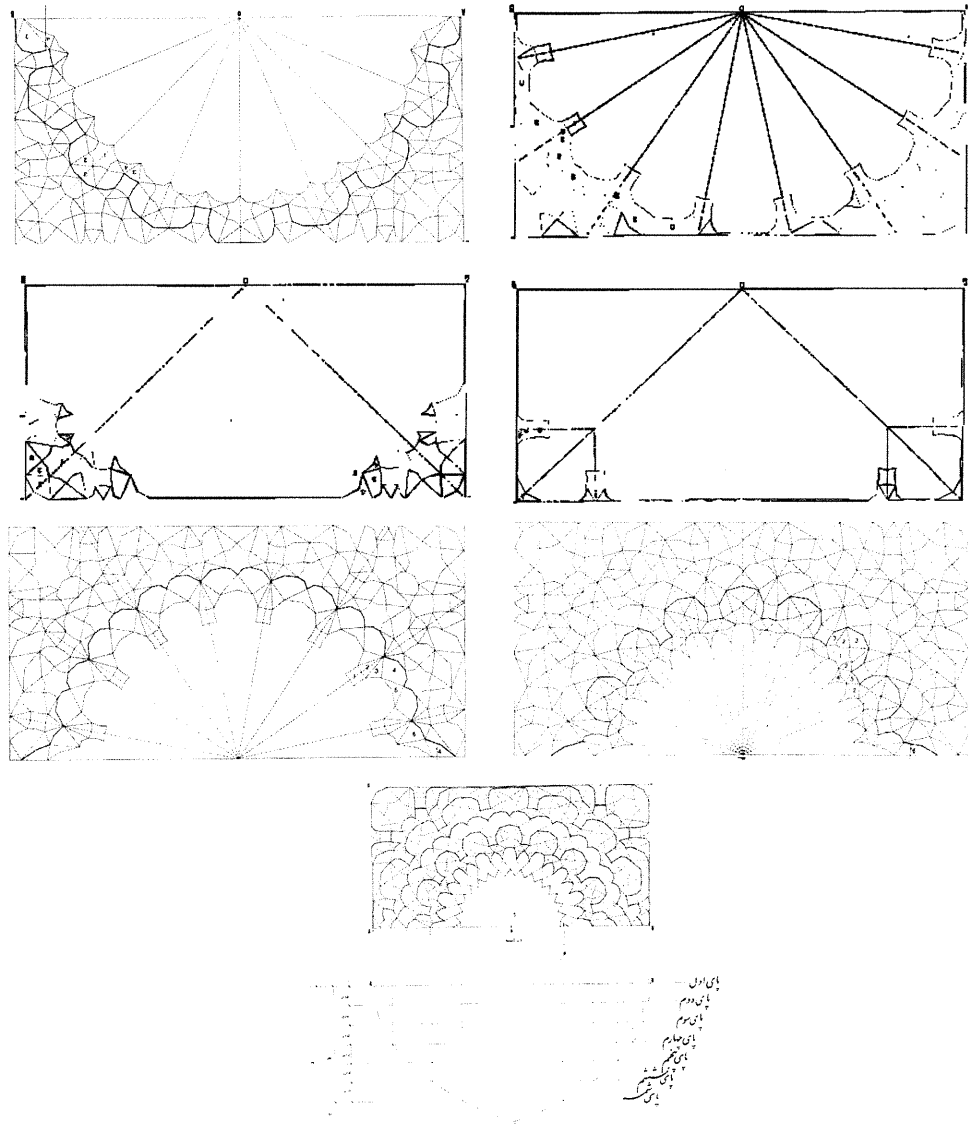


Figure6 Série de dessins tirés d'un manuel iranien montrant la manière de procéder, par niveaux successifs, pour bâtir une demi-coupole à mouqarnas en

Iran

2.5.4.2 La Pierre

Les mouqarnas de pierres reposent sur une stéréotomie précise et ils sont surtout utilisés en Syrie et en Égypte. (Figure 7)

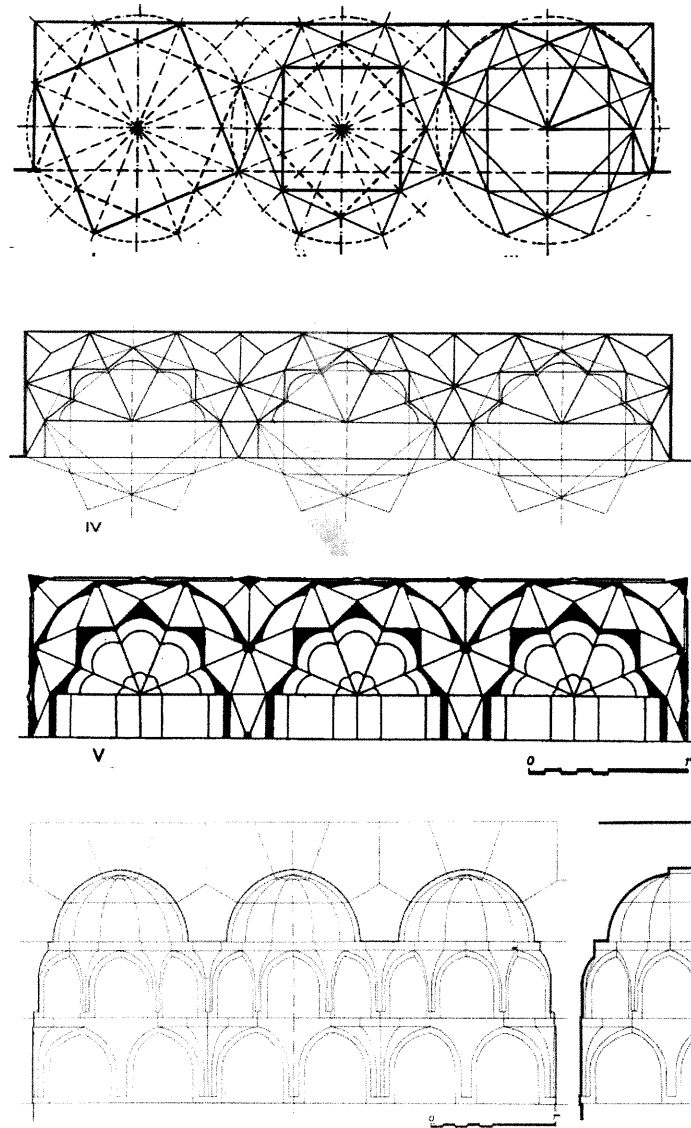


Figure7 Tracé pour la taille de pierre dessiné par Michel Ecochard⁵

⁵ D'après Ecochard, *Filiation des monuments Grecs, Byzantins et islamiques : une question de géométrie*. P. Geuthner, Paris 1978

2.5.4.3 Le Stuc

Le plâtrier commence son ouvrage par le bas, mettant en place les moules sur plusieurs rangés, puis il coule le plâtre. La prise étant faite, il réutilise les moules pour les rangées supérieures par étapes successives jusqu'à la « chéchia ».

Ces mouqarnas seront souvent ciselés et souvent recouverts de décor peint. Une autre technique, plus archaïque et peut être plus fréquente, consiste à utiliser des pièces de bois débitées selon la trame choisie. Ces éléments servent de support à un plâtre projeté manuellement et sculpté.

Leur tracé présente avec celui des mouqarnas en bois une différence capitale : il n'est pas assujéti à une trame quatre-vingt-dix/quarante-cinq degrés. En effet, les négatifs des pièces qui servent de moules peuvent être disposés entre eux selon des angles plus ouverts. Cette technique permet une adaptation plus serrée à la courbure sphérique, et une plus grande liberté d'exécution. (Figure 8)

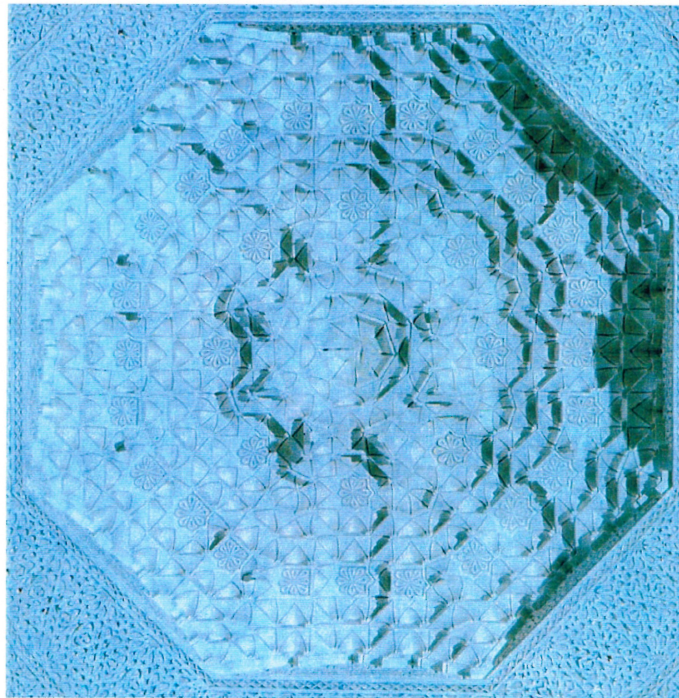


Figure8 Coupole en stuc

2.4.4.4 Le Bois

Cette technique repose sur un système combinatoire opérant sur des modules préfabriqués en bois qui ont été utilisés surtout au Maroc et en Andalousie. Ce système repose sur l'assemblage d'un petit nombre de pièces différentes (voir figure 9); même si l'ensemble d'une composition peut compter facilement des centaines ou des milliers de modules.

Ces modules sont façonnés dans des prismes de trois sortes : base de triangle isocèle, de losange, ou de rectangle. La grandeur des pièces est variable, ainsi que le détail de leurs formes et de leur décoration (face visible sculptée, ciselées ou peintes)

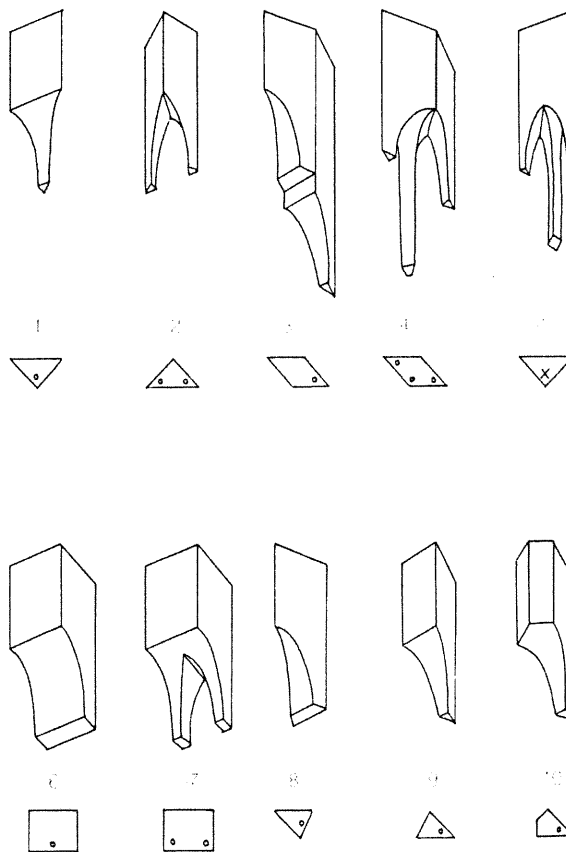


Figure 9 Alphabet des pièces mouqarnas les plus utilisées avec leur représentation 2d⁶

⁶ Arabesque : art décoratif au Maroc p : 316

2.4.5 Les mouqarnas au Maroc

Héritant des acquis de la brillante civilisation andalouse et de ses artisans après la chute de Grenade en 1492, les « maalems », maîtres artisans du Maroc, vont conserver ce savoir-faire et le faire évoluer dans un cadre traditionnel d'ateliers familiaux où la transmission des connaissances se fait de père en fils, de maître en disciple.

Les coupôles à stalactites de la mosquée « Quaraouiyine » à Fès illustrent combien la technique des mouqarnas était avancée dès le XII^{ème} siècle. Y sont en jeu les formes et les principes organisationnels que nous retrouverons plusieurs siècles plus tard, portés à la perfection et déployés dans de plus grandes dimensions au palais d'Alhambra de Grenade. Ce sont quasiment les mêmes qui se manifestent aujourd'hui encore dans les réalisations de prestige au Maroc.

L'aspect modulaire y est assez clairement établi : dans la coupole du mihrab, ce sont bien les mêmes formes élémentaires que l'on retrouve à tous les niveaux (dans les autres coupôles, il y existe quelques changements d'échelle) .

Cet aspect ira se renforcer jusqu'à être pleinement réalisé par la technique des mouqarnas en bois. Ceux-ci composeront de grands puzzles tridimensionnels par la nombreuse répétition de quelques pièces différentes se prêtant à d'innombrables combinaisons savantes.

2.4.6 Quelques applications

2.4.6.1 *les frises*

Pour les « maalems », les chefs artisans au Maroc, appliquer les mouqarnas aux frises constitue un « jeu d'enfant » par rapport à la complexité de leurs applications dans les voûtes. Ces frises, plus ou moins complexes, sont utilisées en corniche, au-dessus des ouvertures et autour des plafonds et des coupôles. Les exemples les plus courants sont à un, deux ou trois niveaux (figure 10).

2.5.6.2 les encorbellements

Les mouqarnas sont utilisés quasi systématiquement pour l'habillage des corbeaux dont le rôle, parfois fonctionnel, est le plus souvent strictement décoratif. Ils peuvent aussi servir de chapiteau par répétition sur les quatre cotés d'une colonne (Figure 11)

2.5.6.3 les arcades:

Les arcades concernent bien entendu les ouvertures. Mais il existe aussi de fausses arcades, purement décoratives, simulées sur les murs par un décor de placage.(Figure 12)

Comme suspendues sans assise véritable, elles planent dans le décor, suggérant un mouvement d'ascendance lente et irréversible soutenue par la collectivité unie de petites pièces s'appuyant les unes sur les autres.

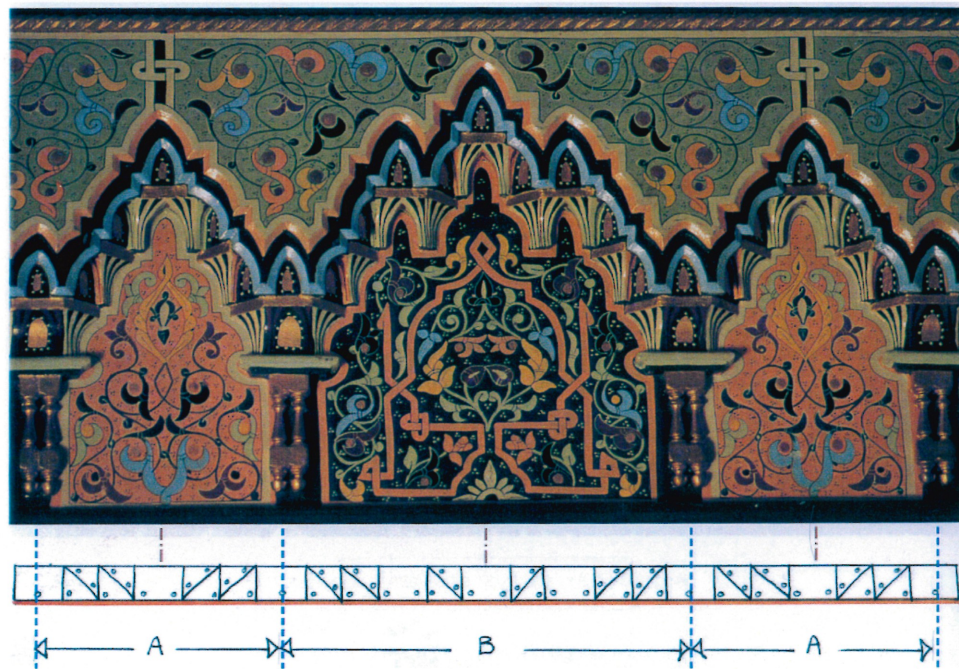


Figure10 Frise à Marrakech, Palmeraie Golf Palace

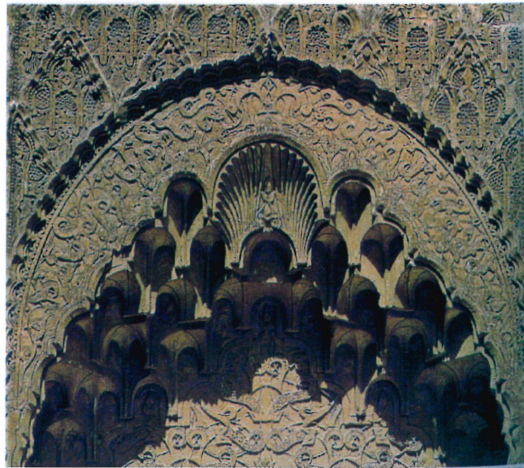


Figure 11 Arcade en Gebes,
Marrakech, medersa Ben Youssef

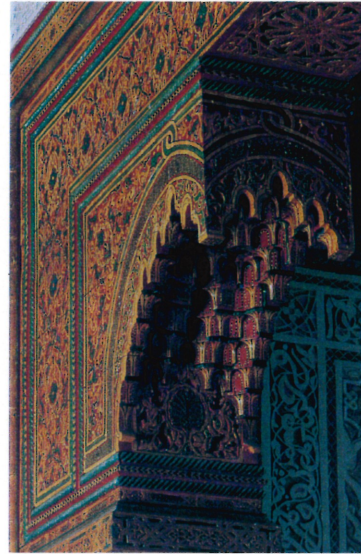


Figure 12 Détail d'une
alcôve du Palais Royal de
Mekhènes

2.5.6.4 *les coupoles*

Les plus spectaculaires mise en scène de mouqarnas se jouent aux théâtres suspendus des coupoles :

« Le spectateur se trouve à la fois submergé par une avalanche de géométrie complexe et entraîné dans un irrésistible mouvement d'élévation »

Jean-Marc Castéra

De base parfois carrée, le plus souvent octogonale, de dimension modeste ou orgueilleuse, les coupoles mouqarnas héritent toujours de la métaphore de la voûte céleste.

Ils peuvent s'adapter dans tous les volumes inversés : intrados d'arc, voûtes semi-cylindriques ou polyédrique, coupoles semi-sphériques ou pyramides tronquées. Ils compliquent encore plus ces espaces internes en y ajoutant des cavités et des reliefs, des ombres et des lumières.

C'est l'invention du « rakhwi », arc brisé de l'épaule, qui a permis l'adaptation du mouqarnas aux différentes courbes de coupôles, de voûtes et de plafonds, dans lesquels toutes les pièces du jeu de construction sont utilisées. L'alphabet de ces pièces permet une variété inépuisable de combinaisons et tout comme les autres tracés à plat, les « maalems » recherchent et inventent constamment de nouvelles configurations.

Les volumes à décorer ne sont jamais similaires et le tracé est, chaque fois, une aventure nouvelle.

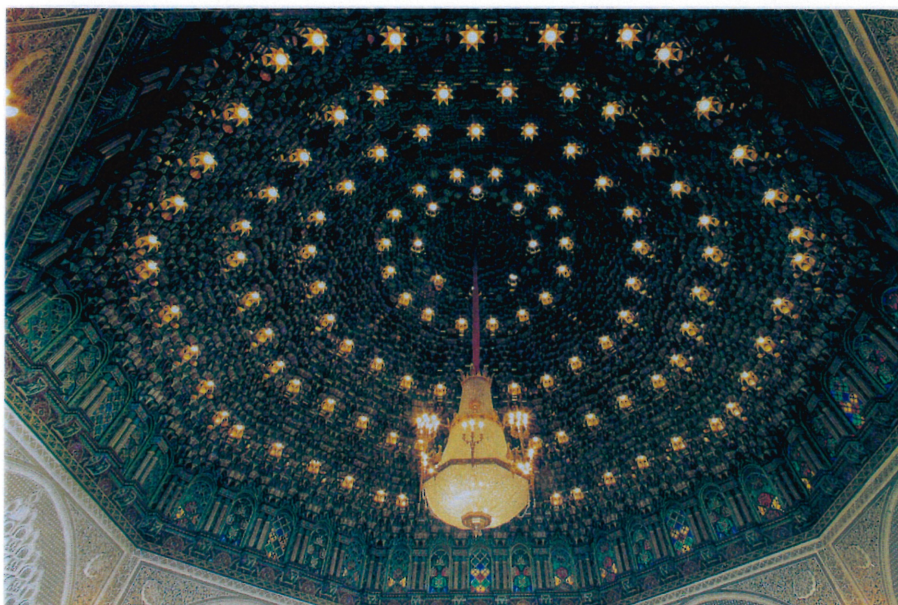


Figure13 Une des deux impressionnantes coupôles à mouqarnas en bois au palais « Dar Essalem »

3 Chapitre3 : Géométrie et règles de compositions des mouqarnas

Dans le chapitre précédent, nous avons montré que les dômes mouqarnas obéissent, comme toute l'architecture musulmane, à des lois géométriques précises. Dans ce chapitre nous analyserons les formes géométriques de base des tracés régulateurs mouqarnas et les éléments qui les constituent.

3.1 Introduction

Les dômes mouqarnas sont une curiosité qui a suscité l'intérêt de plusieurs historiens, archéologues et mathématiciens. Ils ont écrit plusieurs ouvrages sur leur historique ou leur symbolique mais très rarement sur leurs règles géométriques.

La géométrie des mouqarnas est la partie sur laquelle on trouve très peu d'informations. Le premier document à l'avoir traité est « Miftah-El-Hissab » écrit par le géomètre arabe Al-Kashi au quinzième siècle et qui nous apporte une analyse géométrique relativement détaillée par rapport aux autres ouvrages disponibles. Mais la majorité de son discours était dédié à la mesure de la surface des mouqarnas et son but était de présenter une méthode pour calculer la quantité de matériaux dont le maçon ou le constructeur aurait besoin pour la construction de ses voûtes mouqarnas.

Il existe aussi d'autres documents illustrant des tracés régulateurs de compositions mouqarnas (figure 43) qui ont été conservés et à partir desquels on peut en déduire un bon nombre d'informations utiles.

Une des caractéristiques les plus fascinantes des mouqarnas est la géométrie. Bien qu'analysé par des auteurs comme Al-Kashi, Jones et Goury, Mamoun Sakkal ou Notkin et Michel Ecochard (voir bibliographie), les mouqarnas restent un mystère pour la plus part des historiens en architecture, partiellement à cause du niveau mathématique que son analyse demande. Mais on peut par contre les étudier suivant différents niveaux mathématiques : la géométrie des mouqarnas n'était pas traitée seulement par des mathématiciens comme Al-Kashi (figure 14) mais

aussi conçue et construite par des artisans avec une éducation modeste qui consistait seulement en des connaissances de base dans la géométrie euclidienne.



Figure 14 Photos de Al-Kashi

3.2 Définition géométrique générale des mouqarnas par Al-Kashi

Dans l'article⁷ de Yvonne Dold-Samplonius, on trouve la définition que ce mathématicien a donnée des mouqarnas pendant le quinzième siècle:

“ The mouqarnas is a ceiling like a staircase with *facets* and a flat *roof*. Every facet intersects the adjacent one at either a right angle, or half a right angle, or their sum, or another combination of these two. The two facets can be thought of as standing on a plane parallel to the horizon. Above them is built either a flat surface, not parallel to the horizon, or two surfaces, either flat or curved, that constitute their roof. Both facets together with their roof are called one *cell*. Adjacent cells, which have their bases on one and the same surface parallel to the horizon, are called one *tier*.» In addition there are *intermediate elements* which connect the roofs of adjacent cells.”

Al-Kashi⁸

⁷ « Pratical Arabic mathematics : Measuring the Muqarnas by Al-Kashi »

⁸ « Pratical Arabic mathematics : Measuring the Muqarnas by Al-Kashi »

De la définition précédente, on retient les éléments suivants :

- Les mouqarnas forment un plafond comme un escalier avec des *facettes* et un *toit plat* (*la facette d'une cellule* : coté vertical; *Toit d'une cellule* : une surface plane, non parallèle à l'horizon ou 2 surfaces jointes qui peuvent être soit planes soit courbes).
- Chaque facette intersecte celle qui lui est adjacente à un angle droit, à la moitié d'un angle droit, à leur somme, ou une combinaison différente de ces deux angles.
- Les deux facettes peuvent être considérées comme se tenant sur un plan parallèle à l'horizon. Au-dessus d'elles est construit une surface plate, pas parallèle à l'horizon, ou deux surfaces, plates ou/est courbés, qui constituent leur toit.
- Les deux facettes ainsi que leur toit s'appellent une *cellule* (Chaque *cellule* consiste en 2 cotés, ou facettes, avec un toit).
- Des cellules adjacentes, qui ont leurs bases sur la même surface parallèle à l'horizon, s'appellent une *rangée* (*Rang* (*Tier*) : Une rangée de cellule avec leurs bases sur la même surface parallèle à l'horizon).

Al-Kashi distingue 4 types de mouqarnas :

- « Sadhijj » la simple ou la plane : C'est la plus simple des quatre et elle se caractérise par ses contours angulaires.
- « Mutayyan » : Elle est similaire à la précédente sauf que ses rangées n'ont pas toutes la même hauteur.
- « Quaws », l'arc: Elle est caractérisée par ses contours courbes.
- « Shirazi » : c'est le plus complexe des quatre. Les autres types sont représentés en plan exclusivement par des triangles et des quadrilatères. Un mouqarnas « shirazi » contient d'autres polygones comme les pentagones, les hexagones, octogones ou les étoiles multipointes.

3.3 La géométrie des pièces

Les mouqarnas en bois au Maroc résultent de l'assemblage d'un petit nombre de pièces différentes, appelés primitives; même si l'ensemble d'une composition peut compter facilement des centaines ou des milliers de modules.

Ces primitives sont façonnées dans des prismes de trois sortes :

- Base en triangle isocèle
- Base en losange
- Base en rectangle

3.3.1 Le module mouqarnas

Pour faire en sorte que les différentes primitives mouqarnas coïncident entre elles, il faut qu'elles soient construites à partir d'une même unité de mesure. Al Kashi utilise dans son calcul ce qu'il a appelé « miqyas » ou le module (M) des mouqarnas, défini comme base de la facette la plus large.

Ce module représente l'unité de base des mouqarnas et il est égal à 1. Cette procédure facilitera énormément les différents calculs des mouqarnas et à la fin, les résultats de ces calculs pourront être convertis en mesure standard.

En plan, la majorité des projections planes des primitives mouqarnas sont des triangles et des quadrilatères et dans certains cas, on peut trouver d'autres formes géométriques comme le floral, les étoiles ou des polygones...

3.4.2 Les pièces à base de triangle isocèle

3.4.2.1 *Dembouq*

C'est une pièce façonnée dans un prisme dont la base est un triangle rectangle isocèle. Le point qui code sa représentation 2d indique le pied de la pièce situé au sommet où l'angle 90° est formé. Cette pièce est considérée par les artisans comme étant la pièce maîtresse des mouqarnas. (Figures 15 et 16)

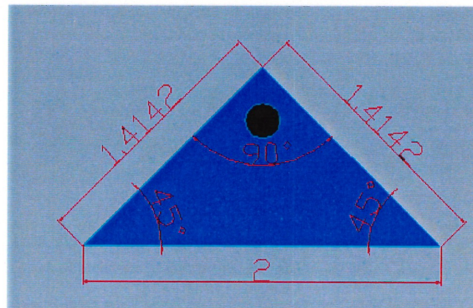


Figure15 Code de représentation 2d de la pièce Dembouq.

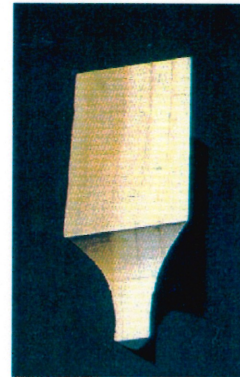


Figure16 Photos du modèle 3d de la pièce : Dembouq en bois de cèdre

3.4.2.2 *Serwaliya (pantalon) ou Bouja*

C'est une pièce façonnée dans un prisme dont la base est un triangle rectangle isocèle. Les 2 points qui codent sa représentation 2d indiquent les 2 pieds de la pièce situés aux 2 sommets où un angle de 45° degrés est formé. (Figures 17 et 18)

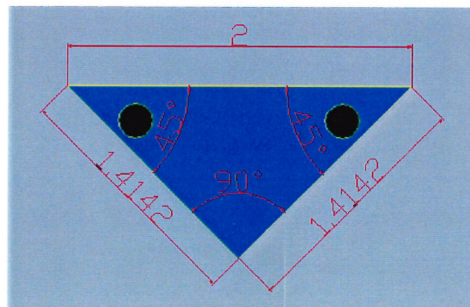


Figure17 Code de représentation 2d de la pièce Serwaliya

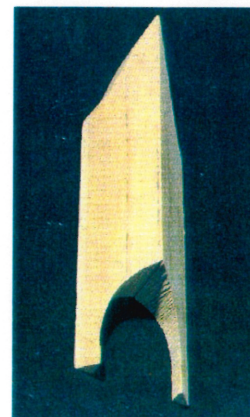


Figure18 Photos de la pièce Serwaliya réalisé en bois de cèdre

3.4.2.3 Chiira ou Serwaliya Sguira (petit pantalon)

Cette pièce est façonnée dans un prisme dont la base est un triangle isocèle rectangle. Le signe qui code sa représentation 2d indique le pied de la pièce situé au sommet où un angle de 90° est formé. (Figures 19 et 20)

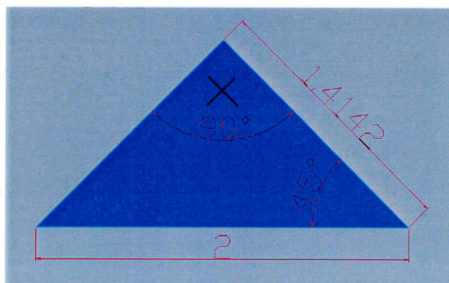


Figure19 Code de la représentation 2d de la pièce « Serwaliya Sguira »

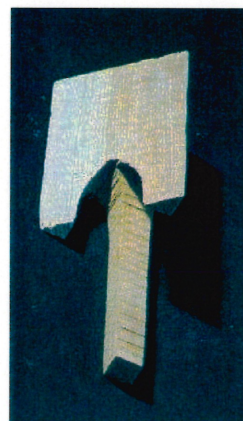


Figure20 Photos de la pièce « Serwaliya Sguira » en bois de cèdre

3.4.2.4 Loza 1

C'est une pièce façonnée dans un prisme dont la base est un triangle isocèle non rectangle. Le point qui code sa représentation 2d indique le pied situé au coté opposé au sommet où un angle de 45° est formé. (Figure 21 et 22)

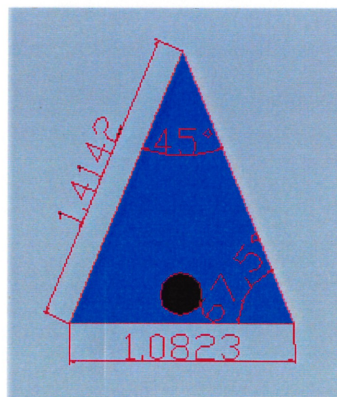


Figure21 Code de la représentation 2d de la pièce « Loza1 »



Figure22 Modèle 3d de la pièce « loza1 »

3.4.2.5 Loza 2

Cette pièce est façonnée dans un prisme dont la base est un triangle isocèle non rectangle. Le point qui code sa représentation de 2d indique le pied situé au sommet où un angle de 45° est formé. (Figure 23 et 24)

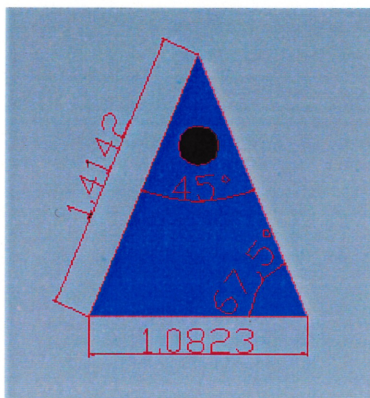


Figure23 Code de représentation 2d de la pièce « Loza 2 »

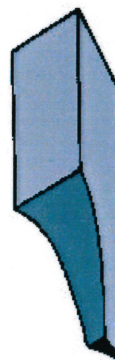


Figure24 Modèle 3d de la pièce « Loza2 »

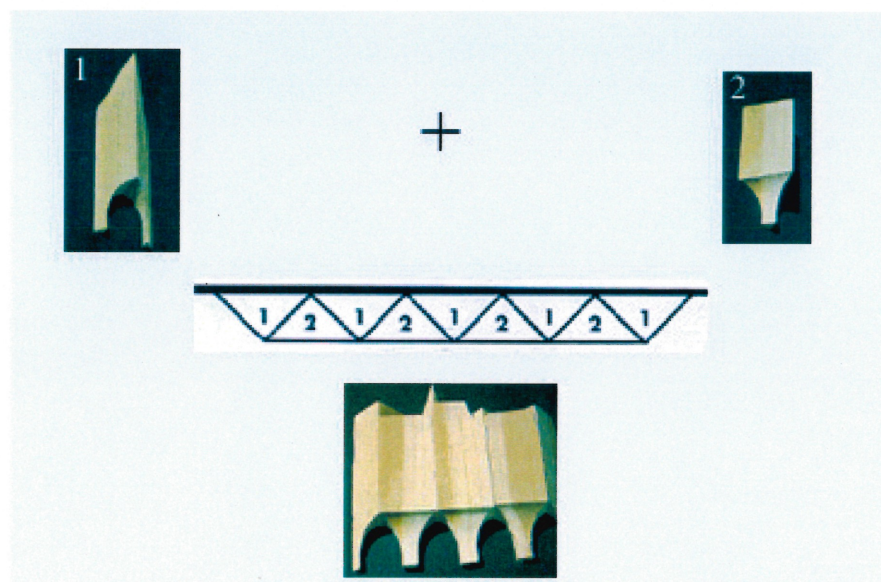


Figure25 Exemple de combinaison de pièces

3.4.3 Les pièces à base de losange

3.4.3.1 *Ktaf* (épaule)

Cette pièce est façonnée dans un prisme dont la base est un losange régulier. Le point qui codent cette pièce indique le pied situé au sommet où un angle de 45° est formé. (Figure 26 et 27)

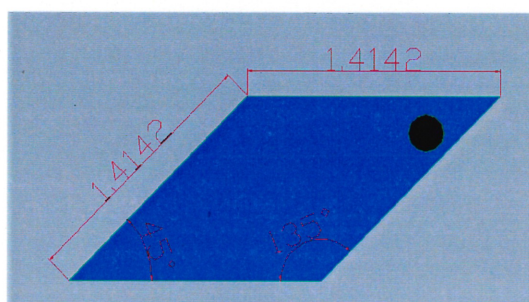


Figure26 Code de représentation
2d de la pièce « Ktaf »

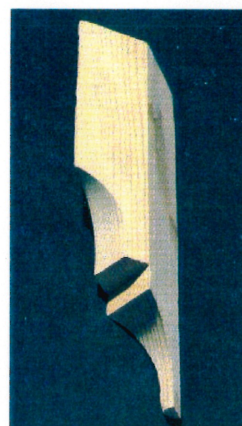


Figure27 Photos de la pièce
« Ktaf » réalisé en bois de cèdre

3.4.3.2 *Charbiya* ou *Chiira* (fin cheveux)

Cette pièce est façonnée dans un prisme dont la base est un losange régulier. Les points qui codent cette pièce indiquent les 3 pieds situés aux sommets où les angles 45° et 135° sont formés. (Figure 28 et 29)

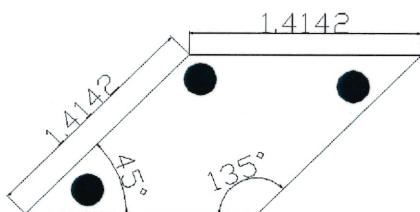


Figure28 Code de représentation
2d de la pièce « Charbiya ».

