

Rapport de Cours

AME-6912

Études et recherches dirigées II

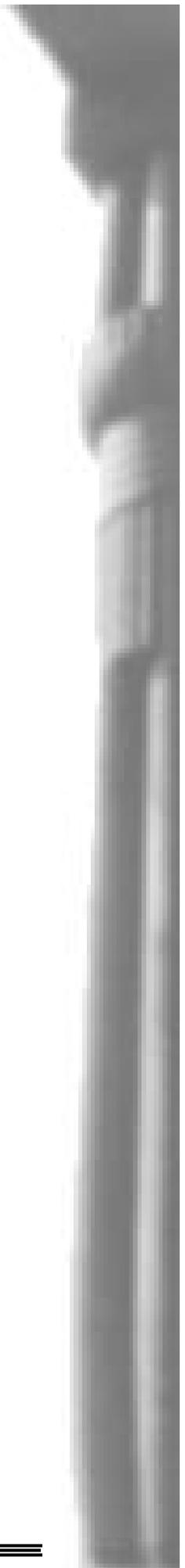
Responsable : Claude Parisel.

**Modélisation volumique
avec
SGDLsoft:**

la représentation de
la colonne égyptienne papyriforme

Hassoun Karam

Septembre 1999.



Plan du Rapport

1. Introduction
2. Les Objectifs
3. La Méthodologie d'étude
4. L'approche de la Modélisation
 - 4.1 Objectifs
 - 4.2 Directives
5. La modélisation
 - 5.1 