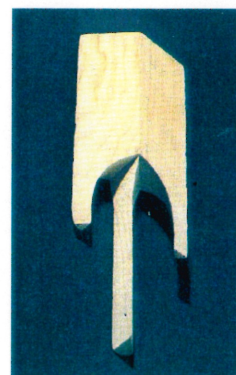


Figure29 Photos de la pièce
« Charbiya » réalisé en bois de cèdre

3.4.4 Les pièces à base de rectangle

3.4.4.1 *T'stiya Masdouda (motif fermé)*

Cette pièce est façonnée dans un prisme dont la base est un rectangle. Le point qui code cette pièce indique le pied situé à un de ses 2 plus grand cotés.

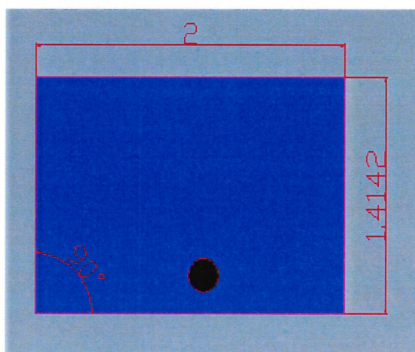


Figure30 Code de représentation
2d de la pièce « T'stiya Masdouda ».

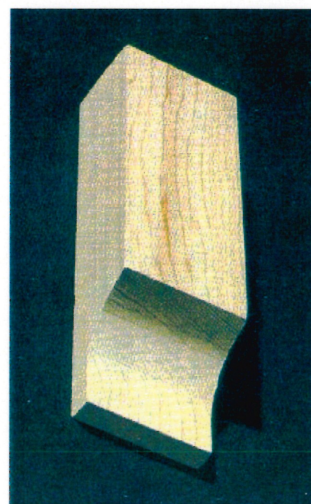


Figure31 Photos de la pièce
« T'stiya Masdouda » en bois de cèdre

3.4.4.2 *T'stiya Maftouha (motif ouvert)*

Cette pièce est façonnée dans un prisme dont la base est un rectangle. Les 2 points qui codent sa représentation 2d indiquent les sommets de un des 2 plus grands cotés.

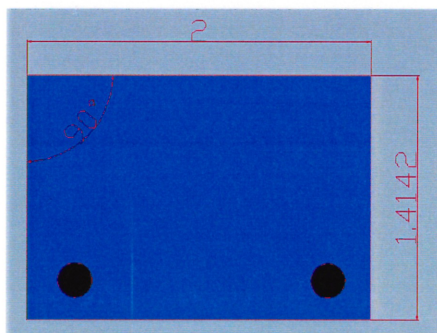


Figure32 Code de représentation
2d de la pièce « T'stiya masdouda ».

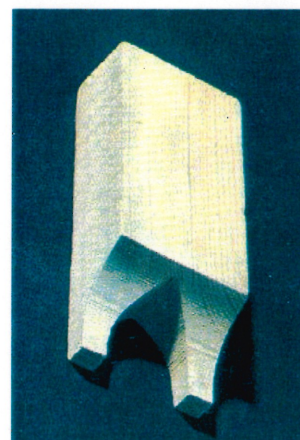


Figure33 Photos de la pièce
« T'stiya Maftouha » en bois de cèdre

3.5 Le plan et le pavage « carrés-losanges »

La découverte des pavages de Penrose (mathématicien et physicien anglais) date des années 1970. Ils étaient considérés comme des curiosités mathématiques avant que des physiciens ne découvrent, en 1984, que la nature ne les avait pas attendus pour s'en servir.

Dans son livre « Géométrie Douce », Jean-Marc Castéra a écrit que « l'art arabe était passé à coté de ces étonnantes structures », mais il a nuancé son opinion dans livre « Arabesque : Art Décoratif au Maroc » après avoir découvert ce même pavage carrés-losanges « En tant que structure sous-jacente à toute composition de mouqarnas modulaires : il n'y pas que des carrés et des losanges dans les plans de mouqarnas, mais les autres formes peuvent toujours se ramener à une décomposition de ces carrés et de ces losanges ».

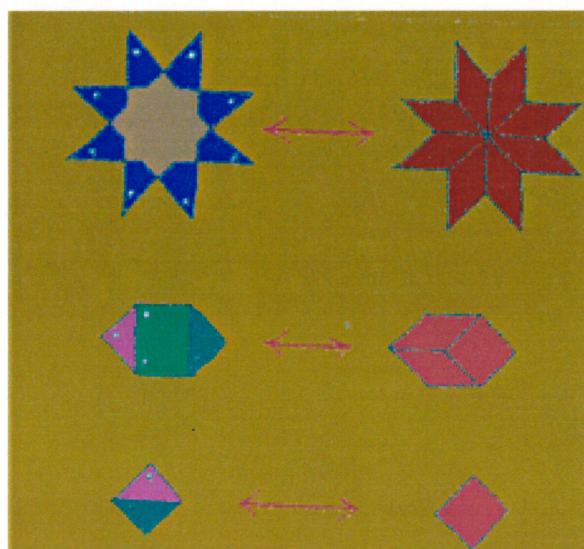


Figure34 Toute composition mouqarnas peut être ramenée à un pavage carrés-losanges.

3.6 Les règles d'assemblage des mouqarnas

Les différentes pièces qui composent l'alphabet des compositions mouqarnas ne sont pas assemblées n'importe comment, il faut assurer un principe de continuité car si les mouqarnas représentent l'idée d'un monde « atomiste », il s'agit aussi

d'un monde harmonieux dans lequel les formes s'organisent et s'assemblent avec amour.

3.6.1.1 Les règles d'orientation

Deux pièces voisines pourront « se faire du pied », « marcher l'une sur l'épaule de l'autre » ou plus rarement, s'il s'agit de pièces semblables, se mettre dos à dos.

Dans notre cas où on veut manipuler les différentes pièces ou primitives du mouqarnas, les règles d'assemblages s'imposeront rapidement d'elles même.

Cependant, il suffit d'orienter les pièces montantes pour assurer la cohérence de l'assemblage par une unique règle simple :

« Deux pièces voisines doivent avoir leurs arrêtes

jointives orientées dans le même sens »

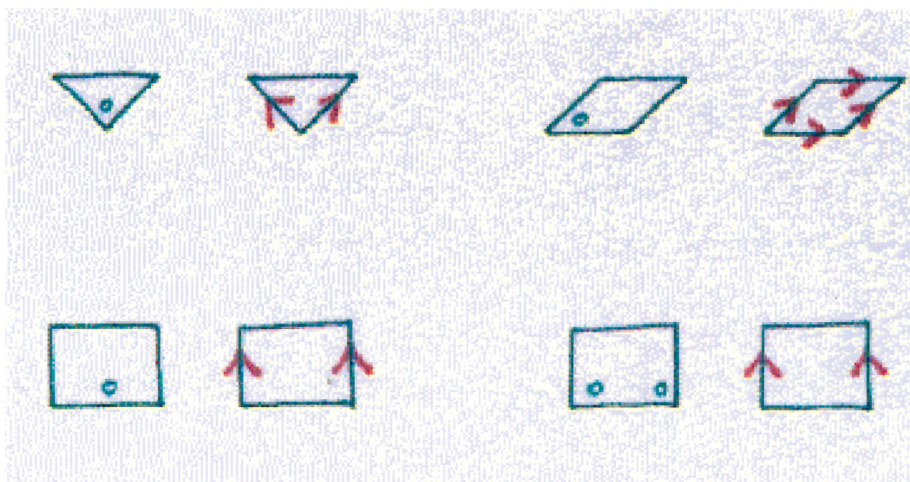


Figure35 Orientation des arrêtes.

Ainsi un plan cohérent de structure à mouqarnas correspond à un réseau orienté construit sur la trame du pavage carré-lozange.

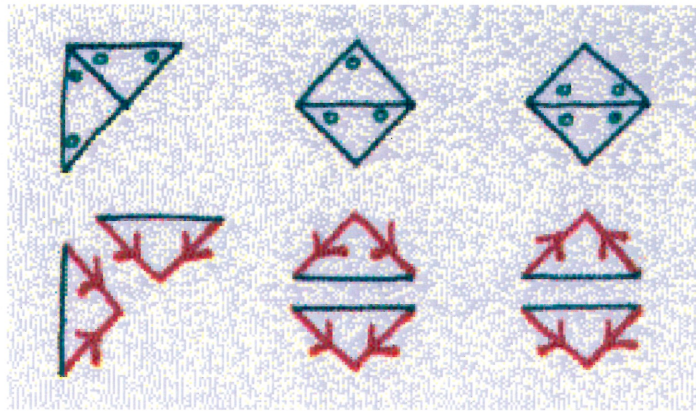


Figure36 Exemples d'assemblage des pièces

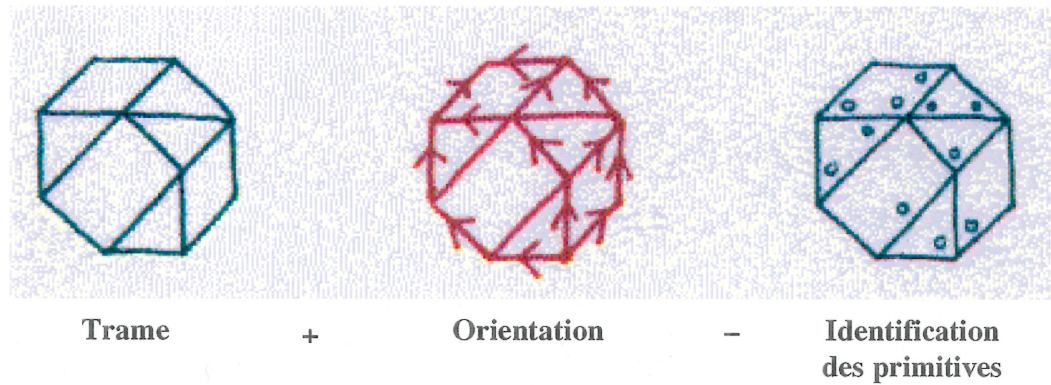


Figure37 Construction d'une structure mouqarnas à partir d'un pavage carré-losange quelconque.

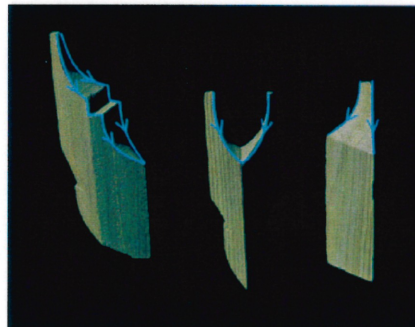


Figure38 Exemple d'orientation sur des pièces réelles

La règle de vérification des tracés mouqarnas

Dans son livre « Arabesques, Art décoratif au Maroc », Jean-Marc Castré a proposé un test d'usage général qui me sera d'une aide appréciable pour la présente recherche qui nous permettra de différencier les tracés « corrects » des tracés dans lesquels les artisans ou les concepteurs ont « triché ».

Il a établi une relation entre la longueur radiale d'une coupole et son demicoté telle que, connaissant cette longueur, nous saurons en déduire quelles pièces doivent occuper le coté.

Pour démontrer sa théorie, l'auteur a pris l'exemple suivant :

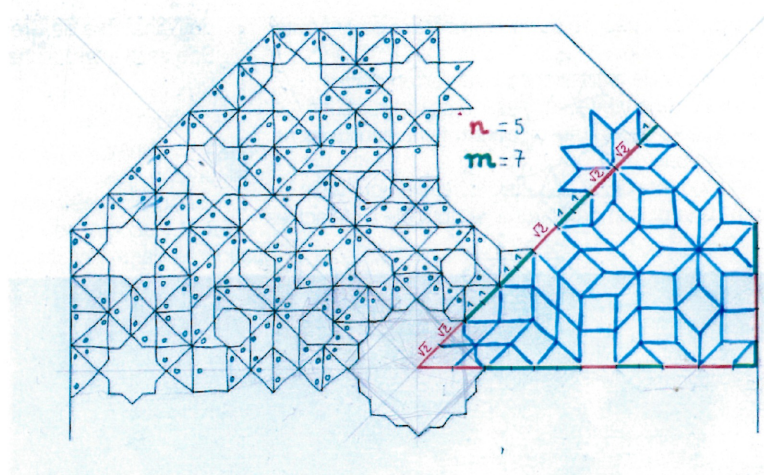


Figure 1 Tracé de la coupole de la salle des Deux Sœurs Palais Alhambra, Grenade

Deux grandeurs sont à considérer : celle des petits segments verts et qui représentent le « miqyas » ou le module et celle des grands segments rouges qui sera donc racine de 2 (la racine carrée de 2 multiplié par module au carré).

Connaissant le nombre n du grand segment et m du petit segment sur l'axe radial, on aimerait savoir quel nombre p des grands segments et q des petits segments doivent occuper le demi-côté de l'octogone.

Donc on a :

$$BC = AB * \operatorname{tg}(22.5)$$

En se souvenant que $\operatorname{tag} (17/8) = \sqrt{2} - 1$,

l'égalité ci-dessus peut s'écrire en fraction de n , m , p et q :

$$p * \sqrt{2} + q = (n * \sqrt{2} + m) * (\sqrt{2} - 1)$$

D'où la solution recherchée :

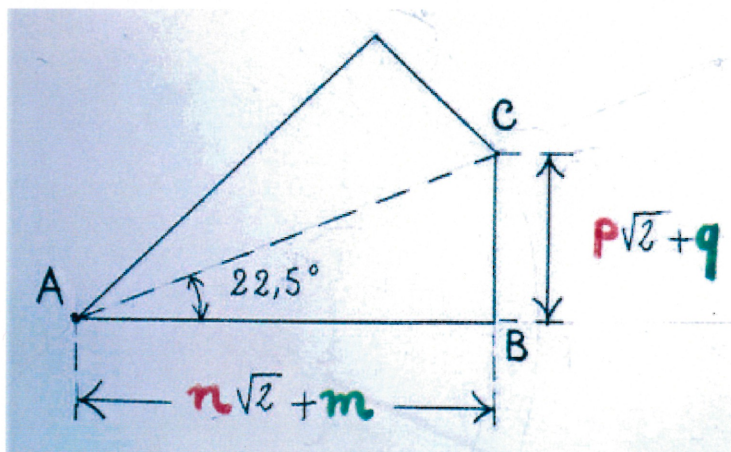
$$P = m - n$$

$$Q = 2 * n - m$$

On vérifie immédiatement dans la figure 39 que :

Avec $n = 5$ et $m = 7$, on trouve $p = 7 - 5 = 2$ et $q = 10 - 7 = 3$.

Nous avons bien deux grands segments et trois petits sur le demi-côté.



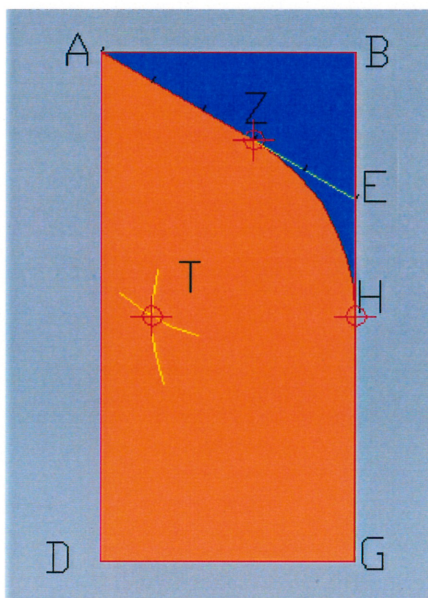
Traduction de la partie d'obtention de la partie creuse

Le maçon dessine un rectangle dont la longueur **AB** est égale au module des mouqarnas et la longueur **BG** est égale au double du module.

À partir du point **A**, il dessine la ligne **AE**, de telle sorte que **BAE = 30** degrés et il divise le segment **AE** en 5. Ils déterminent ainsi, à partir du point **E** le segment **EZ** qui est égal à $2/5^{\text{ème}}$ de **AE** ainsi que le segment **EH** qui est égal à **EZ**.

Ils dessinent 2 arcs de cercles de rayon **ZH** et de centres respectifs **Z** et **H** qui s'intersectent à l'intérieur du rectangle en un point **T**.

L'arc **ZH** de centre **T** est égal à $1/6$ de la circonférence



Méthode du Masson établie par Al-Kashi dans son livre « Miftah al-Hissab »

3.5.1.2 Principe de composition des mouqarnas

La base des tracés compliqués des mouqarnas est le carré, celui-ci, en tournant sur sa diagonale, donne l'étoile à huit points (mthemen) ou l'octogone. Les 2 axes des tracés sont orientés à 45 et 90 degrés. Les pièces cités plus haut sont débitées dans des sections ayant pour coté, soit le coté du carré, soit sa diagonale.

Par conséquent, pour bâtir un plafond dans une demi-sphère, il faudra la transposer en une demi-sphère octogonale, les angles d'articulation étant toujours des multiples de 45 degrés.

L'arc brisé qui forme le faux mouqarnas, la « bouja », devient une petite coupole octogonale, ou plutôt une série de petites coupoles qui, une fois placées régulièrement autour du centre et à la même hauteur, permettent aux mouqarnas de s'adapter à la sphéricité de l'intérieur de la coupole.

Plus nous voudrions suivre la courbure, plus nous devons utiliser les éléments destinés à devenir stalactites.

Nous pourrions ainsi donner aux différentes bandes de mouqarnas qui partagent la demi sphère des profils différents qui s'éclateront plus ou moins du profil de la coupole.

Dans les constructions sous voûtes, les bandes octogonales se réunissent au centre, au sommet, autour d'une petite coupole étoilée en huit, le chapeau (la chéchia).

Le chapeau, ainsi que les coupoles intermédiaires, est composé d'un fond autour duquel se rassemblent en étoiles des pièces très spéciales appelés « louza » toutes disposées sur un même plan horizontal. La face interne est soit saillante de 135 degrés, formée de huit pièces disposées dans un cercle formant un sceau de Salomon, soit concave avec angle rentrant de 135 degrés, formée de quatre pièces disposées sur un cercle formant un octogone régulier.

Nous pouvons transformer le cercle, base de la coupole, en « seau de Salomon », octogone étoilé, ou bien en polygone à seize cotés ou encore en étoile à seize points.

4 Chapitre 4 : État des connaissances

4.1 Objectif

Dans ce chapitre, nous allons présenter deux approches informatiques adoptées pour modéliser des voûtes mouqarnas.

4.2 Article1: « Muqarnas visualisation at the IWR »

Adresse électronique

<http://www.iwr.uni-heidelberg.de/iwr/ngg/Muqarnas/>

Auteurs

La modélisation 3d de la voûte et de ces éléments :

Tobias Illenseer et Christian Reichert

Visualisation de l'architecture :

Kurt Sätzler

Architecture et mathématique islamique :

Yvonne Dold-Samplonius

4.2.1 Objectif

En partant de la définition que le mathématicien arabe « Al Kashi » a donné des mouqarnas dans son livre « Miftah Al-Hisab », les auteurs de cet article ont supposés qu'il serait possible de créer des modèles numériques tridimensionnels des différentes pièces mouqarnas.

Cette modélisation 3d servira à (re)construire des compositions mouqarnas qui ont été détruites ou même jamais construites.

Pour répondre à leurs objectifs, ils ont décidé de reconstruire la chambre forte de l'octogone sud du « Takht-i Süleyman » qui est situé dans une large vallée, à 2000 m d'altitude et à environ 3km de Takab, au nord ouest de Téhéran en Iran. Dans les ruines de ce palais, un plan a été retrouvé et identifié comme étant un tracé régulateur mouqarnas. Ulrich Harb a proposé un plan possible pour le reconstruire et c'est le même qui a été utilisé dans ce projet.

La première étape a consisté à étudier la géométrie de chaque pièce d'après les données trouvées dans le livre de « Al-Kashi » pour construire leurs modèles 3d correspondant sur une station Selicon Graphics. Ensuite, en se basant sur le plan trouvé, ils ont assemblé les différentes pièces pour obtenir une version possible de la voûte du « Takht-i Süleyman ».

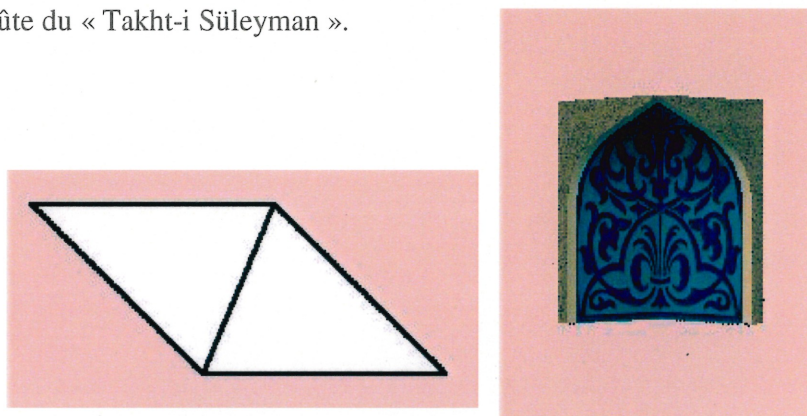


Figure39 La pièce « rhombus » en 2d et en 3d.

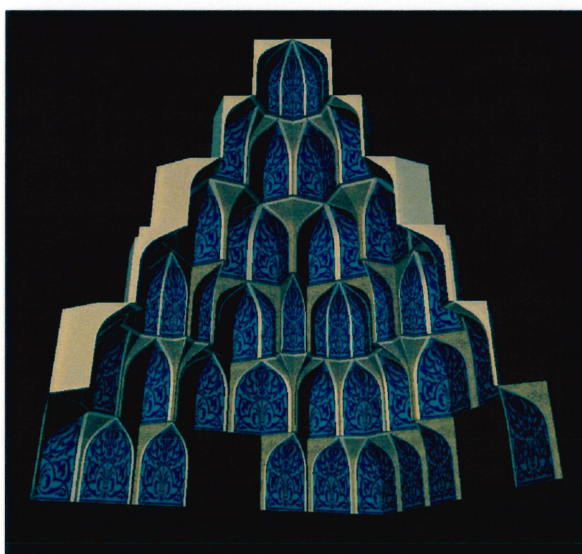


Figure40 Modèle 3d de la voûte de « Takht-i Süleyman »

4.2.2 Conclusion

On remarque 2 grands volets qui caractérisent ce travail :

- Le premier concerne un travail de collecte d'information et d'interprétation de données : Ils ont essayé de comprendre et de créer les liens entre la définition que « Al-Kashi » a donné des mouqarnas et leur plan de départ pour pouvoir établir le(s) lien(s) entre le plan régulateur et le modèle 3d de la voûte.
- Le 2^{ème} concerne la modélisation 3d de la voûte: ils ont modélisé manuellement chaque primitive pour ensuite les regrouper (manuellement aussi) en se basant sur le tracé pour arriver à la fin au modèle 3d.

Les auteurs de cet article ont essayé de comprendre la relation qui existe entre les représentations 2d et les modèles 3d des mouqarnas et, en se basant sur le plan d'une voûte qui a été détruite, ils ont reconstruit son modèle correspondant.

Mais ils ont complètement séparé ces 2 volets, c'est à dire qu'ils ont traité chacune à part et le travail s'est rendu au stade de la représentation simple d'une seule voûte mouqarnas sans coder le processus de génération. On doit donc répéter pour chaque pièce mouqarnas le même travail.

4.3 Article 2: "Mouqarnas : secrets des orientaux, un composant innovateur avec rythmiques et lumière »

Adresse

<http://www.architektur.uni-stuttgart.de/users/Ibrahim/inhalt.htm>

Auteur

Ing diplômés Sayed M. A. Ibrahim

Ce projet a été réalisé en coopération entre l'institut de photogrammétrie et le département d'urbanisme de l'université de Stuttgart.

4.3.1 Objectifs

- La production d'un modèle 3D à partir de photos et à l'aide du logiciel "Photo Modeleur".
- Le transfert de ce modèle 3D vers VRML. (Figure 41)

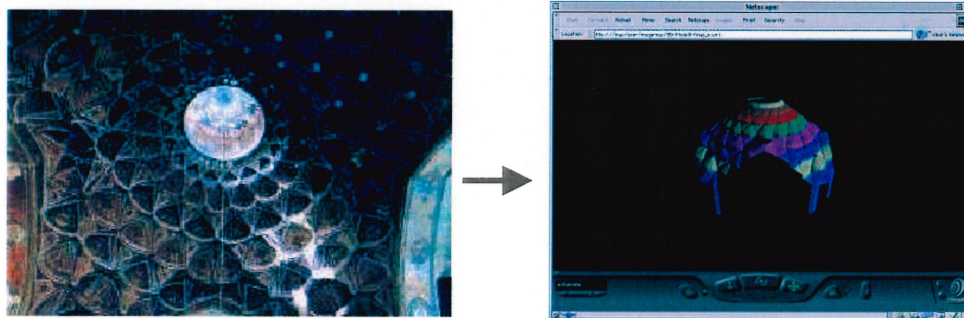


Figure41 Modèle 3d de la voûte étudiée et transférée en VRML

4.3.2 Conclusion

Ce projet constitue une autre approche possible pour la modélisation des voûtes mouqarnas à partir de modèles réels à partir desquels et par des procédés de relevé photogramétrique, on peut obtenir les modèles numérique 3d correspondants.

5 Chapitre 5 : La partie Informatique

5.1 Objectif

Nous nous proposons dans ce chapitre de présenter une méthode qui permet de générer, à partir d'un tracé régulateur d'une voûte mouqarnas en bois, le modèle tridimensionnel correspondant.

5.2 Choix de l'outil

L'objectif de notre recherche est de trouver les moyens informatiques permettant de générer, à partir d'un tracé régulateur bidimensionnel, le modèle tridimensionnel correspondant. L'outil que nous allons utiliser doit forcément avoir les caractéristiques suivantes :

- Un outil qui gère aussi bien le 2d que le 3d.
- Un outil qui nous permet de programmer des actions et de préciser des conditions qui nous permettront de reconnaître nos pièces et de programmer leur construction en 3 dimensions.

Il existe quelques logiciels qui répondent à ces critères (exemple : Archicad, Autocad..) et c'est pourquoi la disponibilité de l'outil a été un critère déterminant dans notre choix car le but de notre présente étude est d'offrir une méthode de traitement disponible pour le plus d'utilisateur possibles.

Notre choix s'est porté sur Autocad, qui à l'aide de son langage fonctionnel Auto-lisp, va répondre à l'ensemble de nos objectifs de recherche.

5.3 Stratégie générale de traitement :

Dans cette partie, nous allons expliquer la stratégie générale de traitement algorithmique que nous avons retenue.

5.3.1 Analyse des données :

Le point de départ est un tracé régulateur d'une voûte mouqarnas en bois. Ce tracé représente la projection plane d'un volume mouqarnas. Il est composé essentiellement de triangles et de quadrilatères juxtaposés. Chaque figure qu'il contient est dotée d'informations et de codes permettant sa lecture et, par conséquent, la reconnaissance des différentes pièces qui constituent le tracé. Ces informations sont la forme de la pièce et les points qui la caractérise.

On distingue deux formes générales des représentations 2d des pièces mouqarnas :

5.3.1.1 Les triangles :

On distingue deux types de triangles :

- Triangle isocèle rectangle : Les pièces "Dembouq", "Serwaliya" et "chiira".
- Triangle isocèle non rectangle : Les pièces "loza1" et "loza2".

5.3.1.2 Les quadrilatères :

Il existe plusieurs types de quadrilatères dans la famille des pièces mouqarnas, mais celles que nous allons étudier et qui sont les plus utilisées sont :

- Les losanges : Les pièces "Ktaf", "charbiya" et "chiira».
- Les rectangles : Les pièces « T'stiya masdouda » et « T 'stiya maftouha ».

5.3.2 Les points qui codent les pièces :

Les points constituent un code que les concepteurs et les artisans ont inventé pour répondre à un besoin de représentation et d'identification des différentes pièces dans un tracé mouqarnas. Ce besoin est né suite à l'invention des mouqarnas modulaires en bois car cet événement s'est accompagné de la création de nouvelles pièces et les artisans se sont retrouvés avec plusieurs pièces qui pouvaient avoir exactement la même projection plane. Les points étaient la solution qui permettait de les distinguer.

Une question très importante se pose alors: est-ce que ce nouveau code représente une solution arbitraire que les artisans ont adoptée ou bien il est doté d'une signification géométrique ? Autrement dit, est-ce que le code par points peut être déduit du tracé ou constitue-t-il un choix arbitraire?

Après une étude et une analyse approfondie de plusieurs tracés régulateurs de mouqarnas en bois, nous avons réussi à dégager une série d'observations qui nous ont conduit à dégager des lois géométriques qui nous permettront de reconnaître chacune des pièces dans le tracé sans avoir recours à leur code. Ces lois permettront à notre travail de pouvoir être généralisé, dans le futur, à tous les tracés régulateurs. En effet, en bâtissant toute notre stratégie de reconnaissance de pièces sur les points, la logique de traitement adoptée ne pourrait être généralisée et exploitée pour l'étude des tracés autres que celles des mouqarnas modulaire en bois.

5.3.2.1 Observation 1:

En calculant la distance qui sépare chaque sommet de la projection plane de n'importe quelle pièce avec le centre du tracé et si on retient la plus grande valeur, on aura 2 cas possibles :

- Si cette valeur correspond à la distance qui sépare un seul sommet du centre du tracé alors ce sommet est codé par un point. L'angle en ce point va déterminer le nombre de points qui codent la pièce.
- Si cette valeur correspond à la distance qui sépare 2 sommets du centre du tracé, ce qui veut dire qu'ils sont équidistants par rapport à ce centre, alors cette pièce est codée par 2 points qui indiquent ces même sommets.

Donc les points indiquent le ou les sommets qui sont les plus éloignés du centre du tracé par rapport aux autres sommets de la projection de la pièce qu'on étudie.

5.3.2.2 Observation 2:

La projection plane de chacune des pièces englobe des informations géométriques sur lesquels on peut s'appuyer pour l'identifier (Exemple : les longueurs de ses cotés et les angles dans chacun de ses sommets).

5.4 Stratégie de traitement :

Notre stratégie se présente sous la forme d'un cheminement du général vers le particulier. C'est à dire partir d'un tracé régulateur représentant une voûte mouqarnas en bois, identifier chacune des pièces qui le constitue et construire leurs modèles tridimensionnels correspondants (figure 42).

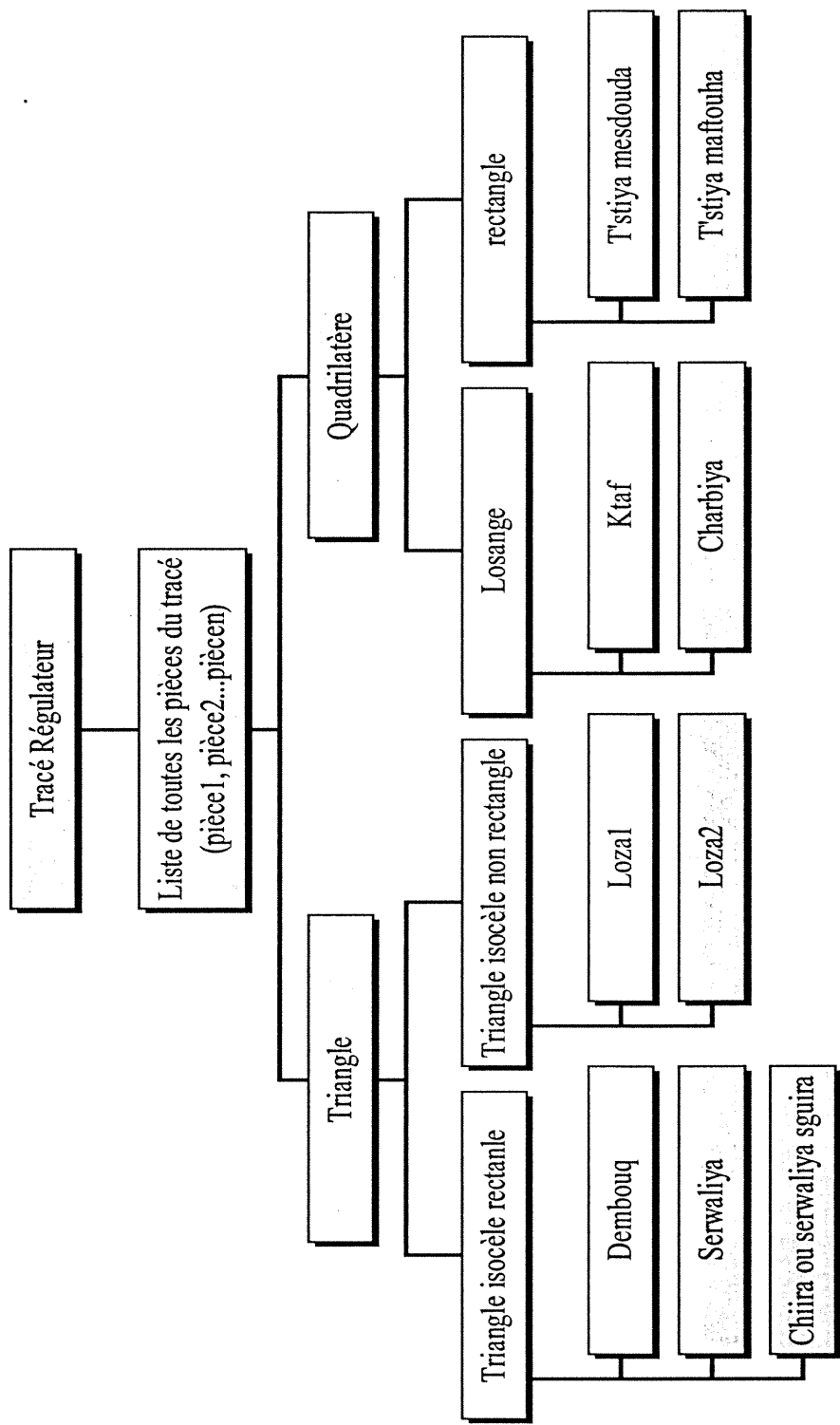


Figure 42 Logique générale de traitement algorithmique d'un tracé régulateur mouqarnas

5.4.1 Étape1 : la représentation du tracé

La première étape consiste à représenter informatiquement le tracé régulateur, objet d'étude. La façon de le représenter dépend directement du type d'informations qu'on veut obtenir.

L'objectif de cette étape est de séparer chacune des représentations planes des pièces qui constituent le tracé régulateur sous la forme d'une liste pour pouvoir l'étudier, déterminer sa nature et de programmer la construction de son modèle tridimensionnel correspondant.

Le logiciel AUTO-CAD offre deux manières possibles pour représenter le tracé :

5.4.1.1 La représentation par région :

« Une région est une zone délimitée par des contours en deux dimensions, définis à partir d'objets fermés appelés « boucles ». Une boucle est une courbe ou une séquence de courbes reliées par leurs extrémités, une aire plane au contour non auto-intersectable. Il peut s'agir d'un ensemble de lignes, de « polylignes », de cercles, d'arcs, d'ellipses, d'Arcs elliptiques, de « splines » ou de solides. La figure doit être fermée et les objets doivent être coplanaires »

Jean-Pierre Couwenbergh⁹ P : 211

5.4.1.2 La représentation sous forme d'un 3df.

« Un 3d face est l'équivalent en 3d d'un solide 2d. chaque sommet d'une face 3d peut avoir une coordonnée z différente. »

Jean-Pierre Couwenbergh Page 472

En sélectionnant le tracé, un seul des deux cas nous permet d'obtenir les informations recherchées sur la position exacte des sommets : c'est la représentation des pièces sous forme d'un « 3d face ».

⁹ L'intégrale de AUTOCAD 14

1. La première information recherchée est une liste de toutes les pièces de ce tracé : Liste : ((pièce-1) (pièce-2) (pièce-3).....(pièce-n))
2. Chaque pièce doit être sous la forme d'une liste de tous les points qui la constituent : Pièce : ((pt-1) (pt-2).....(pt-n))
3. Chaque point doit être sous la forme d'une liste de ses coordonnées dans le repère cartésien dont l'origine est le centre de la trame. Point-1 : (x-1 y-1 z-1)

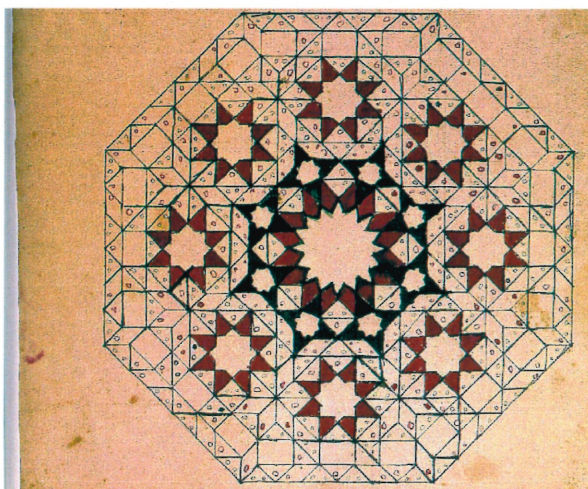


Figure43 Tracé régulateur des mouqarnas en bois¹⁰

La première étape consiste donc à « dessiner » en deux dimensions le tracé régulateur de telle façon que l'origine soit au centre et que chaque pièce soit décrite individuellement par un 3D face.

L'échelle adoptée sera telle que le module de base des mouqarnas soit égal à 100.

¹⁰André Paccard « Le Maroc et l'artisanat traditionnel islamique dans l'architecture » Page303

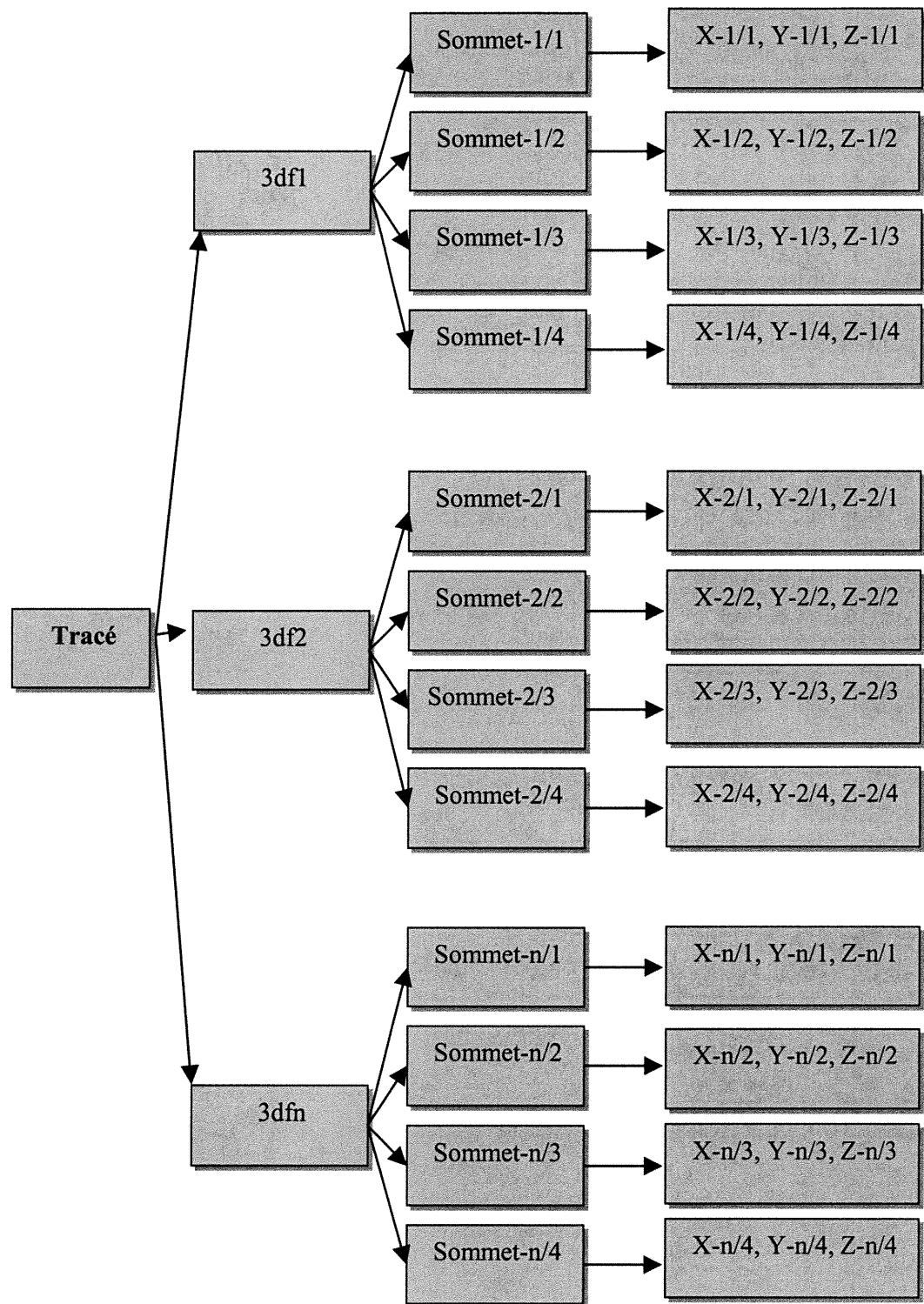


Figure44 Organisation de la liste des entités sélectionnées.

5.4.2 Étape 2 saisie des données

En sélectionnant le tracé on obtient la liste générale de tous les 3df (projection plane des pièces) contenus dans le tracé sous la forme décrite précédemment. Cette liste sera la base de tout le travail qui va suivre car toutes les procédures cherchent les informations dont elles ont besoin dans cette liste (figure 45).

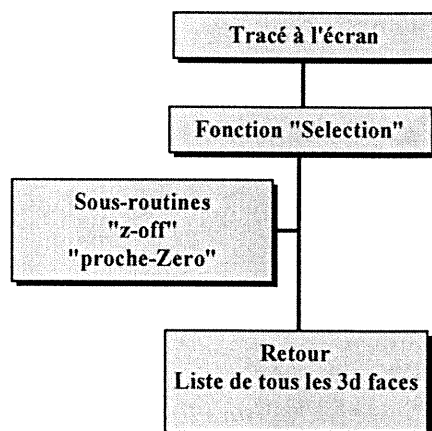


Figure 45 Saisie des données

5.4.3 Étape 3 : La reconnaissance des formes

5.4.3.1 Reconnaissance de la forme générale :

Il s'agit de déterminer la forme générale de la pièce en fonction de la liste des sommets du 3df qui le représente (figure 46).

Dans cette étape on part de la liste des 3d faces sélectionnés et on l'utilise comme argument dans la procédure « tri-quadri » qui va parcourir cette liste en traitant un 3df à la fois. Chaque 3df étant une liste de sommets, on peut identifier le type de polygone qu'il représente:

- Si la longueur de cette liste est égale à 3 (ce qui implique que c'est une liste de trois points et donc c'est un triangle) on va appliquer à la liste de points de ce 3df la procédure « traite-triangle ».

- Si la longueur de cette liste est égale à 4 (ce qui implique que c'est une liste de quatre points et donc c'est un quadrilatère) on va appliquer à la liste de points de ce 3df la procédure « traite-quadrilatère».

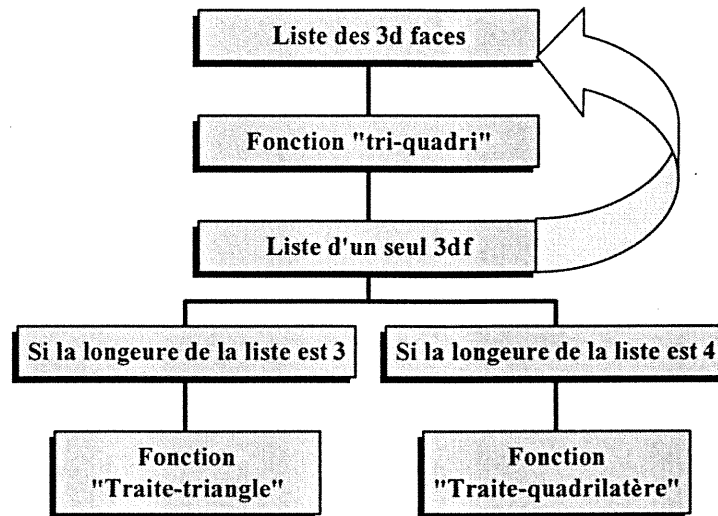


Figure 46 Reconnaissance de la forme générale

Une fois la nature de la forme définie (triangle ou quadrilatère) chaque cas sera traité différemment :

5.4.3.1.1 Si la longueur de la liste du 3df est égale à 3:

On applique à la liste du 3df la procédure « traite-triangle » qui va exécuter les étapes suivantes :

- Calculer les 3 angles formés à chaque sommet de ce triangle et les consigner dans la liste (11).
- Calculer les longueurs d_1 , d_2 et d_3 des 3 cotés du triangle en traitement, c'est à dire la distance entre le point 1 et le point 2, le point 2 et le point 3 et le point 1 et le point 3.

Ces calculs, à partir de la liste des 3 points d'un 3df, sont exécutés par les sous-routines suivantes :

- Pour les angles, ils sont calculés par les sous-routines « angle3df » et « angle-triangle » qui à l'aide de la procédure « cal » de autolisp, calcule les angles dans chacun des 3 sommets du triangle et les retourne dans la liste (l1)
- Pour les distances, elles sont simplement calculées par la procédure « distance ». (Exemple : (setq d1 (distance (car 3df) (cadr 3df))))

Une fois la liste (l1) des angles et les 3 longueurs des cotés des triangles calculés, on va procéder par une série de conditions pré-établis à l'identification de la nature de la pièce en traitement (figure 47) :

Pour les pièces « loza1 » et « loza2 »:

Pour que la pièce représentée sous la forme d'une liste de 3 points soit une « Loza » il faut que la liste répond à la série de conditions suivantes :

- Condition 1: Si d1, d2 et d3 sont différent de 200 (2 fois le module qui est égal à 100).
- Condition 2 : Si les angles 45 et 67.5 sont membres de l1.
- Condition 3 : Si en calculant la distance de chaque point du triangle par rapport au centre du tracé on trouve que 2 points sont équidistants du centre et que cette distance est supérieure à la distance entre le 3ème point de liste et le centre du tracé.

Si ces conditions sont réunies, les informations seront renvoyées vers la procédure « loza1-2 » pour déterminer laquelle des deux pièces il s'agit :

- Condition 1 : Si l'angle au sommet le plus loin du centre du tracé est égal à 67.5° alors la pièce en traitement est loza1. La nature de la pièce ainsi déterminée, la liste du 3df est alors traitée par la procédure « loza1 » pour construire son modèle tridimensionnel correspondant.
- Condition 2 : Si l'angle au sommet le plus loin du centre du tracé est égal à 45° , alors la pièce en traitement est loza2. La nature de la pièce ainsi

déterminée, la liste du 3df est alors traitée par la procédure « loza1 » pour construire son modèle tridimensionnel correspondant.

Pour la pièce « serwaliya » :

Pour que la pièce représentée sous la forme d'une liste de 3 points soit une « serwaliya », il faut qu'elle réponde à la série de conditions suivantes :

- Condition 1 : Si l'une des longueurs d_1 , d_2 ou d_3 est égale à 200 (2 fois le module).
- Condition 2 : Si la liste l1 contient l'un des deux angles suivant 45^0 ou 90^0 .
- Condition 3 : la liste doit vérifier l'une des conditions suivantes :
 - Si en calculant la distance de chaque point du triangle par rapport au centre du tracé, on trouve que 2 points sont équidistants du centre et que cette distance est supérieure à la distance entre le 3ème point de liste et le centre du tracé.
 - Si en calculant la distance de chaque point du triangle par rapport au centre du tracé, on trouve que seul un point est le plus éloigné du centre et que l'angle en ce point est 45^0 .

Pour les pièces « Dembouq » ou « serwaliya sguira » :

Pour que la pièce représentée sous la forme d'une liste de 3 points, soit une « Dembouq » ou « serwaliya sguira », il faut que la liste vérifie la série de conditions suivantes :

- Condition 1 : Si l'une des longueurs d_1 , d_2 ou d_3 est égale à 200
- Condition 2 : Si la liste l1 contient l'un des angles suivant : 45^0 ou 90^0 .
- Condition 3 : Si en calculant la distance de chaque point du triangle par rapport au centre du tracé on trouve qu'un seul point est le plus éloigné du centre et que cette distance est supérieure à la distance entre le 2ème point de la liste et le centre du tracé.

Géométriquement, ces deux pièces ont exactement la même description car leur projection plane et leur orientation est la même. Donc l'utilisateur a le choix d'utiliser l'une ou l'autre suivant sa lecture du tracé régulateur d'origine.

On a donc, dans ce cas, une alternative d'identification que les règles ne permettent pas de réduire à une seule solution.

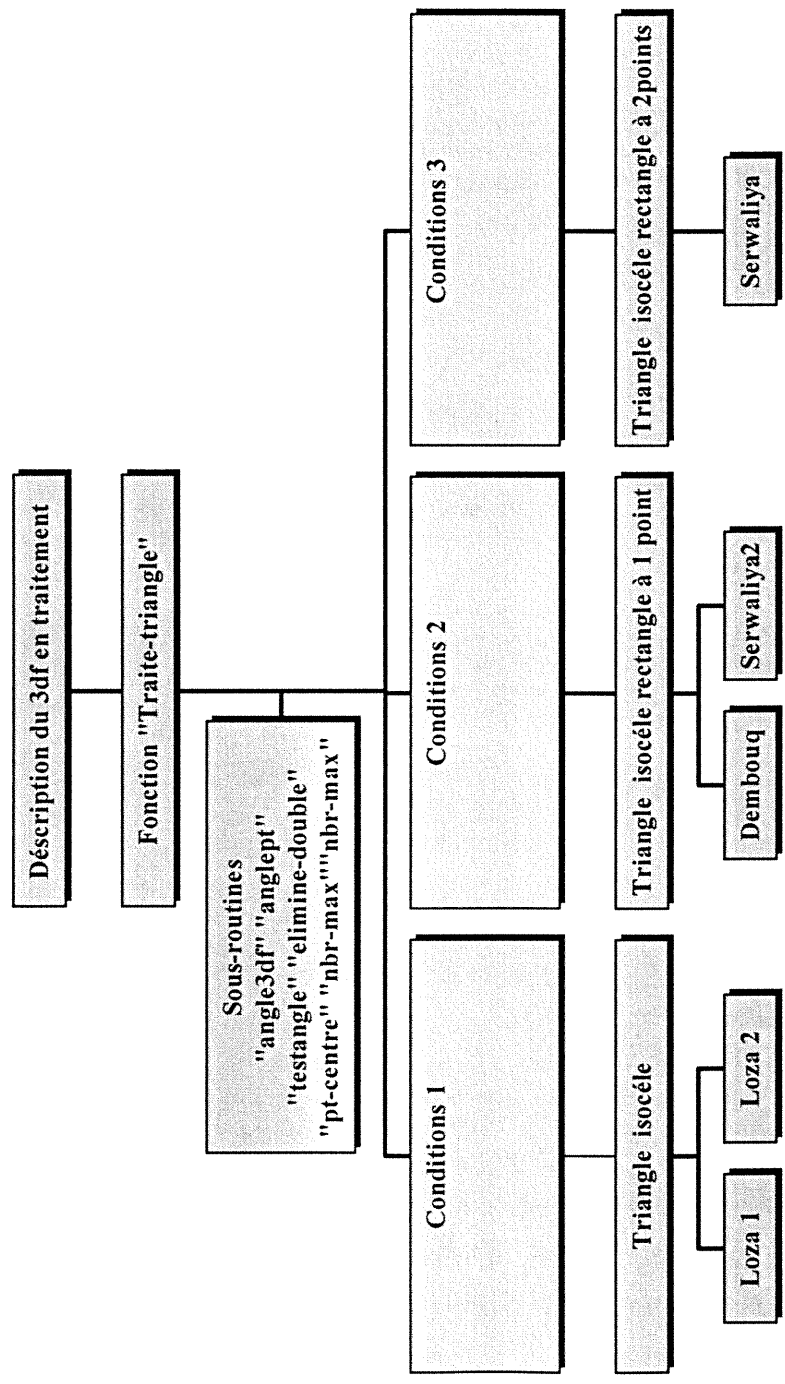


Figure 47 Traitement des triangles

5.4.3.1.2 Si la longueur de la liste du 3df est égale à 4

On applique à la liste du 3df la procédure « traite-quadrilatère » qui va exécuter les étapes suivantes :

- Créer une liste l1 des 4 angles formés à chacun des sommets de ce quadrilatère.
- Calculer les longueurs d1, d2 d3 et d4 des 4 cotés du quadrilatère en traitement, c'est à dire la distance entre le sommet 1 et le sommet 2, le sommet 2 et le sommet 3, le sommet 3 et le sommet 4 et le sommet t 4 et le sommet 1.

Une fois la liste (l1) des angles et les 4 longueurs des cotés du quadrilatère calculés, on va procéder par une série de conditions pré-établies pour identifier la nature exacte de la pièce en traitement (figure 48) :

Pour les losanges :

Pour que la forme de la pièce présentée sous forme d'une liste de 4 points soit un losange, il faut que la liste répond à la série de conditions suivantes :

- Condition 1 : Si les quatre mesures des cotés de la pièce, c'est à dire d1, d2, d3 et d4, soient égales à 200.
- Condition 2 : Si en calculant la distance qui sépare chacun des sommets du quadrilatère au centre du tracé on trouve qu'un seul point est le plus éloigné du centre. C'est à dire que cette distance est supérieure à la distance entre tous les autres éléments de liste et le centre du tracé.

Si la liste du 3df vérifie ces conditions, elle sera traitée par la procédure « losange1-3 » qui va déterminer la nature exacte de la pièce à savoir une ktaf ou une Chiira :

- Condition 1 : Si l'angle au sommet le plus loin du centre du tracé est égal à 45° , alors la pièce en traitement est Ktaf. La nature de la pièce ainsi déterminée, la liste du 3df est alors traitée par la procédure « Ktaf » pour construire son modèle tridimensionnel correspondant.

- Condition 2 : Si l'angle au sommet le plus loin du centre du tracé est égal à 135^0 , alors la pièce en traitement est charbiya. La nature de la pièce ainsi déterminée, la liste du 3df est alors traitée par la procédure « charbiya » pour construire son modèle tridimensionnel correspondant.

Pour les rectangles :

Pour que la forme de la pièce présentée sous forme d'une liste de 4 points soit un rectangle, il faut que la liste réponde à la série de conditions suivantes :

- Condition 1 : Si l'une des quatre mesures des cotés de la pièce, c'est dire d_1 , d_2 , d_3 ou d_4 , est égale à 200.
- Condition 2 : Si en calculant la distance de chaque point du quadrilatère par rapport au centre du tracé on trouve que 2 points sont équidistants du centre et que cette distance est supérieure à la distance entre le 3ème point de liste et le centre du tracé.

Les propriétés géométriques des deux pièces représentées bidimensionnellement par un rectangle sont les mêmes. Donc l'utilisateur a le choix d'utiliser l'une ou l'autre suivant sa lecture du tracé régulateur d'origine.

On a donc, dans ce cas, une autre alternative d'identification que les règles ne permettent pas de réduire à une seule solution.

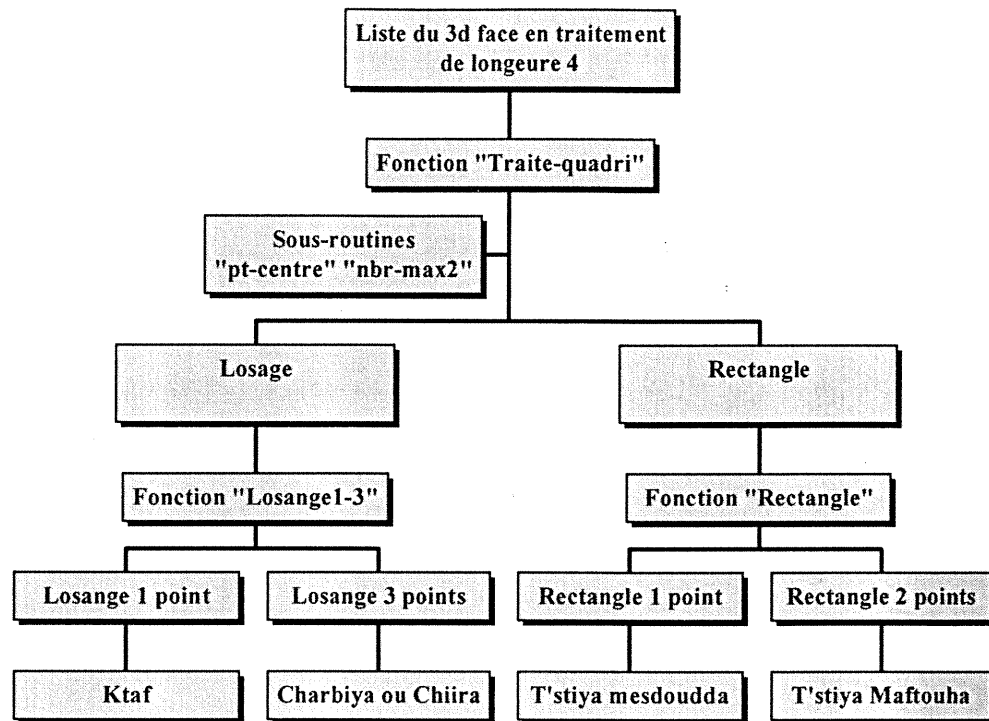


Figure 48 Traitement des quadrilatères

5.4.4 Étape 4 : La construction des pièces :

Une fois la nature de la pièce identifiée, on procède directement à la construction du modèle de la pièce en 3 dimensions. Pour exécuter cette étape, trois stratégies sont possibles :

- La première consiste à construire toutes les pièces au centre du tracé, puis à l'aide de matrices de translation et de rotation, les placer exactement sur la représentation bidimensionnelle de la pièce en traitement.
- La deuxième stratégie consiste à former une bibliothèque de tous les modèles 3d des pièces et puis, une fois la nature de la pièce en traitement connue, on insère le modèle 3d correspondant sous forme d'un bloc en utilisant les éléments de la liste des sommets du 3df comme des points d'insertion.

- La troisième stratégie consiste tout simplement à construire directement le modèle 3d de la pièce en traitement sur le tracé en utilisant à chaque fois les éléments de la liste du 3df.

La première stratégie nous semblait au début la plus intéressante et c'est celle pour laquelle nous allions opter. Mais suite aux problèmes d'imprécision auxquels nous avons eu à faire face avec Autocad, et que nous allons expliquer plus loin, nous avons commencé à avoir des pièces qui se coupent et qui ne sont pas placées exactement là où il le faut. Cette solution n'était donc pas fiable.

La deuxième stratégie est abandonnée pour la même raison que la première. En plus nous voulions avoir des procédures indépendantes d'une bibliothèque de modèles que nous aurons à transférer à chaque poste de travail que nous utiliserons pour exécuter notre programme.

La troisième stratégie est celle que nous avons choisi, car en partant directement des coordonnées des sommets du 3df en traitement, nous sommes sûr que notre pièce sera construite et placée à sa bonne place.

5.4.4.1 la logique de construction des pièces :

La construction de toute pièce se fait directement sur sa représentation 2d, donc il s'agit de trouver la meilleure stratégie possible pour construire les pièces de la même manière quelle que soit sa position sur le tracé. Le mode de traitement doit être valable quand la pièce est proche du centre et quand elle est à son extrémité, aussi bien à gauche du centre qu'à sa droite. Le meilleur procédé que nous avons trouvé était de placer toujours l'UCS (le repère) sur la pièce elle-même et de choisir le centre et les directions de ce nouveau repère en fonction des particularités permanentes de ses sommets.

Dans la géométrie des pièces mouqarnas, les parties courbes sont les éléments déterminants, en fonction desquels toute la stratégie de modélisation doit être étudiée. En fonction de la complexité de ces parties, on peut en déduire la stratégie qu'on devrait adopter pour réussir à programmer le passage 2d/3d des pièces.

On trouve trois grandes stratégies de modélisation des pièces mouqarnas : la stratégie des pièces simples, la stratégie des pièces semi-complexes et la stratégie des pièces complexes.

5.4.4.1.1 Stratégie des pièces simples:

C'est la stratégie utilisée pour modéliser les pièces « Dembouq », « T'stiya mesdouda », « Loza1 » et « Loza2 » (figure 49). La construction de ces pièces passe par 3 étapes :

Construction du prisme :

En partant de la liste des points du 3df, on trace un « pline » reliant les sommets et on l'extrude à la hauteur de la pièce.

Construction des cylindres :

Après un changement d'UCS bien étudié selon les particularités de la forme courbe de la pièce en construction, on place l'ellipse qu'on va ensuite extruder en un cylindre elliptique.

Construction de la pièce :

À l'aide de l'opérateur booléen « Soustraction », on déduit du volume du prisme celui du cylindre pour obtenir la pièce mouqarnas voulue.

5.4.4.1.2 Stratégie des pièces semi-complexes:

C'est la stratégie utilisée pour modéliser les pièces « Ktaf », « T'stiya maftouha » et « Serwaliya-test » (figure 50). La construction de ces pièces passe par 3 étapes :

Construction des prismes :

Dépendamment de la pièce qu'on modélise, on divise la forme 2d en 2 parties. On détermine 2 listes des sommets de chacune de ces parties qu'on extrude ensuite pour obtenir 2 prismes indépendants l'un de l'autre car chacun va être traité tout seul avec la méthode de stratégie simple.

Construction des ellipses :

On construit pour chacun des deux prismes le cylindre elliptique avec la même méthode qu'on utilise pour la stratégie précédente.

Construction de la pièce :

À l'aide de l'opérateur booléen « Soustraction », on déduit de chacun des 2 volumes de prisme le volume du cylindre elliptique qui lui est associé pour ensuite unir les deux volumes résultant en un seul qui représente la pièce mouqarnas voulue.

5.4.4.1.3 Stratégie des pièces complexes:

C'est la stratégie utilisée pour modéliser les pièces « Serwaliya2 » et « Charbiya » (figure 51). C'est presque la même stratégie que la précédente sauf qu'il faut prévoir la construction des « pieds » pour les pièces. Pour les construire, nous avons eu le choix entre 2 stratégies que nous avons finalement appliquées :

5.4.4.2 Stratégie pour la construction du pied de la pièce « Charbiya » :

Pour cette pièce, on trouve la liste des 4 sommets du carré qui représente la section du pied qu'on veut avoir, on dessine le « pline » reliant ces points qu'on extrude pour avoir le volume désiré.

5.4.4.3 Stratégie pour la construction du pied de la pièce « serwaliya2 » :

Pour cette pièce, on trouve la liste des coordonnées du centre du carré qui représente la section du pied qu'on veut avoir, puis à l'aide de la commande « polygon », on dessine le carré de côté (module / 4) qu'on extrude de (3 * module) pour obtenir le volume désiré.

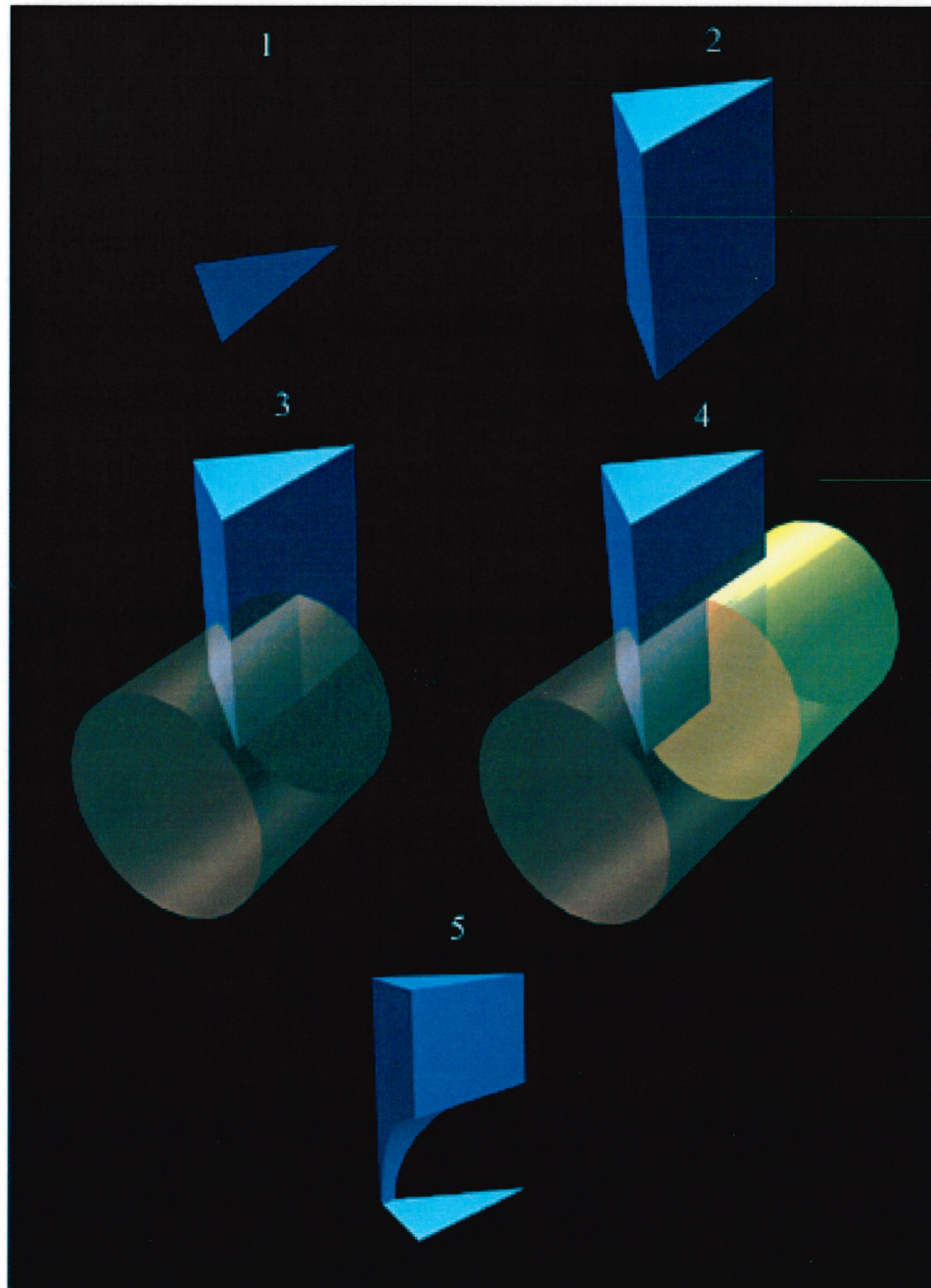


Figure49 Stratégie des pièces simples : Exemple de la pièce « Dembouq »

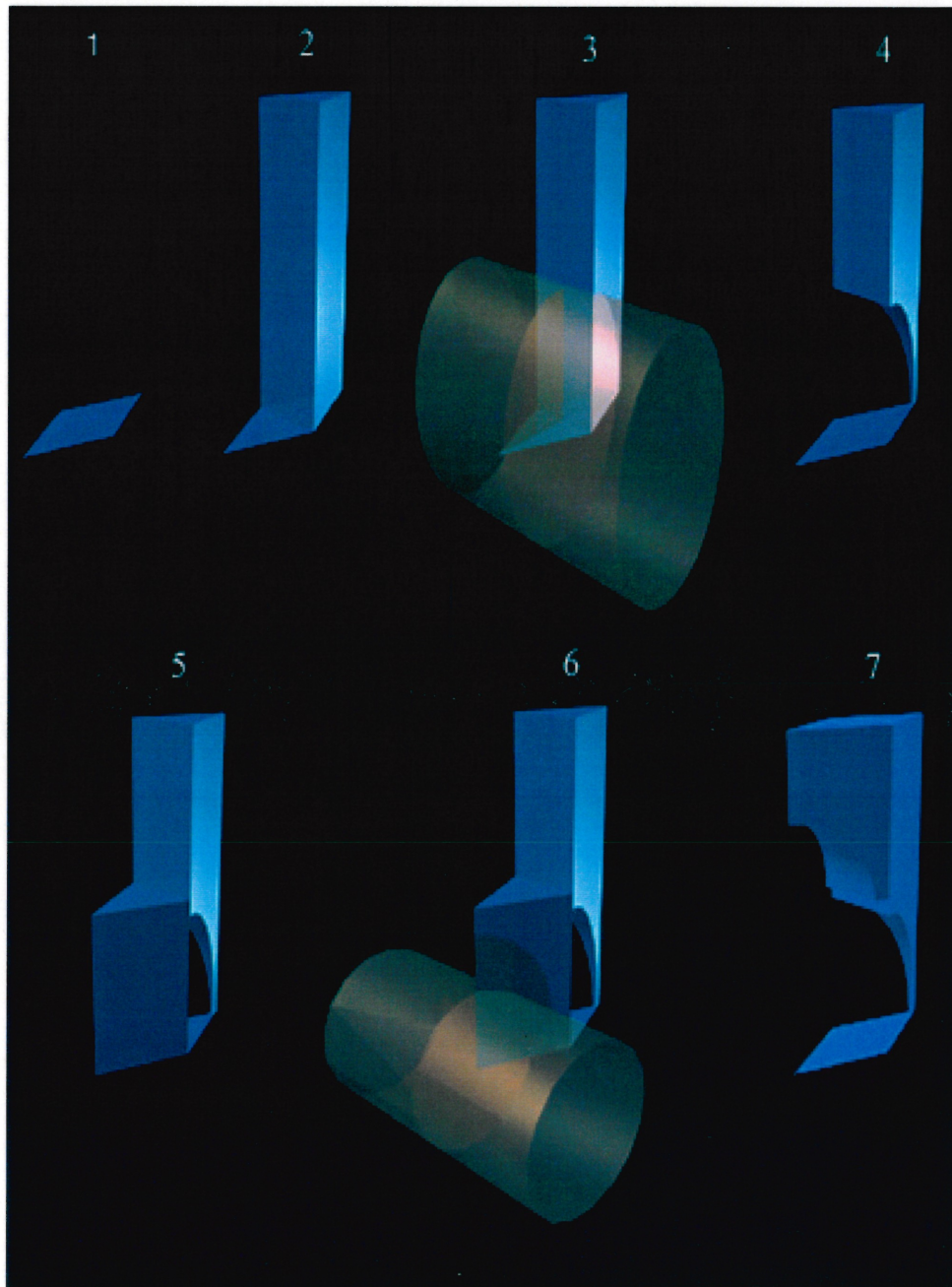


Figure50 Stratégie des pièces semi-complexes : Exemple de la pièce
« Ktaf »

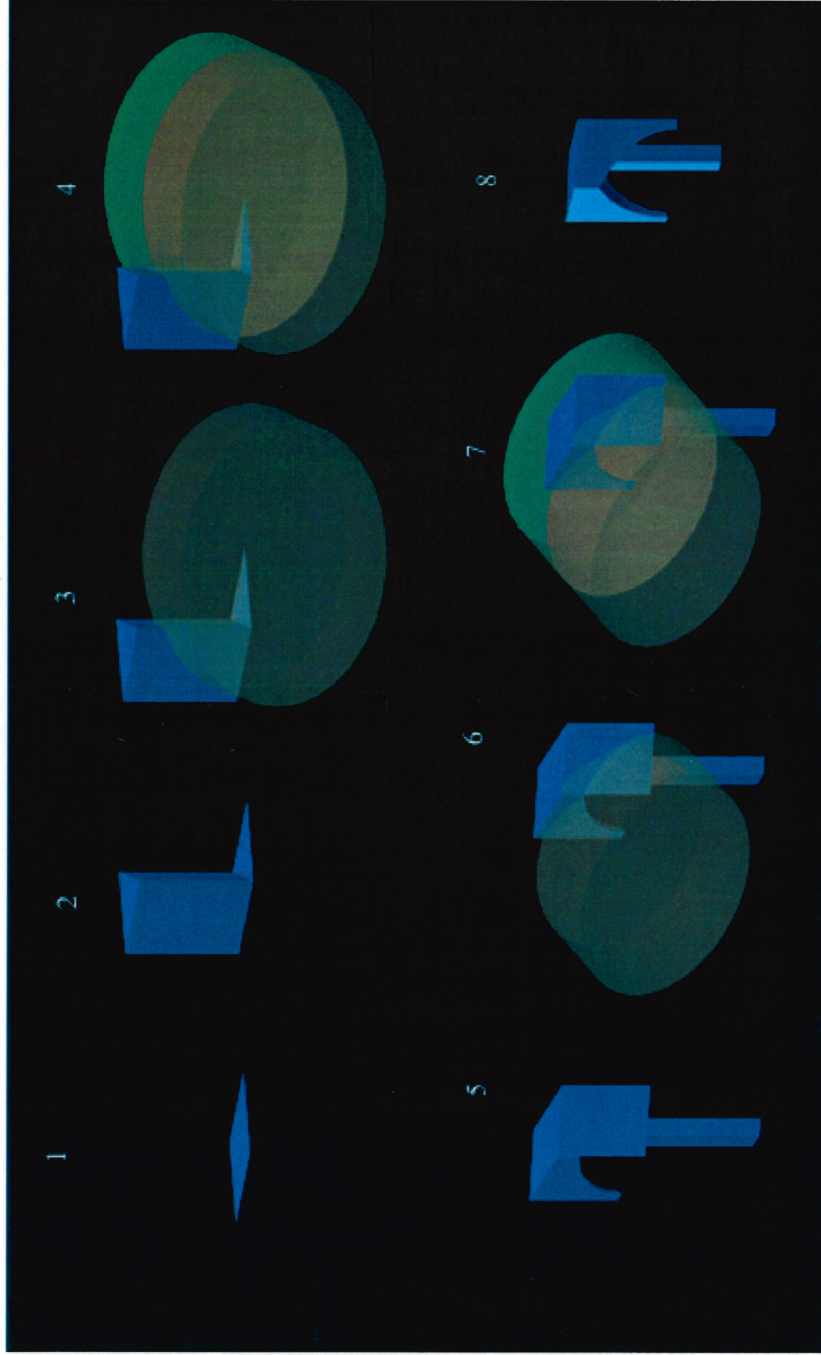


Figure51 Stratégie des pièces complexes : Exemple de la pièce « Charbiya»

5.4.5 Le traitement du tracé : Le système de couches

Dans la partie précédente, nous avons exploré la stratégie informatique pouvant être utilisée pour, à la fois, reconnaître une pièce à partir de sa représentation bidimensionnelle et ensuite construire son modèle tridimensionnel correspondant.

Dans cette partie, il s'agit d'expliquer la logique informatique adaptée pour passer du tracé régulateur des voûtes mouqarnas en bois vers le modèle tridimensionnel qu'il représente.

5.4.5.1 La logique de traitement :

Les pièces qui constituent l'alphabet des mouqarnas sont conçues pour être disposées en une composition qui délimite l'espace de façon partielle ou complète. À cet effet, elles doivent être liées les unes aux autres dans la direction horizontale aussi bien que dans la direction verticale. Les pièces adjacentes situées dans un même niveau horizontal appartiennent à la même rangée, donc une rangée représente les unités situées à la même hauteur et sont adjacentes l'une à l'autre.

Une étape nécessaire dans la transformation d'un tracé régulateur mouqarnas en une composition tridimensionnelle nécessite sa division en rangées distinctes de même hauteur.

On peut distinguer deux types de tracés :

- Les tracés où la disposition des pièces commence à partir du centre.
- Les tracés où les pièces se disposées autour d'une coupole étoilée : La « chéchia ».

On suppose dans les deux cas qu'il n'existe pas de coupoles étoilées autre que celle du centre car le traitement ne sera pas le même.

Dans les deux cas, c'est seulement la première rangée qui sera traitée différemment; pour toutes les rangées suivantes le traitement sera le même.

5.4.5.2 Saisie des données et traitement de la première rangée :

La procédure « mouqarnas » est la procédure principale qui sera initialisée au début du traitement. Elle est programmée de façon à guider l'utilisateur au début du processus de traitement afin de récupérer des données « non-programmables » qui peuvent varier d'un tracé à un autre. L'utilisateur est invité à sélectionner le tracé qu'il veut traiter et ensuite il lui est demandé si les pièces du tracé commencent à partir du centre.

5.4.5.2.1 Les tracés où les pièces commencent au centre

La stratégie de traitement consiste à trouver dans la liste des 3d faces sélectionnées ceux qui ont le centre dans leur liste de sommets. C'est la procédure « trouve » qui parcourt la liste générale des entités sélectionnées en traitant un 3df à la fois pour vérifier si le centre est contenu dans la liste des sommets du 3df. Si oui, il ajoute la liste des sommets de ce 3df à la liste (Sélection) et continue son traitement jusqu'à ce que tous les 3d faces au centre soit identifiés. Ainsi on obtient les 3d faces qui constituent la première rangée.

5.4.5.2.2 Les tracés où les pièces se réunissent autour d'une coupole étoilée

La stratégie consiste à parcourir la liste des segments que forme la coupole étoilée pour trouver dans la liste des 3df sélectionnés ceux qui ont un des segments de la coupole étoilée. Les 3d faces qui vérifient ces conditions forment la première rangée. C'est la procédure « sommetint » qui, en invitant l'utilisateur à sélectionner dans l'ordre les sommets intérieurs du tracé qui forment la coupole étoilée, transforme cette liste en une liste de segments. La procédure « sniwa » intervient ensuite pour chercher dans la liste de tous les 3d faces sélectionnées ceux qui ont les 2 points qui forment un des segments dans leurs listes de sommets. Ainsi on obtient la liste « sélection » des 3dfaces qui constituent la première rangée.

5.4.5.3 Le traitement de la 1^{ère} rangée

Une fois la liste « sélection » des 3dfaces qui forment la première rangée constituée, elle va être traitée récursivement, selon les quatre étapes suivantes :

1^{ère} étape :

La liste « sélection » va être traitée par la procédure « tri-quadri » pour déterminer la nature de chacun des 3dfaces qui la forme et les constructions des pièces tridimensionnelles correspondantes.

2^{ème} étape :

La rangée construite va être placée à sa hauteur correspondante et ceci selon le principe suivant : Le centre du tracé est le point le plus haut du modèle 3d d'une voûte mouqarnas. Donc, plus une rangée de pièces est proche du centre plus elle est située à une hauteur supérieure par rapport à la rangée suivante. En adoptant ce principe, la première rangée est la plus élevée et par suite la plus proche du centre. Elle sera située au même niveau que le tracé, c'est-à-dire à une hauteur ($h = 0$).

3^{ème} étape :

Une fois la liste « sélection » traitée et placée à son niveau correspondant, on enlève de la liste générale des 3dfaces du tracé ceux qui ont été traités (donc la liste (sélection)).

4^{ème} étape :

Il s'agit de transformer la liste des 3df dans la « sélection » en une liste de segments (lseg-f).

5.4.5.4 Le traitement des autres rangées

Une fois la première rangée traitée et la liste (lseg-f) des segments construite, on cherche, toujours avec la procédure « sniwa », dans la liste des 3d faces non traités ceux qui ont un segment commun avec la liste (lseg-f) et on les sélectionne pour obtenir à la fin la liste de tous les 3d faces qui forment la rangée suivante.

Cette liste (selection) va être traitée suivant les quatre étapes suivantes :

1^{ère} étape :

La liste « sélection » va être traitée par la procédure « tri-quadri » pour déterminer la nature de chacun des 3dfaces qui la forme et la construction des pièces tridimensionnelles correspondantes.

2^{ème} étape :

La rangée construite va être placée à sa hauteur correspondante et ceci selon le principe suivant : Le centre du tracé est le point le plus haut du modèle 3d d'une voûte mouqarnas, donc plus une rangée de pièces est proche du centre plus elle est située à une hauteur supérieure par rapport à la rangée suivante. Donc, en suivant ce principe, la rangée suivante est située à une hauteur ($h = h \text{ précédant} - \text{module}$), et plus on avance plus on translate la rangée en traitement de ($- \text{module}$) vers le bas.

3^{ème} étape :

Une fois la liste « sélection » traitée et placée à son niveau correspondant, on ôte de la liste des 3dfaces non traités ceux qui ont été traités (donc la liste sélection).

4^{ème} étape :

Il s'agit de transformer la liste des 3df de la « sélection » en une liste de segments.

Et on procède ainsi récursivement, jusqu'à ce que tous les 3d faces qui forment le tracé soient traités. On obtiendra à la fin le modèle tridimensionnel de la voûte que représente le tracé régulateur dessiné à l'écran.

6 Chapitre 6 : Expérimentation et discussion

Dans cette partie, nous allons expérimenter la méthode proposée dans le chapitre précédant sur 2 tracés régulateurs différents de voûtes mouqarnas en bois.

Deux méthodes de traitement différentes vont être testées sur ces 2 tracés afin de déterminer la pertinence de chacune d'elles.

6.1 Présentation des méthodes :

6.1.1 Méthode de traitement globale des tracés :

En adoptant cette méthode, l'utilisateur laisse la tâche de traitement d'un tracé régulateur au programme informatique proposé. C'est ce dernier qui se chargera des différentes étapes nécessaires pour aboutir, à partir du tracé régulateur, au modèle tridimensionnel de la voûte mouqarnas. L'utilisateur n'intervient que pour sélectionner le tracé, préciser sa nature et, si nécessaire, sélectionner les sommets intérieurs dans le cas où les pièces se réunissent autour d'une coupole étoilée.

6.1.2 Méthode de traitement couche par couche des tracés:

Cette méthode laisse plus d'autonomie à l'utilisateur dans le traitement du tracé régulateur. Le traitement informatique doit être précédé par une lecture ainsi qu'une analyse du tracé régulateur à fin de délimiter les différentes rangées et de les séparer lors du traitement et de la construction tridimensionnelle des pièces. Chacune sera traitée séparément et indépendamment des autres pour être ensuite placée à la hauteur appropriée.

6.2 La méthode de traitement globale

6.2.1 1^{er} cas d'étude :

Le premier cas sur lequel nous allons expérimenter la méthode de traitement proposée est un tracé régulateur mouqarnas, représentant une partie de la coupole octogonale en bois à la « Kasba » de « Talouet » au Maroc¹¹ (figure 51).

À l'aide du logiciel de traitement choisi, on procède à la représentation du tracé à l'écran (figure 52).

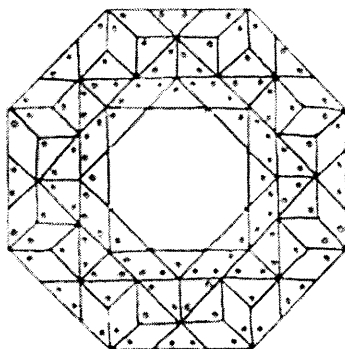


Figure 52 Tracé original

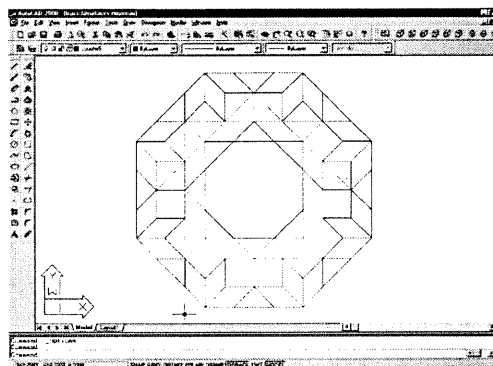


Figure 53 Représentation informatique du tracé

6.2.1.1 Initialisation du traitement : Questions/réponses

En initialisant la procédure « mouqarnas » et après avoir sélectionné le tracé qu'on veut étudier, l'utilisateur doit préciser les trois paramètres suivant :

La nature du tracé à l'écran : dans notre cas c'est un tracé où les pièces ne commencent pas au centre.

Select objects:	
80 entitees choisie(s)	
Est-ce que les pièces commencent à partir du centre du tracé O/N ?	
696.8864, -626.4443, 0.0000	SNAP GRID ORTHO POLAR OSNAP OTRACK LWT MODEL

¹¹ « Arabesques : Art décoratif au Maroc » Jean-Marc Castera Page 382

Le nombre de sommets dans la coupole étoilés : huit sommets dans notre cas.

80 entitees choisie(s)	
Est-ce que les pièces commencent à partir du centre du tracé O/N ?n	
combien de points existent dans la périphérie interieur?:	
.572.4485, -527.3916, 0.0000	SNAP GRID ORTHO POLAR OSNAP OTRACK LWT MODEL

Sélectionner par ordre tous les sommets de la coupole étoilée :

Est-ce que les pièces commencent à partir du centre du tracé O/N ?n	
combien de points existent dans la périphérie interieur?:8	
selectionner les sommets par ordre:	
.572.4485, -527.3916, 0.0000	SNAP GRID ORTHO POLAR OSNAP OTRACK LWT MODEL

6.2.1.2 *Le traitement de la première couche :*

L'implication de l'utilisateur s'achève à ce niveau et c'est le programme informatique qui se charge des étapes suivantes :

- Transformer la liste des sommets de la coupole étoilée en une liste de segments (lseg-f).
- Déterminer les 3d faces qui ont les 2 points d'un des segments de la liste (lseg-f) dans leur liste de sommets.
- Construire le modèle tridimensionnel de la première rangée.

La liste (selection) sera traitée par la procédure « tri-quadri » qui, individuellement pour chacun de ses éléments, va déterminer la nature de sa forme

Le premier élément de cette liste est composé de trois éléments, et donc de 3 sommets : c'est un triangle.

Ce 3df est ensuite traité par la procédure « traite-triangle » qui va déterminer la nature de ce triangle : En calculant la distance qui sépare chacun des sommets par rapport au centre du tracé, on trouve que seul un des sommets est le plus éloigné et l'angle en ce point est 90^0 , donc c'est la pièce « Dembouq ».

Ce 3df est ensuite traité par la procédure « dembouq » qui va construire le modèle de la pièce en 3 dimensions et le placer à sa hauteur correspondante qui est dans

ce cas la hauteur ($h = 0$) car la première rangée est située au niveau même du tracé.

C'est la même pièce qui forme toute la première rangée donc le même principe de construction sera appliqué pour tous les 3d faces de la liste « sélection ».

6.2.1.3 Le traitement des couches suivantes :

Une fois la première rangée traitée, la procédure « traite-trace » parcourt la liste des 3d faces non-traités pour déterminer ceux qui appartiennent à la deuxième rangée. Les éléments qui appartiennent à la deuxième rangée sont ceux qui ont, au moins, un segment en commun avec les 3d faces qui forment la première.

La première étape consiste à transformer la liste (selection) des 3df qui constituent la première rangée en une liste de segments. En sélectionnant un segment à la fois, on cherche dans la liste des éléments non traités ceux qui ont les deux points du segment dans leur liste de sommets (ce qui revient à dire que le segment en traitement constitue une des ses arêtes). On obtient à la fin la liste des 3df constituant la deuxième rangée qu'on appellera (selection).

Il est à remarqué que la liste (selection) représentera toujours la liste des 3df qui vont être construits en 3 dimensions et qui vont déterminer les éléments qui appartiennent à la rangée suivante.

La liste (selection) sera traitée par la procédure « tri-quadri » qui, pour chacun de ses éléments, va déterminer la nature de sa forme.

Le premier élément de cette liste est composé de quatre éléments, et donc de 4 sommets : c'est un quadrilatère.

Ce 3df est ensuite traité par la procédure « traite-quadrilatère » : En calculant les longueurs des arêtes de ce 3df, on constate que les quatre sont égaux et de longueur 141.42, c'est donc un losange¹².

Ce 3df est ensuite traité par la fonction « losange1-3 » qui va déterminer la nature exacte de la pièce : En calculant la distance qui sépare chacun de sommets du centre du tracé on trouve que seul un des sommets est le plus éloigné et l'angle en ce point est 135° donc c'est la pièce " charbiya «.

Cette liste est ensuite traitée par la fonction « charbiya » qui va construire le modèle de la pièce en 3 dimensions et le placé à sa hauteur correspondante qui est dans ce cas l'hauteur -100 (ou -module car le module choisi par défaut dans la représentation du tracé est égal à 100). La deuxième rangé est situé à un niveau inférieur de (- module) par rapport à celle qui la précède.

C'est la même pièce qui forme toute la deuxième rangée donc ça sera le même principe de construction pour tous les 3d faces de la liste « sélection ».

Le traitement du tracé continue ainsi jusqu'à ce que tous les éléments qui forment le tracé régulateur soient traités On obtient ainsi le modèle numérique tridimensionnel de la voûte mouqarnas. (Figure 54)



Figure54 Modèle numérique de la voûte

6.2.2 2^{ème} cas d'étude :

Le deuxième cas sur lequel nous allons tester nos procédures représente une partie du tracé régulateur mouqarnas utilisé par Jean-Marc Castéra ¹³ pour illustrer la partie « Dessiner à main levée. » (figure 55).

À l'aide du logiciel de traitement choisi, on procède à la représentation du tracé à l'écran (figure 55).

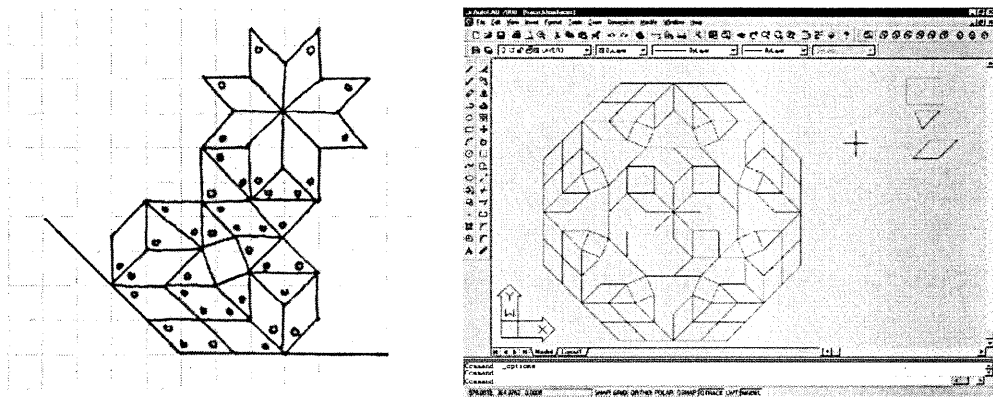


Figure 55 Tracé régulateur mouqarnas et sa représentation à l'écran

6.2.2.1 Initialisation du traitement : Questions/réponses

En initialisant la procédure « mouqarnas » et après avoir sélectionné le tracé qu'on veut étudier, on doit préciser la nature du tracé à l'écran : dans notre cas c'est un tracé où les pièces commencent au centre.

```

Select objects:
80 entitees choisie(s)
Est-ce que les pièces commencent à partir du centre du tracé O/N ?
696.8864, -626.4443, 0.0000 SNAP GRID ORTHO POLAR OSNAP OTRACK LWT MODEL

```

¹³ « Arabesques : Art décoratif au Maroc » Page 332

6.2.2.2 *Le traitement de la première couche :*

Le rôle de l'utilisateur s'achève à ce niveau et c'est le programme informatique qui se charge du reste selon les étapes suivantes :

- Trouver les 3d faces qui ont le centre du tracé dans leur liste de sommets.
- Construire le modèle tridimensionnel de la première rangée.

Le premier 3df sera traité par la procédure « tri-quadri » pour déterminer la nature de sa forme. C'est une liste à 4 éléments, et c'est donc un quadrilatère.

Ce 3df est ensuite traité par la procédure « traite-quadrilatère » qui, en calculant les arrêtes de ce 3df, trouve que les quatre sont égaux et équivalent à 141.42 : c'est donc un losange¹⁴.

Ce 3df est ensuite traité par la procédure « losange1-3 » qui va déterminer la nature exacte de la pièce. En calculant la distance qui sépare chacun des sommets par rapport au centre du tracé, on trouve que seul un des sommets est le plus éloigné et l'angle en ce point est de 90^0 : c'est donc la pièce "Ktaf" .

Cette liste est ensuite traitée par la procédure « ktaf » qui va construire le modèle de la pièce en 3 dimensions et la placer à sa hauteur correspondante

C'est la même pièce qui forme toute la première rangée donc ça sera le même principe de construction pour tous les 3d faces de la liste (sélection).

6.2.2.3 *Le traitement des rangées suivantes*

Une fois la première rangée traitée, la procédure « traite-trace » parcourt la liste des 3d faces non-traités pour déterminer ceux qui appartiennent à la deuxième rangée. Les éléments qui appartiennent à la deuxième rangée sont ceux qui ont, au moins, un segment en commun avec les 3df qui forment la rangée précédente.

La première étape consiste à transformer la liste (selection) des 3df qui constituent la première rangée en une liste de segments. Ensuite, en sélectionnant un segment à la fois, on cherche dans la liste des éléments non traités ceux qui ont

¹⁴ d'après la définition géométrique du losange régulier dans la partie géométrie page 33

les deux points du segment dans leur liste de sommets (ce qui revient à dire que le segment en traitement constitue une des ses arêtes). On aura à la fin la liste des 3df constituants la deuxième rangé qu'on appellera (selection).

Le traitement du tracé continue ainsi jusqu'à ce que tous les éléments qui forment le tracé régulateur soient traités. On aura ainsi le modèle tridimensionnel de la voûte.(Figure 56)

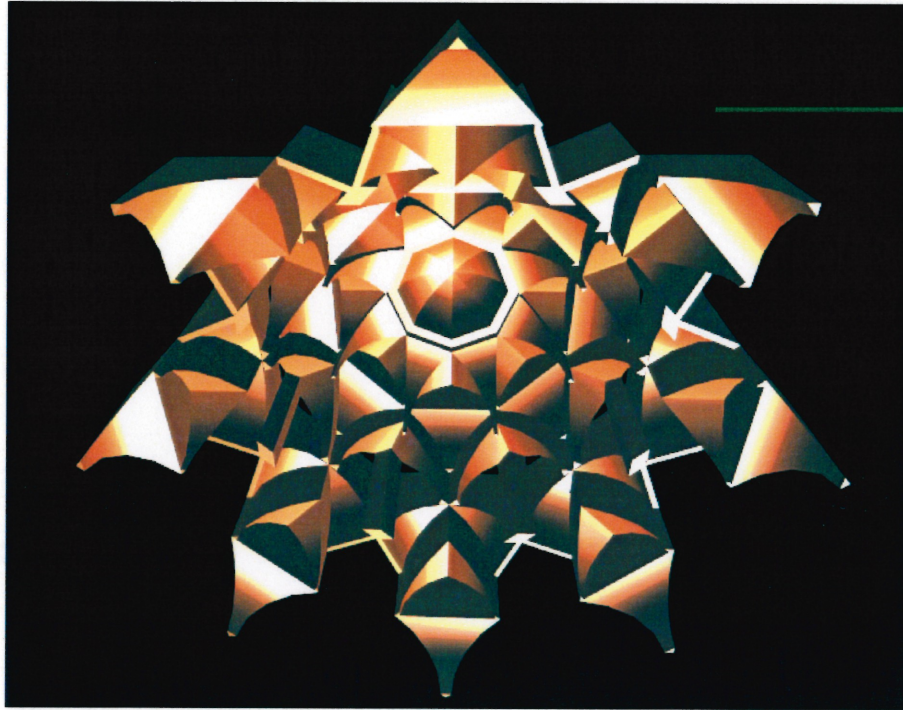


Figure56 Modèle numérique de la voûte

6.3 La méthode de traitement couche par couche :

Cette méthode nécessite une étude préliminaire du tracé avant sa représentation à l'écran. En effet, l'utilisateur doit distinguer les différentes rangées qui constituent le tracé régulateur en traitement, ce qui revient à effectuer manuellement la tâche des procédures « traite-trace » et « sniwa ».

6.3.1 Le traitement préliminaire :

1. La distinction en couches :

Les 3d faces qui appartiennent à la première rangée sont ceux qui ont :

- Au minimum, une de leurs arrêtes en commun avec la coupole étoilée au centre dans le cas du tracé où les pièces du tracé se réunissent autour d'une coupole étoilée.
- Le centre du tracé dans leur liste de sommets dans le cas des tracés qui commencent à partir du centre.

Une fois cette rangée déterminée, on la représente à l'écran en prenant bien soin de la mettre sous une couche (layer) qui lui est propre.

Les 3d faces qui appartiennent à la rangée suivante sont ceux qui ont une des arrêtes qui les constituent en commun avec un ou plusieurs éléments de la rangée précédente. Une fois cette rangée déterminée, on la représente à l'écran en prenant bien soin de la mettre sous une couche (layer) qui lui est propre.

En adoptant le même principe, on détermine les différentes rangées qui constituent le tracé régulateur et on les représente à l'écran sous différentes couches pour pouvoir les séparer au moment du traitement.

2. Le choix des pièces :

Contrairement à la méthode précédente, l'ambiguïté qui accompagne la reconnaissance des pièces « dembouq » et « chiira » d'une part et des pièces « t'stiya maftouha » et « t'stiya masdouha » d'autre part peut être évitée du

moment que chacune des rangées est traitée séparément. On pourra préciser de quelle pièce il s'agit par une simple lecture du tracé régulateur d'origine.

6.3.2 Principe du traitement couche par couche :

Chaque couche est traitée suivant le scénario suivant:

- Initialiser la procédure « mouqarnas ».
- Sélectionner les 3d faces qui forment la couche en traitement.
- Préciser la nature du tracé (tracé où les pièces ne commencent pas par le centre).
- Préciser le nombre de sommets intérieurs.

Ce traitement est répété pour chacune des rangées en précisant dans chaque cas la hauteur appropriée.

Le traitement couche par couche du tracé où les pièces se réunissent autour d'une coupole étoilée

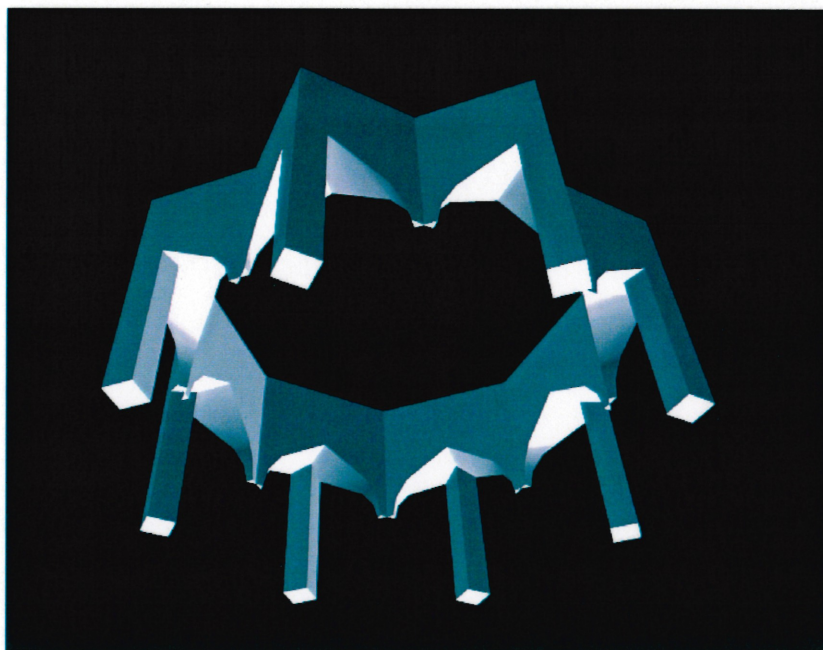


Figure57 Première couche

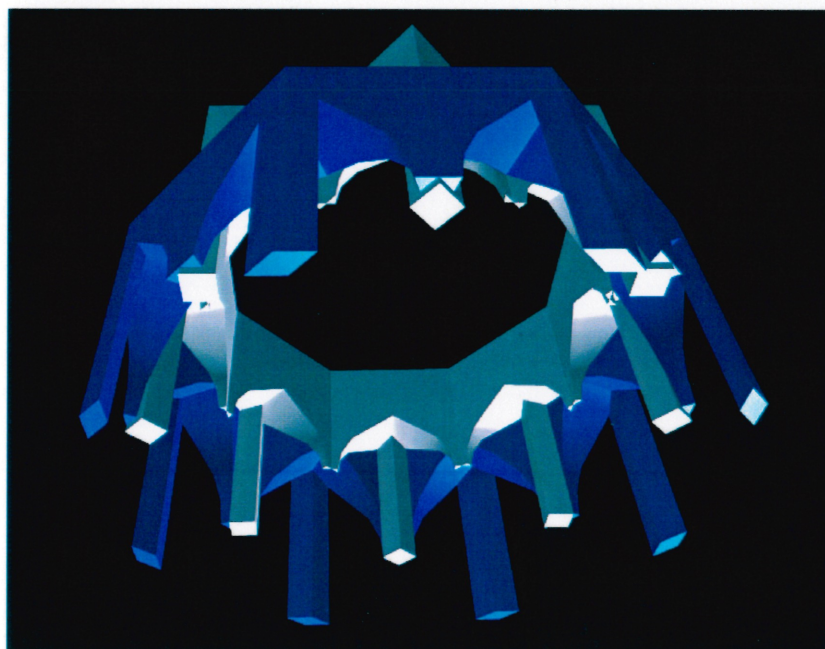


Figure58 Deuxième couche

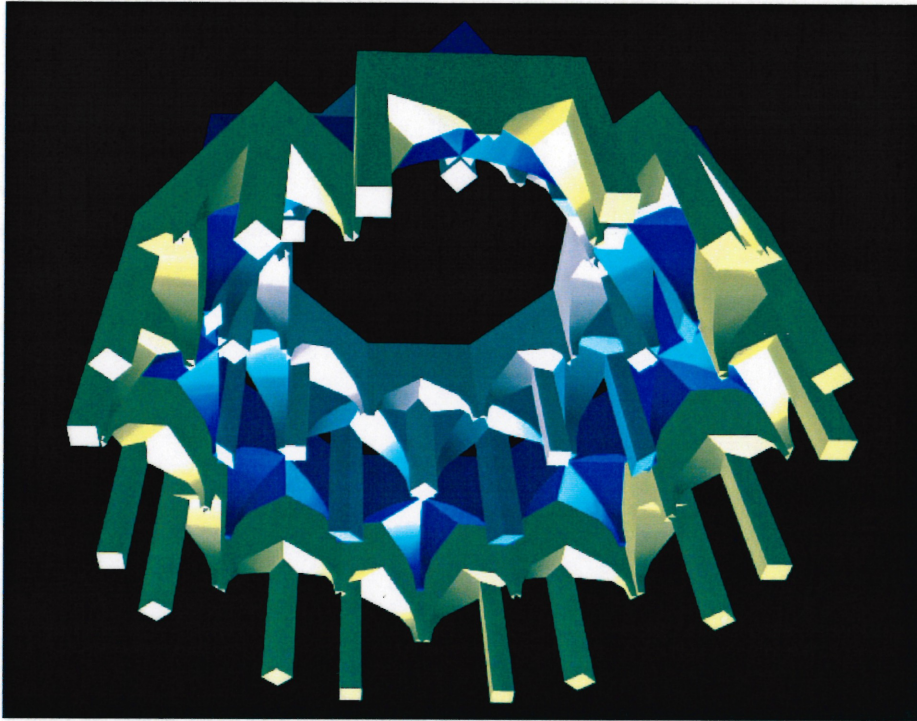


Figure59 Troisième couche

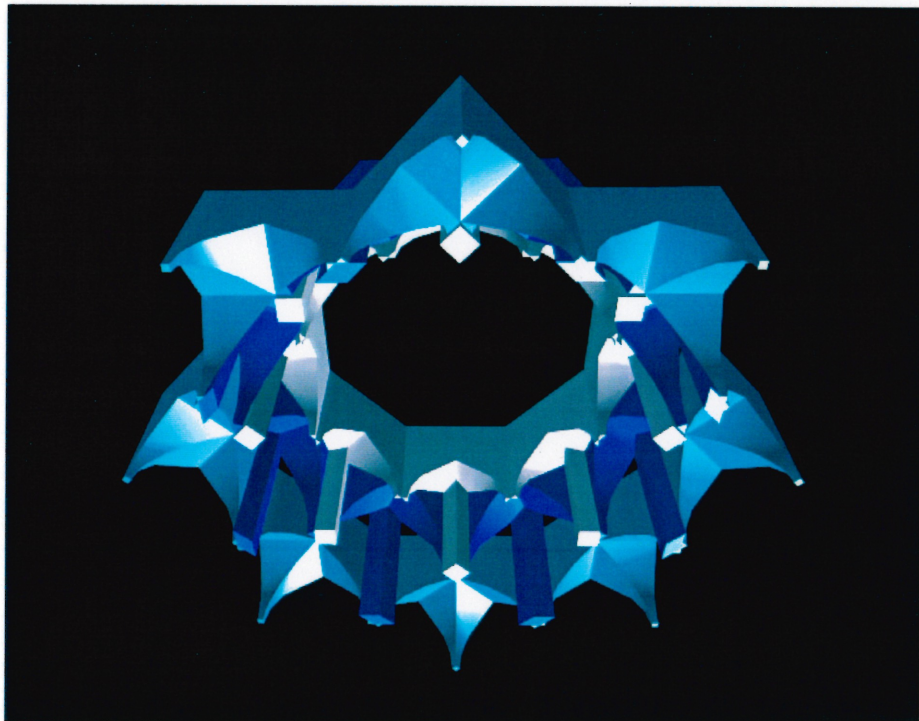


Figure60 Quatrième couche

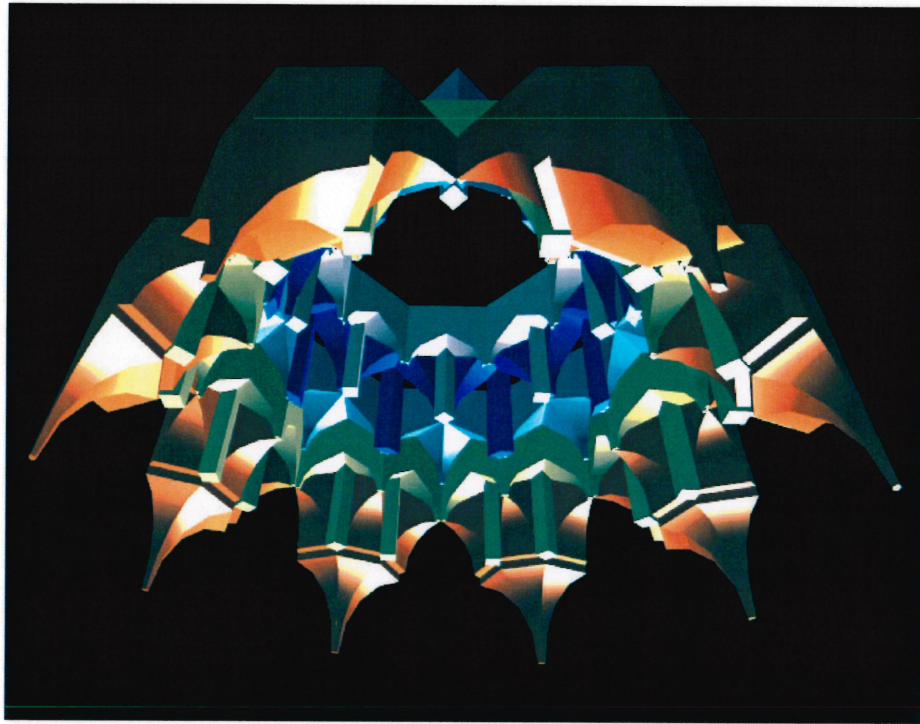


Figure61 Cinquième couche

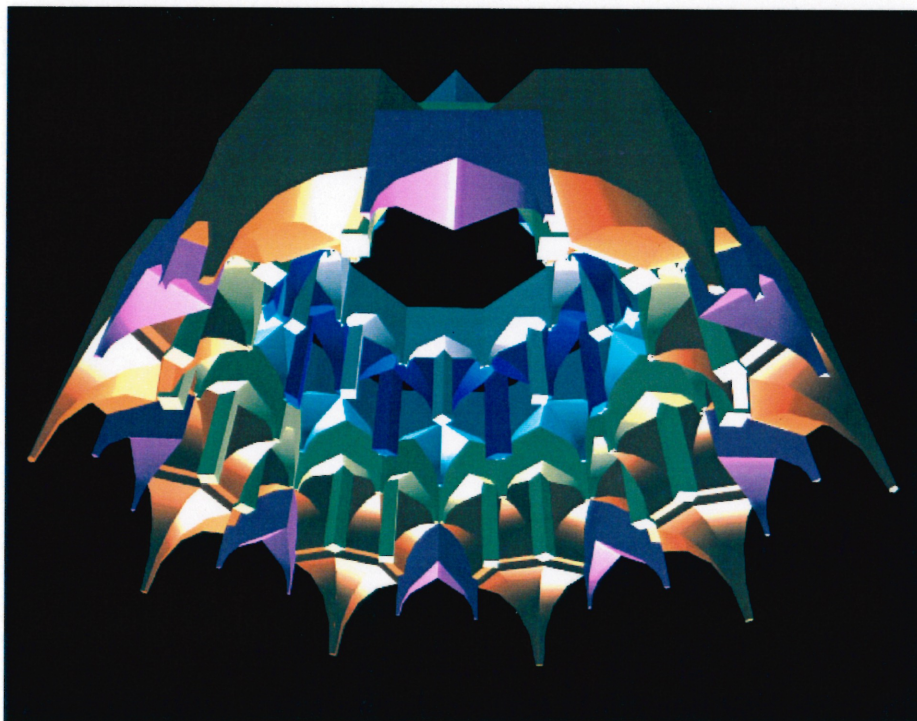
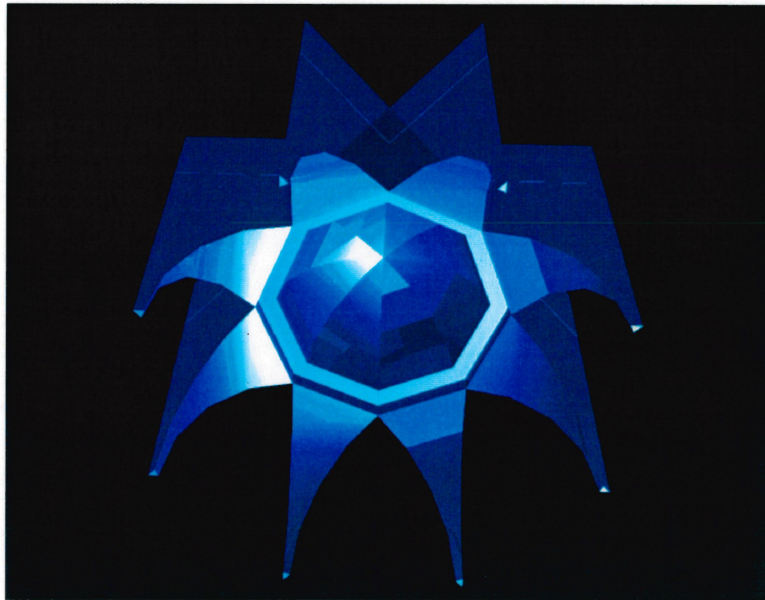
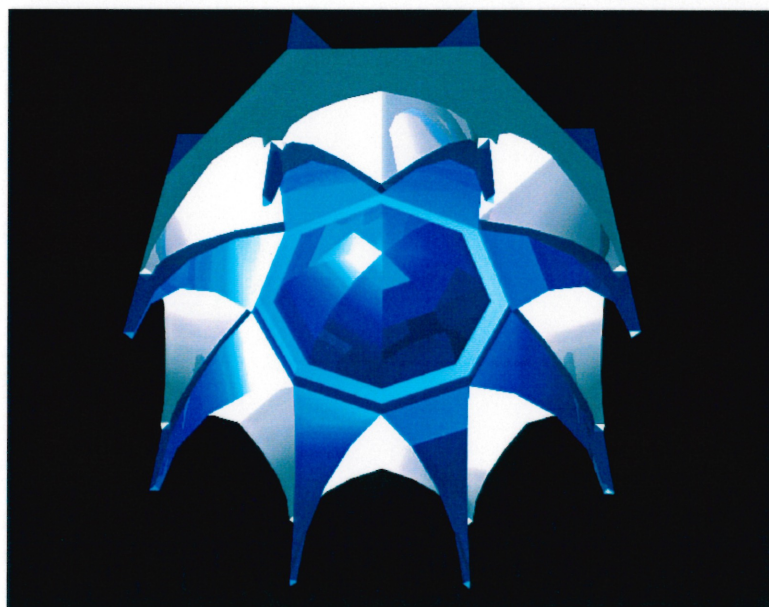


Figure62 Sixième couche

Le traitement couche par couche du tracé où les pièces commencent au centre



Première rangée



Deuxième rangée



Figure63 Troisième rangée

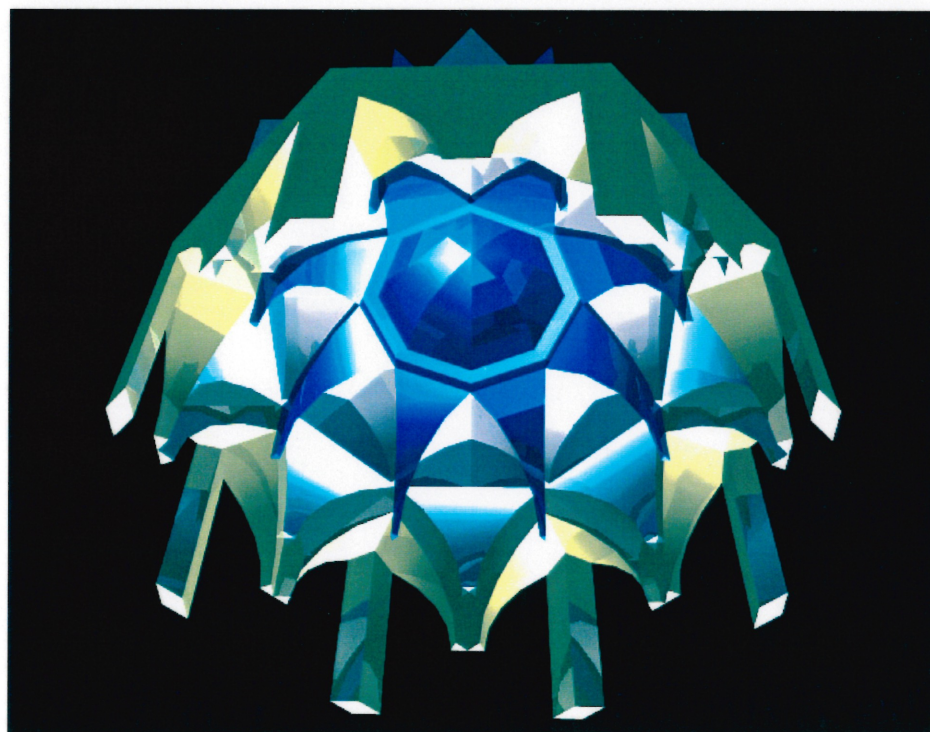


Figure64 Quatrième rangée

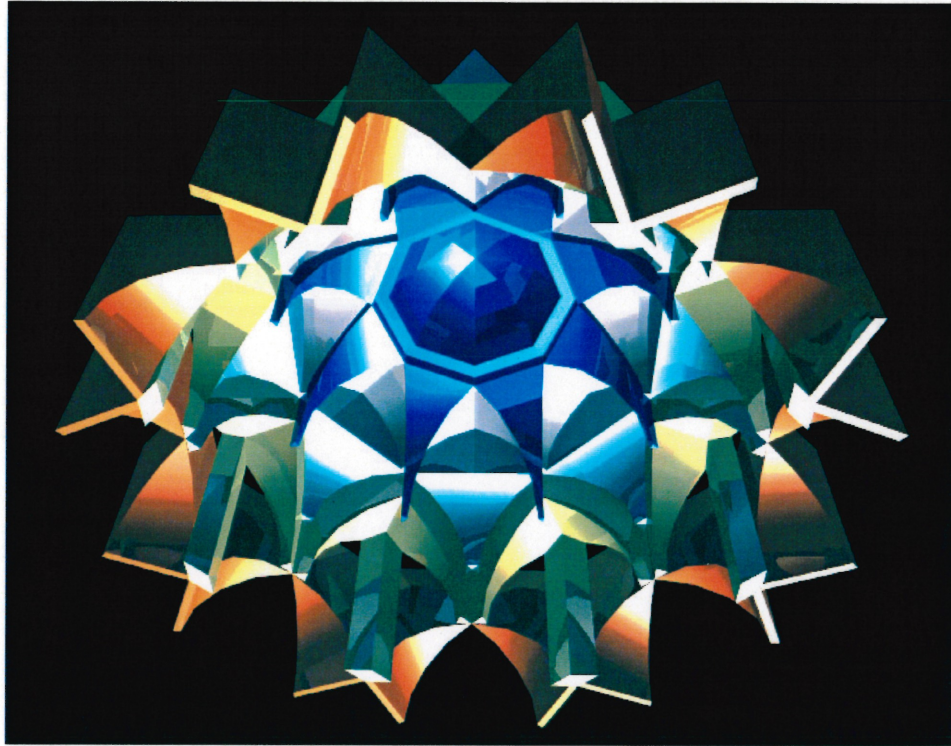


Figure65 Cinquième rangée



Figure66 Sixième rangée

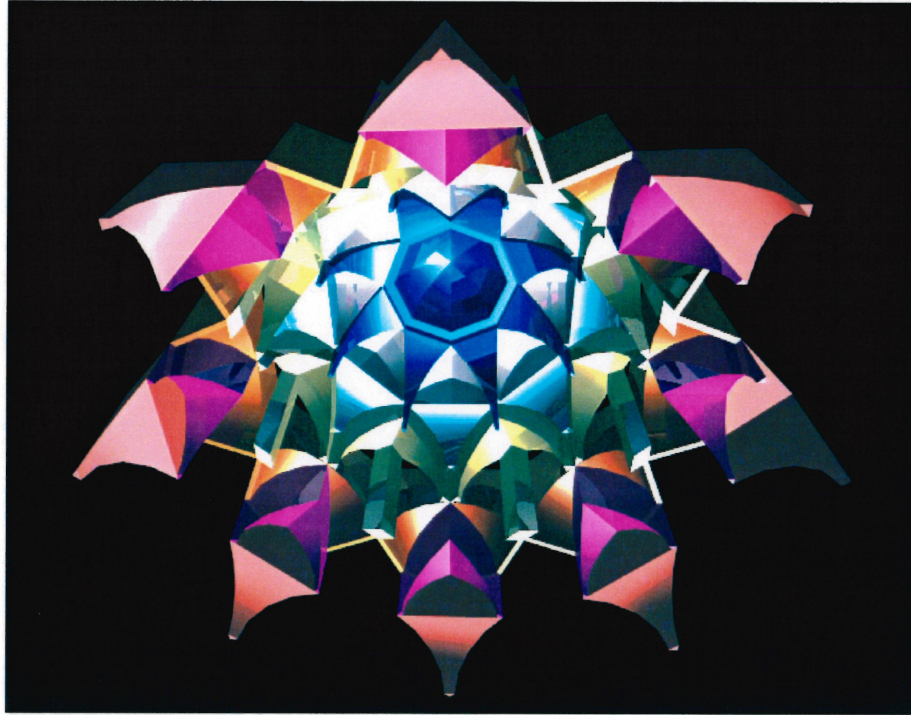


Figure67 Septième rangée

6.4.3 Conclusion du chapitre :

Dans ce chapitre, 2 modes de traitement informatique ont été testés afin d'arriver, en partant d'un tracé régulateur des mouqarnas en bois, au modèle numérique tridimensionnel de la voûte qui le représente. Le but dans chacune des deux méthodes est le même mais le traitement est différent.

Six critères de comparaison ont été adoptés pour dégager les avantages et les inconvénients de chacune d'elles :

6.4.4 Intervention de l'utilisateur :

Pour la méthode de traitement global, l'utilisateur intervient une seule fois au début des opérations et c'est le programme informatique qui se charge de toutes les étapes ultérieures de sélection du tracé.

Pour la méthode de traitement couche par couche, l'utilisateur a beaucoup plus de contrôle car il précise pour chaque rangée les éléments qui la constituent, la nature de ces éléments et la hauteur qu'il lui attribue.

L'utilisateur a plus de contrôle sur les étapes de traitement du tracé dans la méthode de traitement couche par couche que la méthode de traitement globale.

6.3.3.2 Connaissance requise de la géométrie des mouqarnas :

Pour la méthode de traitement global, le rôle de l'utilisateur ne dépasse pas l'étape de représentation du tracé.

Pour la méthode de traitement couche par couche, l'utilisateur doit être capable, en plus de la représentation du tracé, de séparer chacune des rangées qui le composent.

6.3.3.3 Ambiguïté de reconnaissance des pièces :

La description géométrique des pièces « dembouq » et « chiira » d'une part et des pièces « t'stiya maftouha » et « t'stiya masdouda » d'autre part est la même.

Pour la méthode de traitement global, on précise au début du traitement, la pièce que l'on va retenir et même si les deux pièces existent dans le tracé, le programme va les interpréter comme étant la même.

La méthode de traitement couche par couche nous permet d'avoir plus de contrôle sur ce point. En fait, pour chaque couche et par une simple lecture du plan d'origine, on revient sur les procédures pour préciser la nature de la pièce que le programme informatique doit reconnaître et dessiner en 3 dimensions.

6.3.3.4 Contrôle des éléments d'une rangée :

Pour la méthode de traitement global, ce sont les procédures « traite-trace » et « sniwa » qui déterminent les éléments des différentes rangées qui forment le tracé.

Pour la méthode de traitement couche par couche, c'est l'utilisateur qui détermine les différentes rangées et les éléments qui les constituent.

Contrôler les niveaux peut s'avérer de grande utilité dans le traitement des tracés avec des coupes étoilées qui ne sont pas situés au centre car ils n'obéissent pas aux mêmes lois de division des rangées que les autres types de tracés régulateurs des mouqarnas en bois.

6.3.3.5 Vitesse du traitement :

La méthode de traitement global est beaucoup plus rapide que la méthode de traitement couche par couche car la première nécessite peu de travail préliminaire tandis que la deuxième nécessite un travail préliminaire en plus du traitement de chaque couche indépendamment les unes des autres.

	Méthode de traitement globale	Méthode de traitement couche par couche
Intervention de l'utilisateur	Non	Oui
Connaissance requise de la géométrie des mouqarnas	Non	Oui
Ambiguïté de reconnaissance des pièces	Oui	Non
Contrôle des éléments d'une rangée	Non	Oui
Vitesse de traitement	Rapide	Lente

7 Chapitre 7 : Conclusion générale

Au terme ce travail, nous sommes parvenus à proposer un nouvel algorithme de traitement permettant de générer, à partir d'un tracé régulateur de voûte mouqarnas modulaire en bois, le modèle tridimensionnel correspondant.

Nous avons expérimenté cette méthode sur deux types de tracé régulateur mouqarnas. Cette expérimentation démontre que la méthode de traitement proposée permet :

- Une description algorithmique des lois géométriques qui régissent les tracés régulateurs.
- L'identification de la majorité des éléments architecturaux qui composent ces tracés.
- Le traitement des tracés régulateurs mouqarnas à l'aide de deux méthodes différentes:
 - La méthode de traitement globale ou automatique.
 - La méthode de traitement couche par couche ou avec contrôle étape par étape.
- La génération des modèles tridimensionnels de certains types de voûtes mouqarnas.

La méthode de traitement proposée ne permet pas :

- La différenciation entre les représentations bidimensionnelle des pièces qui ont exactement la même description géométrique. (exemple des pièces « dembouq » et « chiira » d'une part et des pièces « t'stiya maftouha » et « t'stiya masdouda » d'autre part)
- En utilisant la méthode de traitement global, les niveaux sont automatiquement définis et donc on peut avoir des erreurs de lecture

7.1 Autres applications possibles :

La méthode de traitement proposée permet de traiter, en plus des tracés régulateurs des dômes mouqarnas en bois, une grande partie des tracés des arcades, des encorbellements et des frises. En fait, le principe de la reconnaissance des projections planes des pièces et la logique de traitement des couches sont, dans plusieurs cas, les mêmes. Donc il est possible de leur appliquer le même algorithme de traitement que les tracés des voûtes mouqarnas afin de générer leur modèle tridimensionnel.

Cette méthode de traitement peut être aussi d'une grande utilité dans les études archéologiques se rapportant à des structures mouqarnas qui ont été détruites et sur lesquels on a très peu d'informations. Elle sera de grand aide dans l'étude et la proposition des modèles possibles de ces structures.

7.2 Travail à compléter:

7.2.1 Généralisation du traitement :

La méthode de traitement proposée pourrait être généralisée pour tous les tracés régulateurs mouqarnas. En fait, les mouqarnas modulaires en bois représentent le jeu de construction le plus compliqué par rapport aux autres car l'alphabet des pièces qui le constituent est beaucoup plus large et leur mode de composition est beaucoup plus élaboré. Le modèle algorithmique proposé peut servir de base à d'autres études visant à généraliser le traitement sur tous les autres structures mouqarnas existantes.

7.2.2 Étude des tracés à base de carré-losange :

Dans son livre « Arabesque : arts décoratifs au Maroc », Jean-Marc Castéra a écrit :

« ...Il n'y a pas que des carrés et des losanges dans les plans de mouqarnas, mais les autres formes peuvent toujours se ramener à une décomposition de ces carrés et de ces losanges..(Voir figure 37 p : 47). S'il est clair que toute structure de mouqarnas s'inscrit dans un pavage

carrés-losanges, la réciproque est-elle vraie ? Tout pavage carrés-losanges peut-il être transformé en structure de mouqarnas cohérente ?»

Cette remarque peut conduire à réfléchir les différents moyens permettant le passage 2d/2d entre un tracé régulateur carrés-losanges et les tracés régulateurs mouqarnas qui lui sont associés. On pourra commencer à réfléchir sur les questions suivantes :

- Est-ce qu'il existe des règles géométriques permettant le passage d'un tracé à base de carré-losange à un tracé mouqarnas ?
- Si ces règles existent, combien de tracés régulateurs mouqarnas peut-on obtenir à partir d'un seul tracé à base de carré-losange ?

En réfléchissant sur ces questions, on pourra enfin créer et inventer une infinité de nouveaux tracés régulateurs mouqarnas.

BIBLIOGRAPHIE

- TABBAA, Yasser (1985), "The Muqarnas Dome: Its Origin And Meaning",
 Dans "*Muqarnas : an annual on Islamic art and architecture*", volume 3,
 Edited by Oleg Grabar, Yale University Press, New Haven, p. 61-74.
- NECIPOGLU, Gülru (1995), The Topkapi Scroll : Geometry And Ornament
 InIslamic Architecture : The Getty center for the history of art and the
 humanities, Santa Monica.
- AL-ASAD, Mohammad (1995), The Mouqarnas: A Geometric Analysis
 Dans "*The Topkapi scroll : geometry and ornament in Islamic
 architecture*", p. 349-359.
- CASTERA, Jean-Marc (1996) Arabesques Art Décoratif Au Maroc
 Arc édition internationale, Coubevoie (Paris) 480p.
- CASTERA, , Jean-Marc (1991) Géométrie Douce : Art Géométrique
 Arabo-Andalou, Mode D'emploi / Jean-Marc Castera, HéliŠne Jolis,
 Atelier 6 1/2, Paris.
- CRITCHLOW, Keith (1976) Islamic Patterns : An Analytical And
 Cosmological Approach, Thames and Hudson, London.
- GRABAR, Oleg (1987) La Formation De L'art Islamique
 Flammarion, Paris p : 259, 296
- GRABAR, Oleg (1996), L'ornement : Formes Et Fonctions Dans L'art
 Islamique, Paris : Flammarion p : 71-86.

- PACCARD, André (1983) Le Maroc Et l'artisanat Traditionnel Islamique
Dans l'architecture 4e ed. Atelier 74, Annecy Tome 1, p: 288-313
- YVONNE, Dold-Samplonius (1992) Practical Arabic Mathematics: Measuring
The Mouqarnas By Al-Kashi Dans "*Centaurus: International magazine*
of the history of mathematics, science and technology", p.:193-239
- SHIRO, Takahashi Muqarnas : A Three-Dimensional Decoration Of Islam
Architecture, <http://www.babot.co.jp/shiro/muqarnas/default0.htm>
- YVONNE, Dold-Samplonius., TOBIAS I., REICHERT C., SÄTZLER K.
 (1998), Muqarnas Visualization At The IWR
<http://www.iwr.uni-heidelberg.de/groups/ngg/Muqarnas/>
- SAKKAL, Mamoun (1988), An Introduction To Muqarnas Domes Geometry
Dans Structural Topology #14 , p:21-34
- SAKKAL, Mamoun (1982), Geometry Of Mouqarnas In Islamic Architecture
 M. Arch. Thesis, University of Washington
- ODEKON, A. (1977) Osmani Oncesi Anadolu Turk Mimarisinde Mukurnasli
Porta Ortuleri, Istanbul
- ZEITOUN, Jean (1977), Trames Planes: Introduction À Une Étude
De Architecturale Des Trames BORDAS, Paris
- MITCHELL, William et McCullough M. (1991), Digital Design Media,
 New York, VNR
- MITCHELL, William (1990), The Logic of Architecture, Cambridge, MIT Press

MITCHELL, William (1986), A Foundation for Computer-aided Design.

Dans : *Environment and Planning B : Planning and Design* 1986,
volume 13, p.135-157.

COUWENBERGH Jean-Pierre (1998) L'intégrale de Autocad 14,

Marabout, Alleur (Belgique).

ECOCHARD, Michel (1978), Filiation des monuments Grecs, Byzantins

et islamiques : une question de géométrie . P. Geuthner, Paris 1978

TERRASSE, Henri (1925), Les arts décoratifs au Maroc, J. Hainaut.

TITUS, Burckhardt (1985), L'art de l'islam, langage et signification,

Sindbad, Paris.

Annexes

Index des fonctions

Fonction	Page
Mouqarnas.....	ii
Selection3df.....	vi
Traite-trace.....	viii
Sommetint.....	x
Tri-quadri.....	xiii
Traite-triangle.....	xiv
Traitequadrilatère.....	xix
Sniwa.....	xxii
Loza1-2.....	xxv
Losange.....	xxviii
Rectangle.....	xxxii
Dembouq.....	xxxvi
Serwaliya.....	xxxviii
Chiira.....	xlii
Loza1.....	xlvi
Laza2.....	xlix
Ktaf.....	li
Charbiya.....	liii
Testiya-m.....	lvii
Testiya-o.....	lix

3df2seg.....	lxiii
Angle-triangle.....	lxv
Angle3df.....	lxvi
Dif-liste.....	lxvii
Elimine-double.....	lxviii
Mil2d.....	lxx
Mil3d.....	lxxi
Nbr-max.....	lxxii
Nbr-max2.....	lxxiii
Proche-zero.....	lxxiv
Pt-centre.....	lxxv
Ptloin3.....	lxxvi
Ptloin4.....	lxxviii
Tliste.....	lxxx
Trouve.....	lxxxiii
z-off.....	lxxxv

Annexe 1:

Construction de la pièce “dembouq”.....page lxxxvii

Annexe 2:

Construction de la pièce “loza1”.....page lxxxviii

Annexe 3:

Construction de la pièce “ktaf”.....page lxxxix

Annexe 4:

Construction de la pièce “serwaliya”.....page xci

Annexe 5:

Construction de la pièce “charbiya”.....page xciii

Annexe 6:

Construction de la pièce “serwaliya-sguira”.....page xcv

```

;*****
;*  M. Sc. Aménagement [CMFAO] *
;*  Anis Semlali      [GRCAO] 2000 *
;*****

; Fonction « mouqarnas »

; Fonction principale de traitement

;-----

; Fonctions appelées

; « elimine-double »

; « 3df2seg »

; « sniwa »

;-----

; Variables locales

; « temp, y, lseg-f »

;-----

; Retour

; l-fin: liste de tous les 3df qui constituent la rangé à traiter

;-----

; Fonctions de traitement

      (load "D:/anis/mouqarnas/fonction de traitement/selection3df.LSP")

      (load "d:/anis/mouqarnas/fonction de traitement/tri-quadri.LSP")

      (load "d:/anis/mouqarnas/fonction de traitement/traite-quadrilatere.LSP")

      (load "d:/anis/mouqarnas/fonction de traitement/traite-trace.LSP")

      (load "d:/anis/mouqarnas/fonction de traitement/rectangle.LSP")

```

(load "d:/anis/mouqarnas/fonction de traitement/losange1-3.LSP")

(load "d:/anis/mouqarnas/fonction de traitement/loza1-2.LSP")

(load "d:/anis/mouqarnas/fonction de traitement/test-niveau.LSP")

(load "d:/anis/mouqarnas/fonction de traitement/sommetint.LSP")

; Fonctions de construction des pièces

(load "d:/anis/mouqarnas/piece3D/dembouq.LSP")

(load "d:/anis/mouqarnas/piece3D/serwaliya.LSP")

(load "d:/anis/mouqarnas/piece3D/chiira.LSP")

(load "d:/anis/mouqarnas/piece3D/loza1.LSP")

(load "d:/anis/mouqarnas/piece3D/loza2.LSP")

(load "d:/anis/mouqarnas/piece3D/ktaf.LSP")

(load "d:/anis/mouqarnas/piece3D/charbiya.LSP")

(load "d:/anis/mouqarnas/piece3D/testiya-m.LSP")

(load "d:/anis/mouqarnas/piece3D/testiya-o.LSP")

; Sous-routines

(load "d:/anis/mouqarnas/sous-routines/3df2seg.LSP")

(load "d:/anis/mouqarnas/sous-routines/dif-liste.LSP")

(load "d:/anis/mouqarnas/sous-routines/corrige-angle.LSP")

(load "d:/anis/mouqarnas/sous-routines/elimine-double.LSP")

(load "d:/anis/mouqarnas/sous-routines/angle-triangle.LSP")

(load "d:/anis/mouqarnas/sous-routines/corrige-angle2.LSP")

(load "d:/anis/mouqarnas/sous-routines/pt-centre.LSP")

(load "d:/anis/mouqarnas/sous-routines/nbr-max.LSP")

(load "d:/anis/mouqarnas/sous-routines/nbr-max2.LSP")


```
(load "d:/anis/mouqarnas/sous-routines/z-off.LSP")  
(load "d:/anis/mouqarnas/sous-routines/proche-zero.LSP")  
(load "d:/anis/mouqarnas/sous-routines/ptloin3.LSP")  
(load "d:/anis/mouqarnas/sous-routines/ptloin4.LSP")  
(load "d:/anis/mouqarnas/sous-routines/tliste.LSP")  
(load "d:/anis/mouqarnas/sous-routines/mil2d.LSP")  
(load "d:/anis/mouqarnas/sous-routines/mil3d.LSP")  
(load "d:/anis/mouqarnas/sous-routines/trouve.LSP")
```

```
;-----
```

```
(defun mouqarnas ()  
  (setq liste (selection3df))  
  (setq repl (getstring "Est-ce que les pièces commencent à partir du centre du  
                        tracé O/N ?"))  
  (if  
    (= repl "o")  
    (and  
      (progn  
        (setq h 0)  
        (setq selection (trouve '(0 0 0) liste))  
        (tri-quadri selection h)  
        (setq liste (dif-liste liste selection))  
        (setq h -100)  
        (while  
          (/= liste '()))
```

```
(and
  (progn
    (setq selection (traite-trace liste ))
    (tri-quadri lfin h)
  )
)
(setq h (- h 100))
(setq liste (dif-liste liste lfin))
)
)
)
)
(if
  (= repl "n")
  (and
    (progn
      (sommetint)
      (setq h 0)
      (tri-quadri selection h)
      (setq liste (dif-liste liste selection))
      (setq h -100)
      (while
        (/= liste '())
        (and
```

```
(progn

  (setq selection (traite-trace liste ))

  (print A)

  (tri-quadri lfin h)

  )

)

(setq h (- h 100))

(setq liste (dif-liste liste lfin))

)

)

)

)

)

)

)

;*****

;*   M. Sc. Aménagement [CMFAO]                                *

;*   Anis Semlali       [GRCAO] 2000                             *

;*****

;-----

; Fonction « selection3df »

; Permet de sélectionner des entités 3df d'Autocad

; Retourne une liste des 3df sélectionnées

;-----

(defun selection3df (); / desc ent rep ssetl x_ent)
```



```

(equal p2 p4)(equal p2 (reverse p4))
(equal p1 p2)(equal p1 (reverse p2))
(equal p1 p3)(equal p1 (reverse p3))
(equal p2 p3)(equal p2 (reverse p3))
)
(setq l (list p1 p2 p3))
(setq l (list p1 p2 p3 p4)))
(setq liste (cons l liste))
)
)
(setq x_ent (+ x_ent 1)) ;compteur des entités à analyser
)
(setq liste (z-off (reverse liste)))
)
)
(prompt "vous n'avez rien choisi !") ;message renvoyé si rien n'est sélectionné
)
)
;*****
;* M. Sc. Aménagement [CMFAO] *
;* Anis Semlali [GRCAO] 2000 *
;*****

; Fonction « traite-trace »

```

```

; Cherche dans la liste générale des 3df ceux qui ont des segments en commun
; avec la liste (selection) des 3df qui forment la dernière couche construite

```

```

;-----

```

```

; Argument

```

```

; (liste): liste de tous les 3df non traités

```

```

;-----

```

```

; Fonctions appelées

```

```

; « elimine-double »

```

```

; « 3df2seg »

```

```

; « sniwa »

```

```

;-----

```

```

; Variables locales

```

```

; temp, y, lseg-f

```

```

;-----

```

```

; Retour

```

```

; (l-fin): liste de tous les 3df qui constituent la rangé à traiter

```

```

;-----

```

```

(defun traite-trace (liste)

```

```

  (setq lseg-f '())

```

```

; transformer la liste des 3df en une liste de segments

```

```

  (setq lseg-f (elimine-double (allo selection)))

```

```

  (setq lfin '())

```

```

  (setq temp '())

```

```

  (mapcar

```

```

(lambda (y)
  (setq temp (sniwa liste y))
  (progn
    (if
      (/= temp nil)
      (setq lfin (append temp lfin))
    )
  );-if
)
);-lambda

lseg-f

);-mapcar

(setq lfin (elimine-double lfin))

);defun

;*****
;*  M. Sc. Aménagement [CMFAO] *
;*  Anis Semlali      [GRCAO] 2000 *
;*****

(defun sommetint ()
  (setvar "osmode" 1)
; sélectionner les points à la périphérie intérieure du tracé
  (setq liste-int ())
  (setq nbr (getint "\n combien de points existent dans la périphérie intérieure?:"))
  (repeat nbr
    (setq pt (getpoint "\n sélectionner les sommets par ordre: "))

```

```
(command "point" pt)

(setq liste-int (cons pt liste-int))

)

(setvar "osmode" 0)

(setq liste-int (reverse liste-int))

; transformer la liste de points en une liste de segments (lseg-f)

(setq lseg-f '())

(setq a (car liste-int))

(setq b (last liste-int))

;(setq lseg-f (cons (list (car liste-int) (last liste-int)) lseg-f))

(while

  (/= liste-int nil)

  (if

    (and

      (/= (car liste-int) nil)

      (/= (cadr liste-int) nil)

    )

    (and

      (progn

        (setq seg (list (car liste-int) (cadr liste-int)))

        (setq lseg-f (cons seg lseg-f))

      )

    )

    (setq liste-int (cdr liste-int))

  )

)
```



```
); - progn
); - and
(setq seg (list a b))
(setq lseg-f (cons seg lseg-f))
(setq lseg-f (reverse lseg-f))
; trouver la liste des 3df qui ont un segment en commun avec la liste
; précédente
; (setq lfin '())
; (setq temp '());
; (mapcar
; '(lambda (y)
;   (setq temp (sniwa liste y))
;   (progn
;     (if
;       (/= temp nil)
;       (setq lfin (append temp lfin))
;     ); - if
;   )
; ); - lambda
; lseg-f
; ); - mapcar
; (setq selection (elimine-double lfin))
; )
;*****
```

;* **M. Sc. Aménagement [CMFAO]** *

;* **Anis Semlali [GRCAO] 2000** *

; **Fonction** « tri-quadri »

; Retourne la nature de la forme du 3df

; **Arguments**

; (liste): liste de tous les 3df d'une même rangé

; h: valeur correspondante à la hauteur de la rangé en cours de traitement

(defun tri-quadri (liste h)

 (mapcar

 '(lambda (3df)

 (progn

 ; (cond

 ; pour les triangles isocèles rectangles codés par 2 points

 (if

 (= (length 3df) 3)

 (TraiterTriangle 3df h))

 (if

 (= (length 3df) 4)

 (TraiterQuadrilatere 3df h))

; (t (list "vous avez introduit une liste qui a moins de 3 éléments ou plus de ;4 éléments"))

); - cond

); - progn

); - lambda

liste

); - mapcar

)

;* M. Sc. Aménagement [CMFAO] *

;* Anis Semlali [GRCAO] 2000 *

; **Fonction** « TraiterTriangle »

; Intervient si la longueur de la liste des 3df = 3 et permet de reconnaître le type

; des triangles

; **Argument**

; 3df

; **Fonctions appelées**

; « corrige-angle »

; fonction qui simplifie l'angle A en un angle compris entre 0 et 90

; « elimine-double »

; fonction qui élimine les éléments doubles dans une liste donnée

; « angle-triangle »

; fonction qui calcule les 3 angles dans un triangle quelconque

; « angle3df »

; fonction qui retourne l'angle compris entre 0 et 90

; « pt-centre »

; fonction qui calcule la distance entre chaque point d'un 3df au centre de la trame

; « nbr-max »

; fonction qui retourne le maximum des éléments de la liste (liste-d)

; « nbr-max2 »

; fonction qui retourne la longueur de la liste (list-temp)

;-----

; **Retour**

; le type du triangle avec son modèle 3d

;-----

(defun TraiterTriangle (3df h)

(setq l1 (angle3df (angle-triangle 3df)))

(progn

(setq d1 (distance (car 3df) (cadr 3df))) ;d1 distance entre pt1 et pt2

(setq d2 (distance (car 3df) (last 3df))) ;d2 distance entre pt2 et pt3

(setq d3 (distance (cadr 3df) (last 3df))) ;d3 distance entre pt1 et pt3

);(cond

; pour les triangles isocèles à 1 point dans l'angle entre les 45 degrés et les ;
triangles isocèles à 1 point dans l'angle aiguë

```
(if
  (or
    (/= nil (member 45 11))
    (/= nil (member 67 11))
    (< (abs (- d1 108.2382)) 0.01);si aucun coté n'est égal à 200
    (< (abs (- d2 108.2382)) 0.01)
    (< (abs (- d3 108.2382)) 0.01)
    ;(= (nbr-max2 (nbr-max (pt-centre 3df))) 2)
  )
  (and
    (loza1-2 3df h)
    (setq 3df (cons 3df 'tri2points1))
  )
); 3ième condition
```

; pour les triangles isocèles rectangles à 2 points

```
(if
  (and
    (= (nbr-max2 (nbr-max (pt-centre 3df))) 2)
    (or
      (/= nil (member 45 11))
      (/= nil (member 90 11))
    )
  )
```

```

(or
  (< (abs(- d1 199.9981)) 0.01);199.9981 au lieu de 200( 2= 2 * module )
  (< (abs(- d2 199.9981)) 0.01);< à cause de l'erreur de osnap
  (< (abs(- d3 199.9981)) 0.01)
); - or
); - and
  (and
    (serwaliya 3df h)
    (setq 3df (cons 3df 'triisorec2points ))
  )
); 1ière condition
;pour les triangles isocèles rectangles à 1 point ou à X
(if
  (and
    ;(= (nbr-max2 (nbr-max (pt-centre 3df))) 1)
    (= (length (ptloin3 3df)) 1)
    (/= nil (member 45 11))
    (/= nil (member 90 11));csa
  ); - and
  (and
    (progn
      (setq pa (car (ptloin3 3df)))
      (setq pb (car (dif-liste 3df (ptloin3 3df))))
      (setq pc (cadr (dif-liste 3df (ptloin3 3df))))
    )
  )
)

```

```

(setq ang (anglept (cal "ang(pa,pb,pc)")))
  (if
    (< (abs (- (abs ang) 90)) 0.5)
    (and
      (dembouq 3df h)
      ;(serwaliya2 3df h)
      (setq 3df (cons 3df 'triisorec1points/x ))
    )
    )
  (if
    (< (abs (- (abs ang) 45)) 0.5)
    (and
      (serwaliya 3df h)
      (setq 3df (cons 3df 'triisorec2points ))
    )
    )
  )
  )
  ); 2ième condition
);progn
)

```

```

;*****
;*  M. Sc. Aménagement [CMFAO] *
;*  Anis Semlali      [GRCAO] 2000 *
;*****
;-----
; Fonction « Traiterquadrilatere »
; Intervient si la longueur de la liste des 3df = 4 et permet de reconnaître le type
; du quadrilatère (rectangle ou losange)
;-----
; Argument
; 3df
;-----
; Retour
; le type du quadrilatère (rectangle ou losange)
;-----
(defun TraiterQuadrilatere (3df h)
  (progn
    (setq d1 (distance (car 3df) (cadr 3df))
          d2 (distance (car 3df) (last 3df))
          d3 (distance (cadr 3df) (caddr 3df))
          d4 (distance (caddr 3df) (last 3df))
    )
    ; pour les rectangles
    (cond

```



```

(
  (and
    (= (nbr-max2 (nbr-max (pt-centre 3df))) 2)
    (or
      (< (abs(- d1 199.9981)) 0.01);199.9981 au lieu de 200( 2= 2 * module )
      (< (abs(- d2 199.9981)) 0.01)
      (< (abs(- d3 199.9981)) 0.01)
      (< (abs(- d4 199.9981)) 0.01)
    )
  ); - and
  (and
    (rectangle 3df h)
    (setq 3df (cons 3df 'rectangle))
  )
)

; pour les losanges
(
  (and
    (> (abs(- d1 199.9981)) 0.01);199.9981 au lieu de 200( 200 = 2 * module )
    (> (abs(- d2 199.9981)) 0.01)
    (> (abs(- d3 199.9981)) 0.01)
    (> (abs(- d4 199.9981)) 0.01)
    (= (nbr-max2 (nbr-max (pt-centre 3df))) 1)
  )
)

```

```

)
(and
  (losange1-3 3df h)
  (setq 3df (cons 3df 'losange ))
)
)
(T (list "je suis désolé vous avez introduit une mauvaise liste"))
); - cond
); - progn
)
;*****
;*  M. Sc. Aménagement [CMFAO] *
;*  Anis Semlali      [GRCAO] 2000 *
;*****
(defun traite-trace (liste)
  (setq lseg-f (3df2seg selection))
  (setq lseg-f (elimine-double lseg-f))
  (setq lfin '())
  (setq temp '())
  (mapcar
    '(lambda (y)
      (setq temp (sniwa liste y))
    )
    lseg-f
  )
  (progn
    (if

```

```
(/= temp nil)
```

```
(setq lfin (append temp lfin))
```

```
); - if
```

```
)
```

```
); - lambda
```

```
lseg-f
```

```
); - mapcar
```

```
(setq lfin (elimine-double lfin))
```

```
); - defun
```

```
*****
```

```
;* M. Sc. Aménagement [CMFAO] *
```

```
;* Anis Semlali [GRCAO] 2000 *
```

```
*****
```

```
(defun sniwa (liste y)
```

```
  (setq lf'())
```

```
  (progn
```

```
(mapcar
```

```
  '(lambda (x)
```

```
(if
```

```
  (= (length x) 4)
```

```
(and
```

```
  (setq pt1 (car x))
```

```
  (setq pt2 (cadr x))
```

```
  (setq pt3 (caddr x))
```

```
(setq pt4 (last x))  
  
(if  
  
  (and  
  
    (or  
  
      (< (distance (car y) pt1) 1)  
      (< (distance (car y) pt2) 1)  
      (< (distance (car y) pt3) 1)  
      (< (distance (car y) pt4) 1)  
    )  
  
    (or  
  
      (< (distance (cadr y) pt1) 1)  
      (< (distance (cadr y) pt2) 1)  
      (< (distance (cadr y) pt3) 1)  
      (< (distance (cadr y) pt4) 1)  
    )  
  
    ) - and  
  
  (setq lf (cons x lf))  
  
  ) - if  
  
  )  
  
(and  
  
  (setq pt1 (car x))  
  
  (setq pt2 (cadr x))  
  
  (setq pt3 (caddr x))  
  
(if
```

```
(and
  (or
    (< (distance (car y) pt1) 1)
    (< (distance (car y) pt2) 1)
    (< (distance (car y) pt3) 1)
  )
  (or
    (< (distance (cadr y) pt1) 1)
    (< (distance (cadr y) pt2) 1)
    (< (distance (cadr y) pt3) 1)
  )
); - and
(setq lf (cons x lf))
); - if
)
); - if
); - lambda
liste
); - mapcar
); - progn
(setq lf (elimine-double lf))
); - defun
```

;*****

;* **M. Sc. Aménagement [CMFAO]** *

;* **Anis Semlali [GRCAO] 2000** *

;*****

;------

; **Fonction « loza1-2 »**

; Détermine la nature du triangle isocèle en cours de traitement

;------

; **Arguments**

; (3df): liste du 3df en traitement

; h: valeur correspondante à la hauteur de la rangé en cours de traitement

;------

; **Fonctions appelées**

; « ptloin3 »

; « dif-liste »

; « anglept »

; « mil3d »

;------

; **Variables locales**

; lpt, l, 3df1, x, x1, pt01, pt02 et pt03

;------

; **Retour**

; renvoi la liste (lpt) vers les fonctions « loza1 » ou « loza2 » pour la construire en

; 3d

```

;-----
(defun loza1-2 (3df h)
  (setvar "osmode" 0)
  (setq 3df1 '());liste des points à la hauteur du tracé
  (mapcar
    '(lambda (x)
      (setq x1 (list (car x) (last x) 0))
      (setq 3df1 (cons x1 3df1))
    )
    3df
  )
  (setq 3df1 (reverse 3df1))
;-----liste des points-----
  (setq pt01 (car (ptloin3 3df1)))
  (setq pt02 (car (dif-liste 3df1 (list pt01))))
  (setq pt03 (cadr (dif-liste 3df1 (list pt01))))
  (cond
    (
      (< (abs (- 45 (anglept (cal "ang(pt01,pt02,pt03)"))))) 2)
    (and
      (setq lpt (list pt01 pt02 pt03))
      (loza2 lpt)
      ;(setq 3df (cons 3df 'loza1))
    )
  )

```

```
)  
(  
  (< (abs (- 67.5 (anglept (cal "ang(pt01,pt02,pt03)))))) 2)  
  (if  
    (< (abs (- 67.5 (anglept (cal "ang(pt02,pt01,pt03)))))) 2)  
    (and  
      (setq lpt (list pt01 pt02 pt03))  
      (loza1 lpt)  
      ;(setq 3df (cons 3df 'loza2))  
    )  
    (and  
      (setq lpt (list pt01 pt03 pt02))  
      (loza1 lpt)  
      ;(setq 3df (cons 3df 'loza2))  
    )  
  )  
  )  
  )  
  (t (list "alach"))  
  )  
; - (princ)  
)
```



```
*****
```

```
;* M. Sc. Aménagement [CMFAO] *
```

```
;* Anis Semlali [GRCAO] 2000 *
```

```
*****
```

```
-----
```

```
; Fonction « losange »
```

```
; Détermine la nature du losange en cours de traitement
```

```
-----
```

```
; Arguments
```

```
; (3df): liste du 3df en traitement
```

```
; h: valeur correspondante à la hauteur de la rangé en cours de traitement
```

```
-----
```

```
; Fonctions appelées
```

```
; « ptloin4 »
```

```
; « dif-liste »
```

```
-----
```

```
; Variables locales
```

```
; lpt, l, 3df1, x, x1, pt01, pt02, pt03, pt04, pt1, pt2, pt3, pt4,l, l02, lo3
```

```
-----
```

```
; Retour
```

```
; Renvoie la liste (lpt) vers les « kataf » et « charbiya »
```

```
-----
```

```
(defun losange1-3 (3df h)
```

```
  (setvar "osmode" 0)
```

```

(setq l '())

(setq lpt '())

(setq 3df1 '());liste des points à la hauteur du tracé

(mapcar
  '(lambda (x)
    (setq x1 (list (car x) (last x) 0))
    (setq 3df1 (cons x1 3df1))
  )
  3df
)

(setq 3df1 (reverse 3df1))

;-----liste des points-----

(setq pt01 (car (ptloin4 3df1)))

(setq l (dif-liste 3df1 (list pt01)))

(setq pt2 (nth 0 l))

(setq pt3 (nth 1 l))

(setq pt4 (nth 2 l))

;(cond

(setq l02 '())

(setq l03 '())

;-----conditions l-----

; (progn

  (if

    (or

```

```

(< (abs (- 45 (anglept (cal "ang(pt01,pt2,pt4)")))) 2)
(< (abs (- 45 (anglept (cal "ang(pt01,pt2,pt3)")))) 2)
(< (abs (- 45 (anglept (cal "ang(pt01,pt3,pt4)")))) 2)
)
;(and
  (progn
    (mapcar
      '(lambda (w)
        (if
          (<= (abs (- (distance pt01 w) 141.42)) 0.1)
          (setq l02 (cons w l02))
          (setq l03 (cons w l03) )
        )
      )
    )
    l
  )
  (setq l02 (reverse l02))
  (setq pt02 (car l02))
  (setq pt04 (last l02))
  (setq pt03 (car (dif-liste l l02)))
  (setq lpt (list pt01 pt02 pt03 pt04))
  (ktaf lpt)
);)
)

```

```

(if
  (or
    (< (abs (- 135 (anglept (cal "ang(pt01,pt2,pt4)))))) 2)
    (< (abs (- 135 (anglept (cal "ang(pt01,pt2,pt3)))))) 2)
    (< (abs (- 135 (anglept (cal "ang(pt01,pt3,pt4)))))) 2)
  )
;and
  (progn
    (mapcar
      '(lambda (w)
        (if
          (<= (abs (- (distance pt01 w) 141.42)) 0.1)
            (setq l02 (cons w l02))
            (setq l03 (cons w l03) )
          )
        )
      )
    l
  )
  (setq l02 (reverse l02))
  (setq pt02 (car l02))
  (setq pt04 (last l02))
  (setq pt03 (car (dif-liste l l02)))
  (setq lpt (list pt01 pt02 pt03 pt04))
  (charbiya lpt)

```

```

;(command "move" aa "" '(0 0 0) '(0 0 h))
);)
)

(setq lpt lpt)
)

;*****
;*  M. Sc. Aménagement [CMFAO] *
;*  Anis Semlali      [GRCAO] 2000 *
;*****
;-----
;Fonction « rectangle »
; Détermine la nature du rectangle en cours de traitement
;-----
;Arguments
; (3df): liste du 3df en cours de traitement
; h: valeur correspondante à la hauteur de la rangé en cours de traitement
;-----
;Fonctions appelées
; « ptloin4 »
; « dif-liste »
;-----
;Variables locales
; lpt, l, 3df1, x, x1, pt01, pt02, pt03, pt04, pt1, pt2, pt3, pt4,
;-----

```

; Retour

; Renvoie la liste (lpt) vers les fonctions « testiya-m » ou « testiya-o » pour la

; construire en 3d

;-----

(defun rectangle (3df h)

 (setvar "osmode" 0)

 (setq l '())

 (setq lpt '())

 (setq 3df1 '());liste des points à la hauteur du tracé

 (mapcar

 '(lambda (x)

 (setq x1 (list (car x) (last x) 0))

 (setq 3df1 (cons x1 3df1))

)

 3df

)

(setq 3df1 (reverse 3df1))

;-----liste des points-----

(setq pt01 (car (ptloin4 3df1)))

(setq l (dif-liste 3df1 (list pt01)))

(setq pt2 (nth 0 l))

(setq pt3 (nth 1 l))

(setq pt4 (nth 2 l))

(cond

```
(; - condition1
(< (abs (- (abs (distance pt01 pt2)) 200)) 2)
(if
  (< (abs (- (abs (distance pt01 pt3)) 141.42)) 2)
    (setq pt02 pt2
          pt03 pt4
          pt04 pt3)
    )
  (setq pt02 pt2
        pt03 pt3
        pt04 pt4)
  )
(; - condition2
(< (abs (- (abs (distance pt01 pt3)) 200)) 2)
(if
  (< (abs (- (abs (distance pt01 pt2)) 141.42)) 2)
    (setq pt02 pt3
          pt03 pt4
          pt04 pt2)
    )
  (setq pt02 pt3
        pt03 pt2
        pt04 pt4)
  )
)
```

```
(; - condition3
(< (abs (- (abs (distance pt01 pt4)) 200)) 2)
(if
(< (abs (- (abs (distance pt01 pt2)) 141.42)) 2)
  (setq pt02 pt4
        pt03 pt3
        pt04 pt2)
  )
(setq pt02 pt4
      pt03 pt3
      pt04 pt2)
  )
); - cond
```

```
(setq lpt (list pt01 pt02 pt03 pt04))
(testiya-o lpt)
)
```

=====

7.2.3 Fonction de construction des modèles tridimensionnels des pièces

=====

.*****


```

;*   Anis Semlali           [GRCAO] 2000           *
;*****
;-----
; Fonction « dembouq »
; Construit le modèle 3d de la pièce dembouq
;-----
; Argument
; (lpt): liste de tous les points du 3df
;-----
; Fonctions appelées
; « mil3d »
; « ptloin3 »
; « dif-liste »
;-----
; Variables locales
; pt01, pt02, pt03, pt04, pt11, pt12, pt13, pt14, mil1, mil2, cy, cy11, pri01 et pri11
;-----
(defun dembouq (3df h)
  (setvar "osmode" 0)
  (setq 3df1 '());liste des points à la hauteur du tracé
  (mapcar
    '(lambda (x)
      (setq x1 (list (car x) (last x) 0))
      (setq 3df1 (cons x1 3df1))

```

```

)
3df
)
(setq 3df1 (reverse 3df1))

```

```
;------liste des points-----
```

```

(setq pt01 (car (ptloin3 3df1)))
(setq pt02 (car (dif-liste 3df1 (list pt01))));(ptloin3 3df1 )))
(setq pt03 (cadr (dif-liste 3df1 (list pt01))));(ptloin3 3df1 )))
(setq pt11 (list (car pt01) (cadr pt01) 200))
(setq pt12 (list (car pt02) (cadr pt02) 200))
(setq pt13 (list (car pt03) (cadr pt03) 200))

```

```
;------construction des faces du prisme -----
```

```

(command "pline" pt01 pt02 pt03 "c" )
(command "extrude" "last" "" 200 "")
(setq der01 (entlast))
(setq mil1 (mil3d pt12 pt13))
(setq mil2 (mil3d pt02 pt03))
(command "ucs" "new" "3point" mil2 pt01 mil1 )
(command "ellipse" "c" '(0 0 0) '(90 0 0) '(0 100 0) )
(command "extrude" "last" "" 200 "")
(setq der02 (entlast))
(command "ellipse" "c" '(0 0 0) '(90 0 0) '(0 100 0) )
(command "extrude" "last" "" -200 "")
(setq der03 (entlast))

```

```
(command "ucs" "w")
```

```
(command "subtract" der01 "" der02 der03 "")
```

```
(command "move" "last" "" (list 0.0 0.0 0.0) (list 0.0 0.0 -200));
```

```
(command "move" "last" "" (list 0.0 0.0 0.0) (list 0.0 0.0 h))
```

```
)
```

```
*****
```

```
;* M. Sc. Aménagement [CMFAO] *
```

```
;* Anis Semlali [GRCAO] 2000 *
```

```
*****
```

```
-----
```

```
; Fonction « serwaliya »
```

```
; Détermine la nature du triangle isocèle rectangle en cours de traitement
```

```
-----
```

```
; Arguments
```

```
; (3df): liste du 3df en cours de traitement
```

```
; h: valeur correspondante à la hauteur de la rangé en cours de traitement
```

```
-----
```

```
; Fonctions appelées
```

```
; « ptloin3 »
```

```
; « dif-liste »
```

```
; « anglept »
```

```
-----
```

```
; Variables locales
```

```
; lpt, l, 3df1, x, x1, cyl1, cyl2, tri1, tri2, pt01, pt02 , pt03, pt11, pt12 et pt13
```

```

;-----
(defun serwaliya (3df h)
  (setvar "osmode" 0)
  (setq 3df1 '());liste des points a la hauteur du tracé
  (mapcar
    '(lambda (x)
      (setq x1 (list (car x) (last x) 0))
      (setq 3df1 (cons x1 3df1))
    )
    3df
  )
  (setq 3df1 (reverse 3df1))
;-----liste des points-----
  (setq pt02 (car (ptloin3 3df1)))
  (setq pt01 (car (dif-liste 3df1 (list pt02 ))))
  (setq pt03 (car (dif-liste 3df1 (list pt02 pt01))))
  (setq pt11 (list (car pt01) (cadr pt01) 200))
  (setq pt12 (list (car pt02) (cadr pt02) 200))
  (setq pt13 (list (car pt03) (cadr pt03) 200))
;-----construction des faces du prisme -----
  (setq mil1 (mil3d pt12 pt13))
  (setq mil2 (mil3d pt02 pt03))
  (command "pline" mil2 pt03 pt01 "c" )
  (command "extrude" "last" "" 200 "")

```

```
(setq tri01 (entlast))
```

```
(command "pline" mil2 pt02 pt01 "c" )
```

```
(command "extrude" "last" "" 200 "")
```

```
(setq tri02 (entlast))
```

```
;-----construction des ellipses -----
```

```
(command "ucs" "w")
```

```
(command "ucs" "new" "3point" pt02 pt12 pt03 )
```

```
(command "ellipse" "c" '(0 0 0) '(117.75 0 0) '(0 190 0) );,135.25
```

```
(command "extrude" "last" "" 150 "")
```

```
(setq cyl11 (entlast))
```

```
(command "ucs" "w")
```

```
(command "ucs" "new" "3point" pt02 pt12 pt03 )
```

```
(command "ellipse" "c" '(0 0 0) '(117.75 0 0) '(0 190 0) )
```

```
(command "extrude" "last" "" -150 "")
```

```
(setq cyl12 (entlast))
```

```
(command "ucs" "w")
```

```
(command "ucs" "new" "3point" pt03 pt13 pt02 )
```

```
(command "ellipse" "c" '(0 0 0) '(117.75 0 0) '(0 190 0) )
```

```
(command "extrude" "last" "" -150 "")
```

```
(setq cyl21 (entlast))
```

```
(command "ucs" "w")
```

```
(command "ucs" "new" "3point" pt03 pt13 pt02 )
```

```
(command "ellipse" "c" '(0 0 0) '(117.75 0 0) '(0 190 0) )
```

```
(command "extrude" "last" "" 150 "")
```

```

      (setq cyl22 (entlast))

;-----ellipse2 -----

; (command "ucs" "w")

; (command "ucs" "new" "3point" pt01 mil2 pt11 )

; (command "ellipse" "c" '(0 0 0) '(90 0 0) '(0 100 0) )

; (command "extrude" "last" "" 200 "")

; (setq cyl31 (entlast))

; (command "ellipse" "c" '(0 0 0) '(90 0 0) '(0 100 0) )

; (command "extrude" "last" "" -200 "")

; (setq cyl32 (entlast))

;-----construction de la pièce -----

      (command "ucs" "w")

; (command "subtract" tri02 "" cyl31 "")

; (command "subtract" tri01 "" cyl32 "")

      (command "subtract" tri01 "" cyl11 cyl12 "")

      (command "subtract" tri02 "" cyl21 cyl22 "")

      (command "union" tri01 tri02 "")

      (command "move" "last" "" (list 0.0 0.0 0.0) (list 0.0 0.0 -200))

      (command "move" "last" "" (list 0.0 0.0 0.0) (list 0.0 0.0 h))

)

```

```

;*****
;*  M. Sc. Aménagement [CMFAO] *
;*  Anis Semlali      [GRCAO] 2000 *
;*****
;-----
; Fonction « chiira »
; Détermine la nature du triangle isocèle rectangle en cours de traitement
;-----
; Arguments
; (3df): liste du 3df en cours de traitement
; h: valeur correspondante à la hauteur de la rangé en cours de traitement
;-----
; Fonctions appelées
; « ptloin3 »
; « dif-liste »
; « anglept »
;-----
; Variables locales
; lpt, l, 3df1, x, x1, pt01, pt02 et pt03, pt11, pt12 et pt13, cyl1, cyl2, tri1, tri2
; centre-pied (xp yp zp), pied, pied2
;-----
(defun serwaliya2 (3df h)
  (setvar "osmode" 0)
  (setq 3df1 '());liste des points à la hauteur du tracé

```

; Ajouter la coordonnée en Z pour tous les points du 3df

(mapcar

'(lambda (x)

(setq x1 (list (car x) (last x) 0))

(setq 3df1 (cons x1 3df1))

)

3df

)

(setq 3df1 (reverse 3df1))

;-----liste des points-----

; le premier point le plus loin du centre dans la liste du 3df

(setq pt01 (car (ptloin3 3df1)))

; le deuxième point le plus loin du centre dans la liste du 3df

(setq pt02 (car (dif-liste 3df1 (list pt01))))

; le dernier point du 3df

(setq pt03 (cadr (dif-liste 3df1 (list pt01))))

;Extruder le triangle formé par pt01, pt02 et pt03

(setq pt11 (list (car pt01) (cadr pt01) 200))

(setq pt12 (list (car pt02) (cadr pt02) 200))

(setq pt13 (list (car pt03) (cadr pt03) 200))

;----- construction des faces du prisme-----

(setq mil1 (mil3d pt12 pt13))

(setq mil2 (mil3d pt02 pt03))

(command "pline" mil2 pt03 pt01 "c")


```
(command "extrude" "last" "" 200 "")
```

```
; nommer la dernière entité créée
```

```
(setq tri01 (entlast))
```

```
(command "pline" mil2 pt02 pt01 "c" )
```

```
(command "extrude" "last" "" 200 "")
```

```
(setq tri02 (entlast))
```

```
;-----construction des ellipses-----
```

```
(command "ucs" "w")
```

```
(command "ucs" "new" "3point" pt02 pt12 pt03 )
```

```
(command "ellipse" "c" '(0 0 0) '(117.75 0 0) '(0 190 0) );;135.25
```

```
(command "extrude" "last" "" 150 "")
```

```
(setq cyl11 (entlast))
```

```
(command "ucs" "w")
```

```
(command "ucs" "new" "3point" pt02 pt12 pt03 )
```

```
(command "ellipse" "c" '(0 0 0) '(117.75 0 0) '(0 190 0) )
```

```
(command "extrude" "last" "" -150 "")
```

```
(setq cyl12 (entlast))
```

```
(command "ucs" "w")
```

```
(command "ucs" "new" "3point" pt03 pt13 pt02 )
```

```
(command "ellipse" "c" '(0 0 0) '(117.75 0 0) '(0 190 0) )
```

```
(command "extrude" "last" "" -150 "")
```

```
(setq cyl21 (entlast))
```

```
(command "ucs" "w")
```

```
(command "ucs" "new" "3point" pt03 pt13 pt02 )
```

```
(command "ellipse" "c" '(0 0 0) '(117.75 0 0) '(0 190 0) )
```

```
(command "extrude" "last" "" 150 "")
```

```
(setq cyl22 (entlast))
```

```
;----- construction du pied de la pièce -----
```

```
(setq xp (+ (/ (+ (car pt01) (car mil2)) 2) (/ (- (car pt01) (car mil2)) 4)))
```

```
(setq yp (+ (/ (+ (cadr pt01) (cadr mil2)) 2) (/ (- (cadr pt01) (cadr mil2)) 4)))
```

```
(setq zp (caddr pt01) )
```

```
(setq centre-pied (list xp yp zp))
```

```
(command "ucs" "w")
```

```
(command "polygon" 4 centre-pied "i" 25)
```

```
(setq pied (entlast))
```

```
(if
```

```
(or
```

```
(< (abs xp) 0.01)
```

```
(< (abs yp) 0.01)
```

```
)
```

```
(command "rotate" pied "" centre-pied 45)
```

```
)
```

```
(command "extrude" pied "" 200 "")
```

```
(setq pied (entlast))
```

```
(command "move" pied "" (list 0.0 0.0 0.0) (list 0.0 0.0 -100))
```

```
(command "polygon" 4 centre-pied "i" 25)
```

```
(setq pied2 (entlast))
```

```
(if
```

```

(or
  (< (abs xp) 0.01)
  (< (abs yp) 0.01)
)

(command "rotate" pied2 "" centre-pied 45)

)

(command "extrude" pied2 "" -100 "")

(setq pied2 (entlast))

(command "move" pied2 "" (list 0.0 0.0 0.0) (list 0.0 0.0 -100))

;----- construction de la pièce -----

(command "ucs" "w")

(command "subtract" tri01 "" cyl11 cyl12 "")

(command "subtract" tri02 "" cyl21 cyl22 "")

(command "union" tri01 tri02 pied pied2 "")

(command "move" tri01 tri02 "" (list 0.0 0.0 0.0) (list 0.0 0.0 -200))

(command "move" tri01 tri02 "" (list 0.0 0.0 0.0) (list 0.0 0.0 h))

)

;*****
;*  M. Sc. Aménagement [CMFAO] *
;*  Anis Semlali      [GRCAO] 2000 *
;*****
;-----

; Fonction « loza1 »

; Construit le modèle 3d de la pièce loza1

```

```

;-----
; Argument
; (lpt): liste de tous les points du 3df
;-----

; Fonctions appelées
; « mil3d »
;-----

; Variables locales
; pt01, pt02, pt03, pt04, pt11, pt12, pt13, pt14, mil1, mil2, cy, cyl1, pri01 et pri11
;-----

;=====
; Fonction Loza1
;=====

(defun loza1 (lpt)
  (setq pt01 (car lpt))
  (setq pt02 (cadr lpt))
  (setq pt03 (last lpt))
  (setq pt11 (list (car pt01) (cadr pt01) 400))
  (setq pt12 (list (car pt02) (cadr pt02) 400))
  (setq pt13 (list (car pt03) (cadr pt03) 400))

;----- construction des faces du prisme -----

  (setq mil1 (mil3d pt11 pt12))
  (setq mil2 (mil3d pt01 pt02))

```

```
(command "pline" pt01 pt02 pt03 "c" )
```

```
(command "extrude" "last" "" 200 "")
```

```
(setq der01 (entlast))
```

```
;-----construction des ellipses-----
```

```
(command "ucs" "new" "3point" mil2 pt03 mil1 )
```

```
(command "ellipse" "c" '(0 0 0) '(120.655 0 0) '(0 100 0))
```

```
(command "extrude" "last" "" 200 "")
```

```
(command "move" "last" "" '(0 0 0) '(130.655 0 0))
```

```
(setq der02 (entlast))
```

```
(command "ellipse" "c" '(0 0 0) '(120.655 0 0) '(0 100 0))
```

```
(command "extrude" "last" "" -200 "")
```

```
(command "move" "last" "" '(0 0 0) '( 130.655 0 0))
```

```
(setq der03 (entlast))
```

```
;-----construction de la pièce-----
```

```
(command "ucs" "w")
```

```
(command "subtract" der01 "" der02 der03 "")
```

```
(command "move" der01 "" (list 0.0 0.0 0.0) (list 0.0 0.0 -200))
```

```
(command "move" der01 "" (list 0.0 0.0 0.0) (list 0.0 0.0 h))
```

```
)
```

```

;*****
;*  M. Sc. Aménagement [CMFAO] *
;*  Anis Semlali      [GRCAO] 2000 *
;*****
;-----
; Fonction « loza2 »
; Construit le modèle 3d de la pièce loza2
;-----
; Argument
; (lpt): liste de tous les points du 3df
;-----
; Fonctions appelées
; « mil3d »
;-----
; Variables locales
; pt01, pt02, pt03, pt04, pt11, pt12, pt13, pt14, mil1, mil2, cy, cyl1, pri01 et pri11
;-----
(defun loza2 (lpt)
  (setq pt01 (car lpt))
  (setq pt02 (cadr lpt))
  (setq pt03 (last lpt))
  (setq pt11 (list (car pt01) (cadr pt01) 400))
  (setq pt12 (list (car pt02) (cadr pt02) 400))
  (setq pt13 (list (car pt03) (cadr pt03) 400))

```

;------ construction des faces du prisme-----;

```
(setq mil1 (mil3d pt12 pt13))  
  
(setq mil2 (mil3d pt02 pt03))  
  
(command "pline" pt01 pt02 pt03 "c" )  
  
(command "extrude" "last" "" 200 "")  
  
(setq der01 (entlast))
```

;------ construction des ellipses-----;

```
(command "ucs" "new" "3point" mil2 pt01 mil1 )  
  
(command "ellipse" "c" '(0 0 0) '(120.655 0 0) '(0 100 0) )  
  
(command "extrude" "last" "" 200 "")  
  
(setq der02 (entlast))  
  
(command "ellipse" "c" '(0 0 0) '(120.655 0 0) '(0 100 0) )  
  
(command "extrude" "last" "" -200 "")  
  
(setq der03 (entlast))  
  
(command "ucs" "w")  
  
(command "subtract" der01 "" der02 der03 "")  
  
(command "move" der01 "" (list 0.0 0.0 0.0) (list 0.0 0.0 -200))  
  
(command "move" der01 "" (list 0.0 0.0 0.0) (list 0.0 0.0 h))  
  
)
```

```

;*****
;*  M. Sc. Aménagement [CMFAO] *
;*  Anis Semlali      [GRCAO] 2000 *
;*****
;-----
; Fonction « Ktaf »
; Construit le modèle 3d de la pièce ktaf
;-----
; Argument
; (lpt): liste de tous les points du 3df
;-----
; Variables locales
; pt01, pt02, pt03, pt04, pt11, pt12, pt13, pt14, mil1, mil2, cy, cyl1, cyl0, cyl01,
; cyl111
; cyl12, cyl21, cyl22, los01 et los11
;-----
(defun ktaf (lpt)
  (setvar "osmode" 0)
  (progn
    (setq pt01 (nth 0 lpt))
    (setq pt02 (nth 1 lpt))
    (setq pt03 (nth 2 lpt))
    (setq pt04 (nth 3 lpt))
    (setq pt11 (list (car pt01) (cadr pt01) 400))

```



```
(setq pt12 (list (car pt02) (cadr pt02) 400))
```

```
(setq pt13 (list (car pt03) (cadr pt03) 400))
```

```
(setq pt14 (list (car pt04) (cadr pt04) 400))
```

```
;----- construction des faces du prisme-----
```

```
(setq mil1 (mil3d pt12 pt14))
```

```
(setq mil2 (mil3d pt02 pt04))
```

```
(command "pline" pt01 pt02 pt04 "c" )
```

```
(command "extrude" "last" "" 400 "")
```

```
(setq los01 (entlast))
```

```
;----- construction des ellipses-----
```

```
(command "ucs" "new" "3point" mil2 pt01 mil1 )
```

```
(command "ellipse" "c" '(0 0 0) '(120.655 0 0) '(0 180 0) )
```

```
(command "extrude" "last" "" 200 "")
```

```
(setq cylos02 (entlast))
```

```
(command "ellipse" "c" '(0 0 0) '(120.655 0 0) '(0 180 0) )
```

```
(command "extrude" "last" "" -200 "")
```

```
(setq cylos03 (entlast))
```

```
(command "ucs" "w")
```

```
(command "subtract" los01 "" cylos02 cylos03 "")
```

```
;----- construction des faces du prisme2 -----
```

```
(command "pline" pt03 pt02 pt04 "c" )
```

```
(command "extrude" "last" "" 200 "")
```

```
(setq los11 (entlast))
```

-----construction des ellipses2-----

```
(command "ucs" "new" "3point" pt03 pt01 pt13 )
(command "ellipse" "c" '(0 0 0) '(110.655 0 0) '(0 100 0) )
(command "extrude" "last" "" 200 "")
(setq cylos12 (entlast))
(command "ellipse" "c" '(0 0 0) '(110.655 0 0) '(0 100 0) )
(command "extrude" "last" "" -200 "")
(setq cylos13 (entlast))
(command "ucs" "w")
(command "subtract" los11 "" cylos12 cylos13 "")
(command "move" los11 "" '(0 0 0) '(0 0 200) )
(command "union" los01 los11 "")
(command "move" "last" "" (list 0.0 0.0 0.0) (list 0.0 0.0 -400))
(command "move" "last" "" (list 0.0 0.0 0.0) (list 0.0 0.0 h))
)
)
```

;* **M. Sc. Aménagement [CMFAO]** *

;* **Anis Semlali [GRCAO] 2000** *

; **Fonction « charbiya »**

; **Construit le modèle 3d de la pièce charbiya**

; Argument

; (lpt): liste de tous les points du 3df

; Variables locales

; pt01, pt02, pt03, pt04, pt11, pt12, pt13, pt14, mil1, mil2, cy, cyl1, cyl0, cyl01, cyl111

; cyl12, cyl21, cyl22, los01 , los11, pd(xd, yd, zd) et pg (xg, yg, zg)

; Retour

; (list-temp) : liste qui retourne les points les plus loin du centre (1 ou 2 points)

(defun charbiya (lpt)

(setvar "osmode" 0)

(progn

(setq pt01 (nth 0 lpt))

(setq pt02 (nth 1 lpt))

(setq pt03 (nth 2 lpt))

(setq pt04 (nth 3 lpt))

(setq pt11 (list (car pt01) (cadr pt01) 400))

(setq pt12 (list (car pt02) (cadr pt02) 400))

(setq pt13 (list (car pt03) (cadr pt03) 400))

(setq pt14 (list (car pt04) (cadr pt04) 400))

-----construction des faces du prisme-----

(setq mil1 (mil3d pt12 pt14))

```
(setq mil2 (mil3d pt02 pt04))  
  
(command "pline" pt01 pt02 pt03 "c" )  
  
(command "extrude" "last" "" 200 "")  
  
(setq los01 (entlast))
```

-----construction des ellipses-----

```
(command "ucs" "new" "3point" pt04 pt02 pt14 )  
  
(command "ellipse" "c" '(0 0 0) '(241.3101 0 0) '(0 180 0) )  
  
(command "extrude" "last" "" 200 "")  
  
(setq cylos02 (entlast))  
  
(command "ellipse" "c" '(0 0 0) '(241.3101 0 0) '(0 180 0) )  
  
(command "extrude" "last" "" -200 "")  
  
(setq cylos03 (entlast))  
  
(command "ucs" "w")  
  
(command "subtract" los01 "" cylos02 cylos03 "")
```

-----construction des faces du prisme2-----

```
(command "pline" pt01 pt04 pt03 "c" )  
  
(command "extrude" "last" "" 200 "")  
  
(setq los11 (entlast))
```

-----construction des ellipses2-----

```
(command "ucs" "new" "3point" pt02 pt04 pt12 )  
  
(command "ellipse" "c" '(0 0 0) '(241.3101 0 0) '(0 180 0) )  
  
(command "extrude" "last" "" 200 "")  
  
(setq cylos12 (entlast))  
  
(command "ellipse" "c" '(0 0 0) '(241.3101 0 0) '(0 180 0) )
```

```
(command "extrude" "last" "" -200 "")  
  
(setq cylos13 (entlast))  
  
(command "ucs" "w")  
  
(command "ucs" "new" "3point" mil2 pt02 mil1 )  
  
(setq xp (+ (/ (+ (car pt01) (car pt03)) 2) (/ (- (car pt01) (car pt03)) 4)))  
  
(setq yp (+ (/ (+ (cadr pt01) (cadr pt03)) 2) (/ (- (cadr pt01) (cadr pt03))  
4)))  
  
(setq zp (caddr pt01) )  
  
(setq xg (+ (/ (+ (car pt01) (car pt02)) 2) (/ (- (car pt01) (car pt02)) 4)))  
  
(setq yg (+ (/ (+ (cadr pt01) (cadr pt02)) 2) (/ (- (cadr pt01) (cadr pt02))  
4)))  
  
(setq zg (caddr pt01) )  
  
(setq xd (+ (/ (+ (car pt01) (car pt04)) 2) (/ (- (car pt01) (car pt04)) 4)))  
  
(setq yd (+ (/ (+ (cadr pt01) (cadr pt04)) 2) (/ (- (cadr pt01) (cadr pt04))  
4)))  
  
(setq zd (caddr pt01) )  
  
(setq centre-pied (list xp yp zp))  
  
(setq pd (list xd yd zd))  
  
(setq pg (list xg yg zg))  
  
(command "ucs" "w")  
  
(command "pline" pt01 pd centre-pied pg "c")  
  
(setq pied (entlast))  
  
(command "extrude" pied "" 200 "")  
  
(setq pied (entlast))
```

```

(command "move" "last" "" (list 0.0 0.0 0.0) (list 0.0 0.0 -200))

;(command "move" "last" "" (list 0.0 0.0 0.0) (list 0.0 0.0 h))

(command "pline" pt01 pd centre-pied pg "c")

(setq pied2 (entlast))

(command "extrude" pied2 "" -100 "")

(setq pied2 (entlast))

(command "move" "last" "" (list 0.0 0.0 0.0) (list 0.0 0.0 -200))

;(command "move" "last" "" (list 0.0 0.0 0.0) (list 0.0 0.0 h))

(command "ucs" "w")

(command "subtract" los11 "" cylos12 cylos13 "")

(command "move" los11 los01 "" (list 0.0 0.0 0.0) (list 0.0 0.0 -200))

(command "union" los01 los11 pied pied2 "")

;(setq aie (entlast))

(command "move" los01 los11 pied pied2 "" (list 0.0 0.0 0.0) (list 0.0 0.0
h))

)

)

;*****
;*  M. Sc. Aménagement [CMFAO] *
;*  Anis Semlali      [GRCAO] 2000 *
;*****
;-----
;Fonction « testiya-m »

; Construit le modèle 3d de la pièce testiya-m

```

```

;-----
; Argument
; (lpt)
;-----

; Variables locales
; pt01, pt02, pt03, pt04, pt11, pt12, pt13, pt14, rec, cyl, cyl2
;-----

(defun testiya-m (lpt)
  (setvar "osmode" 0)
  (progn
    (setq pt01 (nth 0 lpt))
    (setq pt02 (nth 1 lpt))
    (setq pt03 (nth 2 lpt))
    (setq pt04 (nth 3 lpt))
    (setq pt11 (list (car pt01) (cadr pt01) 200))
    (setq pt12 (list (car pt02) (cadr pt02) 200))
    (setq pt13 (list (car pt03) (cadr pt03) 200))
    (setq pt14 (list (car pt04) (cadr pt04) 200))
    ;-----construction des faces du prisme-----
    (command "pline" pt01 pt02 pt03 pt04 "c" )
    (command "extrude" "last" "" 400 "")
    (setq rec (entlast))
    ;-----construction des ellipses-----
    (command "ucs" "new" "3point" pt04 pt14 pt01)
  )
)

```

```
(command "ellipse" "c" '(0 0 0) '(200 0 0) '(0 121.32 0) )
(command "extrude" "last" "" 200 "")
(setq cyl (entlast))
(command "ellipse" "c" '(0 0 0) '(200 0 0) '(0 121.32 0) )
(command "extrude" "last" "" -200 "")
(setq cyl2 (entlast))
(command "ucs" "w")
(command "subtract" rec "" cyl cyl2 "")
(command "move" "last" "" (list 0.0 0.0 0.0) (list 0.0 0.0 -400))
(command "move" "last" "" (list 0.0 0.0 0.0) (list 0.0 0.0 h))
))
```

```
*****
```

```
;* M. Sc. Aménagement [CMFAO] *
```

```
;* Anis Semlali [GRCAO] 2000 *
```

```
*****
```

```
-----
```

```
; Fonction « testiya-o »
```

```
; Construit le modèle 3d de la pièce testiya-o
```

```
-----
```

```
; Argument
```

```
; (lpt): liste des points du 3df en cours de traitement
```

```
-----
```

```
; Fonction appelée
```

```
; « mil3d »
```

; Variables locales

; pt01, pt02, pt03, pt04, pt11, pt12, pt13, pt14, mil1, mil2, mil3, mil01, mil02,

; mil03,

; mil04, mil11, mil12, mil13, cy, cyl1, cyl0, cyl01, cyl11, cyl12, cyl21, cyl22,

; rec1 et rec2

; Retour

; (list-temp): liste qui retourne les points les plus loin du centre (1 ou 2 points)

(defun testiya-o (lpt)

(setvar "osmode" 0)

(progn

(setq pt01 (nth 0 lpt))

(setq pt02 (nth 1 lpt))

(setq pt03 (nth 2 lpt))

(setq pt04 (nth 3 lpt))

(setq pt11 (list (car pt01) (cadr pt01) 200))

(setq pt12 (list (car pt02) (cadr pt02) 200))

(setq pt13 (list (car pt03) (cadr pt03) 200))

(setq pt14 (list (car pt04) (cadr pt04) 200))

-----construction des faces du prisme-----

(setq mil1 (mil3d pt01 pt02))

(setq mil2 (mil3d pt03 pt04))

```
(setq mil3 (mil3d pt13 pt14))

(command "pline" pt01 mil1 mil2 pt04 "c" )

(command "extrude" "last" "" 400 "")

(setq rec1 (entlast))

(command "pline" mil1 pt02 pt03 mil2 "c" )

(command "extrude" "last" "" 400 "")

(setq rec2 (entlast))
```

-----construction des ellipses-----

```
;les cylindres au ses de la largeurs

(setq mil01 (mil3d pt01 pt04))

(setq mil11 (mil3d pt11 pt14))

(setq mil02 (mil3d pt02 pt03))

(setq mil12 (mil3d pt12 pt13))

(setq mil03 (mil3d pt03 pt04))

(setq mil13 (mil3d pt13 pt14))

(setq mil04 (mil3d pt01 pt02))

(command "ucs" "new" "3point" pt04 pt14 pt01)

(command "ellipse" "c" '(0 0 0) '(200 0 0) '(0 121.32 0) )

(command "extrude" "last" "" 200 "")

(setq cyl (entlast))

(command "ellipse" "c" '(0 0 0) '(200 0 0) '(0 121.32 0) )

(command "extrude" "last" "" -200 "")

(setq cyl1 (entlast))

(command "ellipse" "c" '(0 0 0) '(200 0 0) '(0 121.32 0) )
```

```
(command "extrude" "last" "" 200 "")  
(setq cyl0 (entlast))  
(command "ellipse" "c" '(0 0 0) '(200 0 0) '(0 121.32 0) )  
(command "extrude" "last" "" -200 "")  
(setq cyl01 (entlast))  
;les cylindres au sens de la longueur  
(command "ucs" "w")  
(command "ucs" "new" "3point" pt03 pt13 pt04 )  
(command "ellipse" "c" '(0 0 0) '(235.25 0 0) '(0 190 0) )  
(command "extrude" "last" "" 150 "")  
(setq cyl11 (entlast))  
(command "ellipse" "c" '(0 0 0) '(235.25 0 0) '(0 190 0) )  
(command "extrude" "last" "" -150 "")  
(setq cyl12 (entlast))  
(command "ucs" "w")  
(command "ucs" "new" "3point" pt04 pt14 pt03 )  
(command "ellipse" "c" '(0 0 0) '(235.25 0 0) '(0 190 0) )  
(command "extrude" "last" "" -150 "")  
(setq cyl21 (entlast))  
(command "ucs" "w")  
(command "ucs" "new" "3point" pt04 pt14 pt03 )  
(command "ellipse" "c" '(0 0 0) '(235.25 0 0) '(0 190 0) )  
(command "extrude" "last" "" 150 "")  
(setq cyl22 (entlast))
```

```

(command "ucs" "w")

(command "subtract" rec1 "" cyl11 cyl12 cyl cyl11 "")

(command "subtract" rec2 "" cyl21 cyl22 cyl0 cyl01 "")

(command "union" rec1 rec2 "")

(command "move" "last" "" (list 0.0 0.0 0.0) (list 0.0 0.0 -400))

(command "move" "last" "" (list 0.0 0.0 0.0) (list 0.0 0.0 h))

))

```

```

=====
;

```

;Les Sous-routines

```

=====
;

```

```

;*****
;

```

```

;*  M. Sc. Aménagement [CMFAO] *

```

```

;*  Anis Semlali      [GRCAO] 2000 *

```

```

;*****
;

```

```

;-----

```

; Fonction « 3df2seg »

; Transforme une liste de 3df en une liste de segments

```

;-----

```

; Argument

; la liste des 3df

```

;-----

```

; Retour

```
; la liste (lseg)
```

```
-----
```

```
(defun 3df2seg (selection)
```

```
  ;on constitue une liste de segments
```

```
  ;initialisation
```

```
  (setq lseg '())
```

```
  ;construction de la liste des segments dans « lseg »
```

```
  (mapcar
```

```
    '(lambda (3df)
```

```
      (setq seg1 (list (car 3df) (cadr 3df)))
```

```
      (setq seg2 (list (cadr 3df)(caddr 3df)))
```

```
      (setq seg3 (list (caddr 3df) (last 3df)))
```

```
      (setq seg4 (list (last 3df)(car 3df)))
```

```
      (setq lseg (cons seg1 lseg))
```

```
      (setq lseg (cons seg2 lseg))
```

```
      (setq lseg (cons seg3 lseg))
```

```
      (setq lseg (cons seg4 lseg))
```

```
    )
```

```
  selection
```

```
  )
```

```
(setq lseg lseg)
```

```
)
```

```
*****  
;* M. Sc. Aménagement [CMFAO] *  
;* Anis Semlali [GRCAO] 2000 *  
*****  
;-----  
; Fonction « angle-triangle »  
  
; Calcule les 3 angles dans un triangle quelconque  
  
;-----  
; Argument  
  
; (list): une liste de 3df  
  
;-----  
; Fonction appelée  
  
; « cal »: pour calculer l'angle  
  
;-----  
; Variables locales  
  
; pt1 : premier point de la liste  
  
; pt2 : deuxième point de la liste  
  
; pt3 : troisieme point de la liste  
  
; ang1: angle en pt1  
  
; ang2: angle en pt2  
  
; ang3: angle en pt3  
  
;-----  
; Retour  
  
; (listeangle): liste de tous les angles dans le triangle
```

```

;-----
(defun angle-triangle (3df)

  (setq pt1 (car 3df))

  (setq pt2 (cadr 3df))

  (setq pt3 (last 3df))

  (setq ang1 (cal "ang(pt1,pt2,pt3)"));calcule l'angle en pt1

  (setq ang2 (cal "ang(pt2,pt1,pt3)"));calcule l'angle en pt2

  (setq ang3 (cal "ang(pt3,pt1,pt2)"));calcule l'angle en pt3

  (setq listeangle (list ang1 ang2 ang3));renvoie une liste avec tous les angles

; du triangle

)

;*****
;*  M. Sc. Aménagement [CMFAO] *
;*  Anis Semlali      [GRCAO] 2000 *
;*****
;-----

; Fonction « angle3df »

; Retourne l'angle compris entre 0 et 90

;-----

; Argument

; (listeangle): liste de tous les angles dans le triangle

;-----

; Retour

; (listeangle-f): liste finale des angles

```

```

;-----
(defun angle3df (listeangle)

  (mapcar

    '(lambda (A)

      (cond

        ((< (abs (- A 315)) 0.01)

          (setq listeangle (subst 45 A listeangle))) ;remplace 315 par 45

        ((< (abs (- A 292.5)) 0.01)

          (setq listeangle (subst 67.5 A listeangle)));remplace 292.5 par 67.5

        ((< (abs (- A 270)) 0.01)

          (setq listeangle (subst 90 A listeangle))) ;remplace 290 par 90

        ); - cond

      ); - lambda

    listeangle ;liste du mapcar

  ); - mapcar

  (setq listeangle-f listeangle);renvoie une listeangle finale

); - defun

;*****
;*  M. Sc. Aménagement [CMFAO] *
;*  Anis Semlali      [GRCAO] 2000 *
;*****
;-----
; Fonction « dif-liste »

```



```
; Prend comme argument liste1 et liste2 pour retourner une nouvelle liste3 qui est
; liste1 - liste2
```

```
-----
```

```
(defun dif-liste (liste1 liste2)
```

```
  (setq liste3 '())
```

```
  (mapcar
```

```
    '(lambda (x)
```

```
      (if
```

```
        (= nil (member x liste2))
```

```
        (setq liste3 (cons x liste3))
```

```
      )
```

```
    )
```

```
  liste1
```

```
  )
```

```
  (setq liste1 (reverse liste3))
```

```
  )
```

```
*****
```

```
;*   M. Sc. Aménagement [CMFAO]   *
```

```
;*   Anis Semlali           [GRCAO] 2000   *
```

```
*****
```

```
-----
```

```
; Fonction « elimine-double »
```

```
; Élimine les éléments doubles dans une liste donnée
```

```
-----
```

; Argument

; (list): une liste de 3df

; Variables locales

; (liste-unique)

; Retour

; la même liste d'entrée

(defun elimine-double (3df)

(setq liste-unique '())

(mapcar

'(lambda (b)

(if

(= nil (member b liste-unique))

(setq liste-unique (cons b liste-unique))

)

)

3df

)

(setq 3df (reverse liste-unique))

)

```
*****
;* M. Sc. Aménagement [CMFAO] *
;* Anis Semlali [GRCAO] 2000 *
*****
;-----
;fonction « mil3d »
; Retourne le milieu de 2 points dont la coordonnée z est différente de 0
;-----
;Arguments
; pt1 et pt2
;-----
;Variables locales
; x1, x2, y1, y2, xmid et ymid
;-----
;Retour
; (milieu): le point milieu du segment pt1 et pt2
;-----
(defun mil2d (pt1 pt2)
  (setq x1 (car pt1) y1 (cadr pt1) )
  (setq x2 (car pt2) y2 (cadr pt2) )
  (setq xmid (/ (+ x1 x2) 2))
  (setq ymid (/ (+ y1 y2) 2))
  (setq milieu (list xmid ymid ))
)
```

```

;*****
;*  M. Sc. Aménagement [CMFAO] *
;*  Anis Semlali      [GRCAO] 2000 *
;*****
;-----
; Fonction « mil3d »
; Retourne le milieu de 2 points dont la coordonnée z est différente de 0
;-----
; Arguments
; pt1 et pt2
;-----
; Variables locales
; x1, x2, y1, y2, z1, z2,xmid, ymid et zmid
;-----
; Retour
; (milieu): le point milieu du segment pt1 et pt2
;-----
(defun mil3d (pt1 pt2)
  (setq x1 (car pt1 ) y1 (cadr pt1) z1 (last pt1))
  (setq x2 (car pt2 ) y2 (cadr pt2) z2 (last pt2))
  (setq xmid (/ (+ x1 x2) 2))
  (setq ymid (/ (+ y1 y2) 2))
  (setq zmid (/ (+ z1 z2) 2))
  (setq milieu (list xmid ymid zmid))

```

```

)
;*****
;*  M. Sc. Aménagement [CMFAO] *
;*  Anis Semlali      [GRCAO] 2000 *
;*****
;-----
; Fonction « nbr-max »
; Retourne le maximum de la liste (liste-d)
;-----
; Argument
; (list-d): liste des distances
;-----
; Variables locales
; loin: le point le plus éloigné du centre
;-----
; Retour
; (list-temp): liste qui retourne les points les plus loin du centre (1 ou 2 points)
;-----
(defun nbr-max (list-d)
  (setq loin (max (max (max (car list-d) (cadr list-d)) (caddr list-d)))); retourne
le max de list-d
  (setq list-temp '());commencer avec une (list-temp) nulle
  (mapcar
    '(lambda (m)

```

```

      (if
        (< (abs (- m loin )) 0.01);si c'est le point le plus loin ou presque (boge
osnap)

        (setq list-temp (cons m list-temp ))

        )

      )

list-d

)

(setq list-temp (reverse list-temp))

)

```

```

;*****

```

```

;* M. Sc. Aménagement [CMFAO] *

```

```

;* Anis Semlali [GRCAO] 2000 *

```

```

;*****

```

```

;-----

```

```

;Fonction « nbr-max2 »

```

```

; Retourne la valeur de la longueur de la liste (list-temp)

```

```

;-----

```

```

;Argument

```

```

; (list-temp): liste des points les plus loin du centre du 3df (1 ou 2 points)

```

```

;-----

```

```

;Variable locale

```

```

; loin: le point le plus éloigné du centre

```

```

;-----

```

; Retour

; d: nombre d'éléments dans la liste (list-temp)

(defun nbr-max2 (list-temp)

(setq d (length list-temp))

)

;* M. Sc. Aménagement [CMFAO] *

;* Anis Semlali [GRCAO] 2000 *

; Fonction « proche-zero »

; Annule les valeurs inférieures à 0.001

; Argument

; une liste d'un 3df

; Retour

; la même liste en annulant les valeurs proches de 0

(defun proche-zero (list);enregistré sous "proche-zero"

(setq pr '())

(mapcar

'(lambda (r)

```

(if
  (and (< r 0.001)(> r -0.001))
  (setq pr (cons 0 pr))
  (setq pr (cons r pr))
  )
)

list
)

(setq pr (reverse pr))
)

;*****
;*  M. Sc. Aménagement [CMFAO] *
;*  Anis Semlali      [GRCAO] 2000 *
;*****

;-----

; Fonction « pt-centre »

; Calcule la distance entre chaque point d'un 3df et le centre de la trame

;-----

; Argument

; (list): liste d'un 3df

;-----

; Variable locale

; centre: centre de la trame

;-----

```


; Retour

; (list-d): liste des distances de tous les points du 3df par rapport au centre de la

; trame

(defun pt-centre (3df)

 (setq centre (list 0 0 0)); centre du tracé

 (setq list-d '()); commencer avec une valeur de liste (list-d) nil

 (mapcar

 '(lambda (pt)

 (setq y (distance centre pt));distance pt/centre

 (setq list-d (cons y list-d))

)

 3df

)

 (setq list-d (reverse list-d))

)

;* **M. Sc. Aménagement [CMFAO]** *

;* **Anis Semlali [GRCAO] 2000** *

; Fonction « ptloin3 »

; Trouve le point le plus loin par rapport au centre de la trame d'un 3df triangle

; Argument

; une liste (3df)

; Retour

; le point le plus loin

(defun ptloin3 (3df)

(setq centre (list 0 0 0)); centre du tracé

(setq pt1 (car 3df))

(setq pt2 (cadr 3df))

(setq pt3 (last 3df))

(setq d1 (distance centre pt1))

(setq d2 (distance centre pt2))

(setq d3 (distance centre pt3))

(setq liste-distance (list d1 d2 d3))

(setq M (max (max (car liste-distance) (cadr liste-distance)) (last liste-distance)))

(setq ltotal '())

(mapcar

'(lambda (pt)

(if

(< (abs(- (distance centre pt) M)) 0.001)

(setq ltotal (cons pt ltotal))

)

```

)
3df
)
(setq ltotal (reverse ltotal))
);defun
;*****
;*  M. Sc. Aménagement [CMFAO] *
;*  Anis Semlali      [GRCAO] 2000 *
;*****
;-----
; Fonction « ptloin4 »
; Trouve le point le plus loin par rapport au centre de la trame d'un 3df
; quadrilatère
;-----
; Argument
; une liste (3df)
;-----
; Retour
; le point le plus loin
;-----
(defun ptloin4 (3df)
  (setq centre (list 0 0 0)); centre du tracé
  (setq pt1 (car 3df))
  (setq pt2 (cadr 3df))

```

```
(setq pt3 (caddr 3df))

(setq pt4 (last 3df))

(setq d1 (distance centre pt1))

(setq d2 (distance centre pt2))

(setq d3 (distance centre pt3))

(setq d4 (distance centre pt4))

(setq liste-distance (list d1 d2 d3 d4))

(setq M (max (max (max (car liste-distance) (cadr liste-distance)) (caddr
liste-distance)) (last liste-distance))); retourne le max de la liste (list-d)

(setq ltotal '())

(mapcar

'(lambda (pt)

  (if

    (< (abs(- (distance centre pt) M)) 0.001)

    (setq ltotal (cons pt ltotal ))

    )

  )

3df

)

(setq ltotal (reverse ltotal))

)
```

```
*****
```

```
;* M. Sc. Aménagement [CMFAO] *
```

```
;* Anis Semlali [GRCAO] 2000 *
```

```
*****
```

```
-----
```

```
;Fonction « TLISTE »
```

```
; Construit et sélectionne les 3df d'une liste s'ils contiennent un segment d'une
```

```
; liste
```

```
-----
```

```
(defun tliste (liste lseg)
```

```
  ;on cherche les 3df qui ont au moins un segment de la liste des segments
```

```
  (setq lfin '())
```

```
  (mapcar
```

```
    '(lambda (x)
```

```
      (setq temp (t3df x l3df))
```

```
      (if
```

```
        (/= temp nil)
```

```
        (setq lfin (append temp lfin))
```

```
      )
```

```
    )
```

```
  lseg
```

```
)
```

```
(setq lfin (elimine-double (reverse lfin)))
```

```
)
```

; Fonction « T3DF »

; Retient le ou les 3df d'une liste s'ils contiennent un segment donné

; Arguments

; seg le segment de référence

; (l3d) liste des 3df à analyser

; (lf) liste des 3df à construire

; Fonction appelée

; « 3DF2SEG »

; Variables locales

; (seg3df) : liste des segments de chaque 3df

; Résultat

; Reste de la liste des segments de seg3df si seg a été trouvé. Nul si non

; Retour

; (lf) : liste de 3df qui répondent à la condition

(defun t3df (seg l3df)

(setq lf '())

(mapcar

```
'(lambda (y)
  ;construction de la liste des segments du 3df
  (setq seg3df (3df2seg (list y)))
  (setq x (car seg))
  (setq y1 (last seg))
  ;vérification de l'existence du segde reference dans la liste
  (cond
    (
      (member seg seg3df)
      (setq resultat (member seg seg3df))
    )
    (
      (member (reverse seg) seg3df)
      (setq resultat (member (reverse seg) seg3df))
    )
  )
  (setq resultat nil)
)
(setq lf (cons y lf))
(progn
  (setq seg (reverse seg))
  (setq resultat (member seg seg3df))
  (if
    (/= resultat nil)
    (setq lf (cons y lf))
```

```

)
)
)
;)
l3df
)
(setq lf (reverse lf))
)
;*****
;*  M. Sc. Aménagement [CMFAO] *
;*  Anis Semlali      [GRCAO] 2000 *
;*****
;-----
; Fonction « trouve »
; Trouve tous les 3df qui ont un point en commun
;-----
; Argument
; la liste des 3df
;-----
; Retour
; la liste (lseg)
;-----
(defun trouve (pt liste)
  (setq x (car pt) y (cadr pt))

```



```
(setq selection '())  
  
(mapcar  
  
'(lambda (3df)  
  
  (mapcar  
  
    '(lambda (ptx)  
  
      (if  
  
        (and  
  
          (<= (abs(- (car ptx) x)) 0.00001)  
  
          (<= (abs (- (cadr ptx) y)) 0.00001)  
  
        )  
  
      (setq selection (cons 3df selection))  
  
    )  
  
  )  
  
  3df  
  
  )  
  
  )  
  
  liste  
  
  )  
  
(setq selection selection)  
  
)
```

```
*****
```

```
;* M. Sc. Aménagement [CMFAO] *
```

```
;* Anis Semlali [GRCAO] 2000 *
```

```
*****
```

```
-----
```

```
; Fonction « z-off »
```

```
; Annule la coordonnée en z
```

```
-----
```

```
; Argument
```

```
; une liste de 3df
```

```
-----
```

```
; Retour
```

```
; la même liste sans la coordonnée en z
```

```
-----
```

```
(defun z-off (liste)
```

```
  (mapcar
```

```
    '(lambda (3df)
```

```
      (mapcar
```

```
        '(lambda (pt)
```

```
          (setq pt (list (car pt) (cadr pt)))
```

```
        )
```

```
      3df
```

```
    )
```

```
  )
```

liste

)

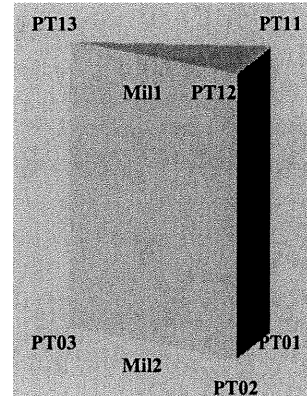
)

;------

Annexe1 Construction de la pièce « Dembouq »

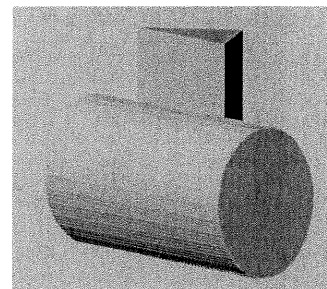
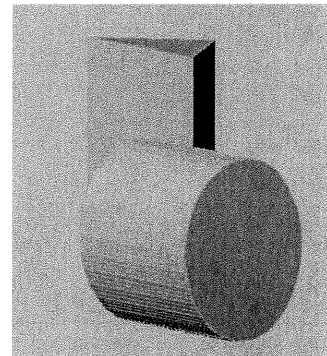
Construction du prisme :

- Dessiner le pline reliant pt01, pt02 et pt03
- Extrusion du pline de 200 unités
- Définir les points pt11, pt12 et pt13
- Définir mil1 le milieu de pt12 et pt13
- Définir mil2 le milieu de pt02 et pt03
- Nommer pt12 le point au-dessus de pt02



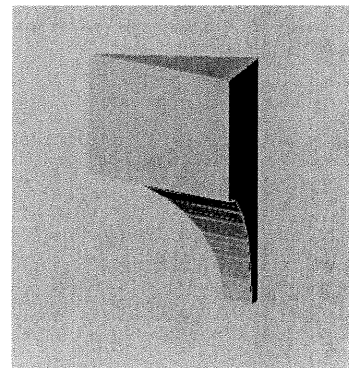
Construction des ellipses :

- Définir un nouveau UCS de centre mil2 et de direction pt01 et mil2
- Construire l'ellipse de centre (0 0 0) et de sommet sur le petit axe le point de coordonnées (80 0 0) et de sommet sur le grand axe le point de coordonnées (0 100 0)
- Extruder cette ellipse de 200
- Reconstruire la même ellipse
- L'extruder de -200



Construction de la pièce :

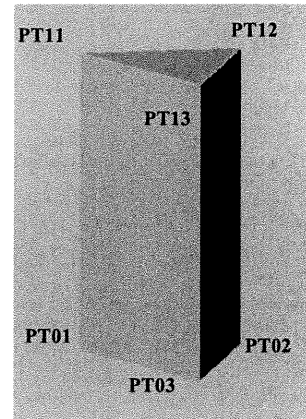
Il suffit d'enlever du prisme les 2 cylindres elliptiques pour obtenir la pièce « Dembouq »



Annexe 2 : : Construction de la pièce « Loza1 »

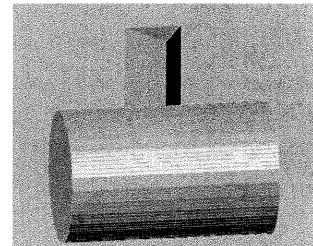
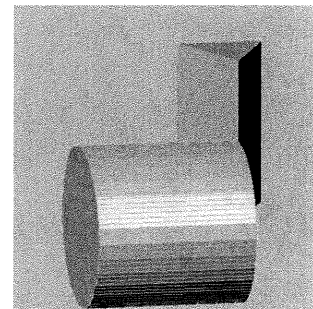
Construction des faces du prisme :

- construire le pline qui lie pt01 pt02 pt03
- extruder ce pline d'une hauteur 200
- nommer pt11 le point au-dessus de pt01
- nommer pt12 le point au-dessus de pt02



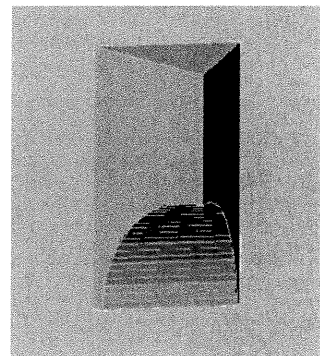
Construction des ellipses :

- Nommer mil1= pt11 * pt12
- Nommer mil2 = pt01* pt02
- Définir un nouveau UCS de centre mil2 et de direction pt03 et mil1
- Dans le nouveau repère, construire l'ellipse de centre (0 0 0) et qui passe par les points (120.655 0 0) et (0 100 0)
- Extruder cette ellipse de 200



Construction de la pièce :

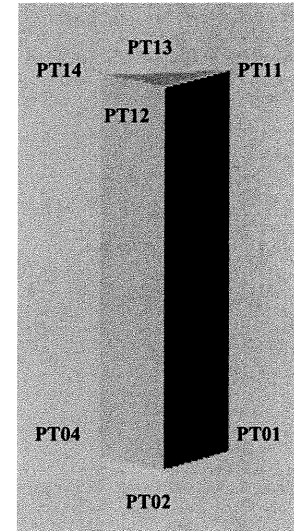
il suffit d'enlever du prisme les 2 ellipses extruder et on aura la pièce « Loza1 »



Annexe 3: Construction de la pièce « Ktaf »

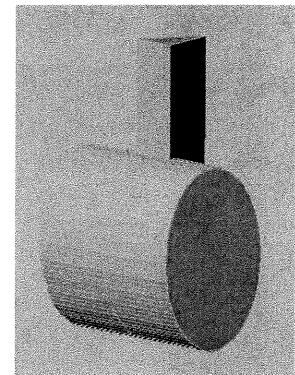
Construction du prisme :

- Définir mil1 le milieu de pt12 et pt14
- Définir mil2 le milieu de pt02 et pt04
- Dessiner le pline reliant pt01, pt02 et pt04
- Extrusion du pline de 400 unités
- Définir les points pt11, pt12, pt13 et pt14
- Dessiner le pline reliant pt03, pt02 et pt04
- Extrusion du pline de 200 unités



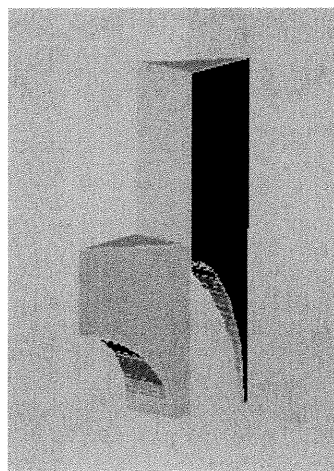
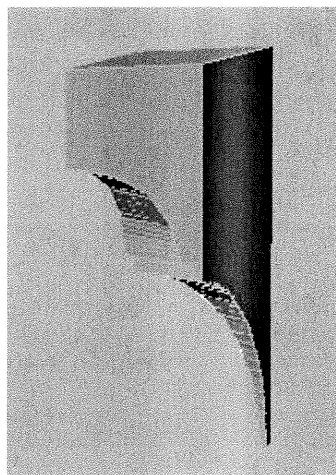
Construction des ellipses :

- Définir un nouveau UCS de centre mil2 et de direction pt02 et mil1.
- Construire l'ellipse de centre (0 0 0) et de sommet sur le petit axe le point de coordonnées (0 180 0) et de sommet sur le grand axe le point de coordonnées (120.655 0 0)
- Extruder cette ellipse de 200
- Redessiner la même ellipse
- L'extruder de -200
- Définir un nouveau UCS de centre pt03 et de direction pt01 et pt13.
- Construire l'ellipse de centre (0 0 0) et de sommet sur le petit axe le point de coordonnées (0 100 0) et de sommet sur le grand axe le point de coordonnées (110.655 0 0)
- Extruder cette ellipse de 200
- Nommer cyl03 le dernier cylindre elliptique dessiner.
- Redessiner la même ellipse
- L'extruder de -200
- Nommer cyl04 le dernier cylindre elliptique dessiner.



Construction de la pièce :

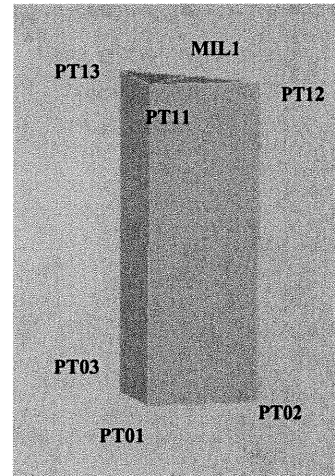
À l'aide de l'opérateur booléen « soustraction », on enlève du volume du prisme pri01 les deux volumes des cylindres elliptiques los1 et los2 et du volume du prisme pri02 les deux volumes des cylindres elliptiques los3 et los4 puis enfin unir les 2 volumes obtenues pour obtenir la pièce « ktaf ».



Annexe 4: Construction de la pièce « Serwaliya »

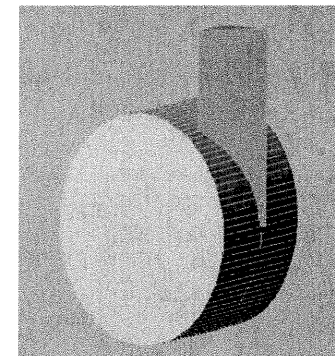
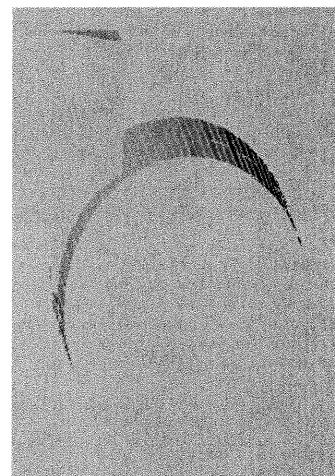
Construction du prisme :

- Définir mil1 le milieu de pt12 et pt13
- Définir mil2 le milieu de pt02 et pt03
- Dessiner le pline reliant pt01, pt03 et mil2
- Extrusion du pline de 400 unités
- Dessiner le pline reliant pt01, pt02 et mil2
- Extrusion du pline de 400 unités



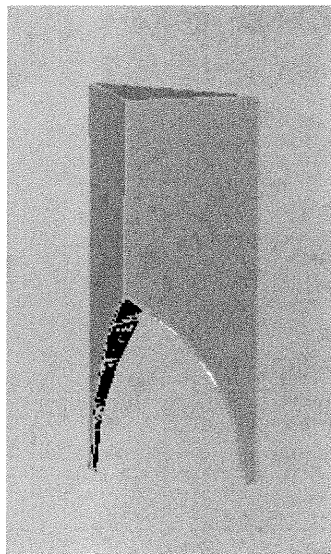
Construction des ellipses :

- Définir un nouveau UCS de centre pt02 et de direction pt12 et pt03
- Construire l'ellipse de centre (0 0 0) et de sommet sur le petit axe le point de coordonnées (0 190 0) et de sommet sur le grand axe le point de coordonnées (235.25 0 0)
- Extruder cette ellipse de 150
- Reconstruire la même ellipse
- L'extruder de -150



Construction de la pièce :

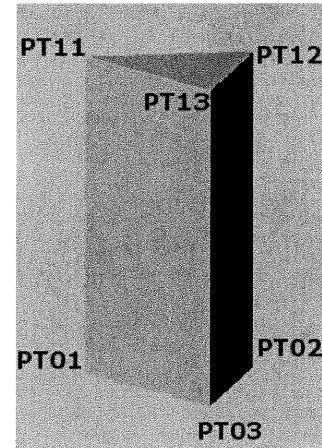
À l'aide de l'opérateur booléen « soustraction », on enlève du volume du prisme les deux volumes des cylindres elliptiques pour obtenir la pièce « sewaliya ».



Annexe 5 : Construction de la pièce « Charbiya »

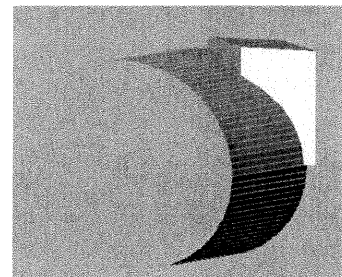
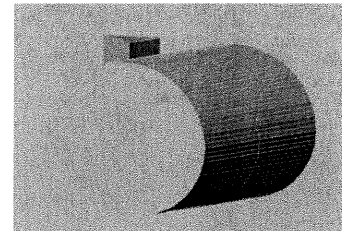
Construction du prisme :

- Définir mil1 le milieu de pt12 et pt14
- Définir mil2 le milieu de pt02 et pt04
- Dessiner le pline reliant pt01, pt02 et pt03
- Extrusion du pline de 400 unités
- Définir les points pt11, pt12, pt13 et pt14
- Dessiner le pline reliant pt01, pt03 et pt04
- Extrusion du pline de 200 unités



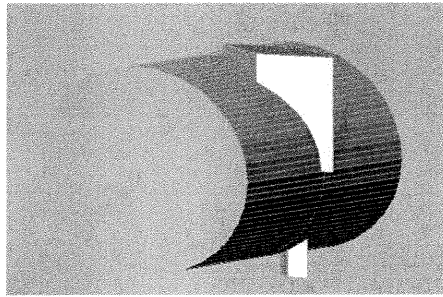
Construction des ellipses :

- Définir un nouveau UCS de centre pt04 et de direction pt02 et pt14.
- Construire l'ellipse de centre (0 0 0) et de sommet sur le petit axe le point de coordonnées (0 180 0) et de sommet sur le grand axe le point de coordonnées (241.3101 0 0)
- Extruder cette ellipse de 200
- Redessiner la même ellipse
- L'extruder de -200
- Définir un nouveau UCS de centre pt02 et de direction pt04 et pt12.
- Construire l'ellipse de centre (0 0 0) et de sommet sur le petit axe le point de coordonnées (0 180 0) et de sommet sur le grand axe le point de coordonnées (241.3101 0 0)
- Extruder cette ellipse de 200
- Redessiner la même ellipse
- L'extruder de -200

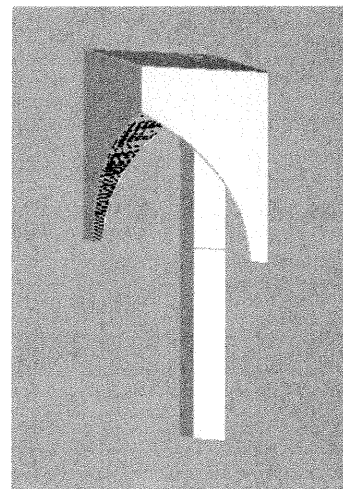
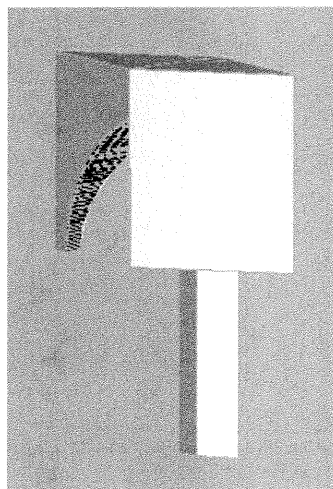


Construction des pieds de la pièce :

Ici, on divise la base triangulaire du pied de la pièce en 2 triangles qui auront en commun le segment formé par les points pt01 et centre-pied. Les autres points seront « pd » qui est le point situé à droite et « pg » situé à gauche.

**Construction de la pièce :**

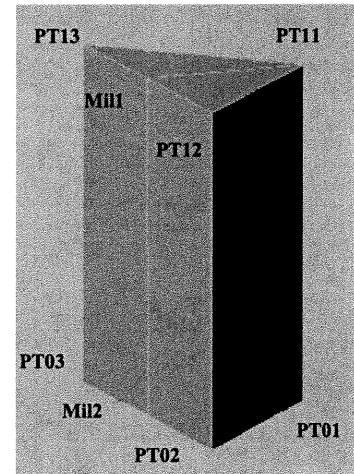
À l'aide de l'opérateur booléen « soustraction », on enlève du volume du prisme pri01 les deux volumes des cylindres elliptiques los1 et los2 et du volume du prisme pri02 les deux volumes des cylindres elliptiques los3 et los4 puis enfin unir les 2 volumes obtenues



Annexe 6 : : Construction de la pièce « Serwaliya Sguira »

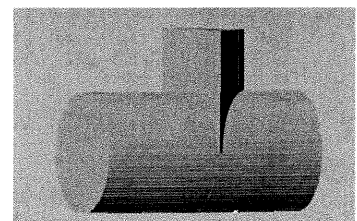
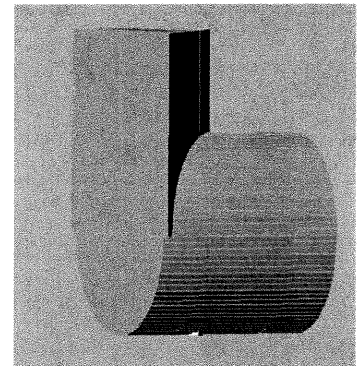
Construction du prisme :

- Définir mil1 le milieu de pt12 et pt13
- Définir mil2 le milieu de pt02 et pt03
- Dessiner le pline reliant pt01, pt02 et mil2
- Extrusion du pline de 200 unités
- Nommer prisme1 le prisme obtenu
- Dessiner le pline reliant pt01, pt02 et mil2
- Extrusion du pline de 200 unités
- Nommer prisme2 le prisme obtenu



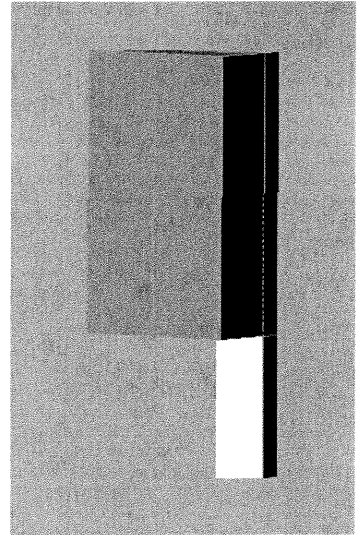
Construction des ellipses :

- Définir un nouveau UCS de centre mil2 et de direction pt02 et mil1
- Construire l'ellipse de centre (0 0 0) et de sommet sur le petit axe le point de coordonnées (90 0 0) et de sommet sur le grand axe le point de coordonnées (0 100 0)
- Extruder cette ellipse de 200
- Nommer le cylindre elliptique obtenu cyl1
- Reconstruire la même ellipse
- L'extruder de -200
- Nommer le cylindre elliptique obtenu cyl2



Construction du pied de la pièce :

- Définir les coordonnées x_p , y_p et z_p du centre du premier pied de la pièce (centre-pied).
- Revenir à l'UCS initial
- À l'aide de la commande « polygon » dessiner le carré de coté 25.
- Placer le pied sur la pièce
- L'extruder de 200
- Faire la même chose pour la deuxième partie de la pièce et l'extruder de -100



Construction de la pièce :

- À l'aide de l'opérateur booléen « soustraction », on enlève du volume du prisme les deux volumes des cylindres elliptiques pour obtenir la pièce « dembouq ».
- À l'aide de la commande « union », unir le dernier volume obtenu avec les pieds de la pièce

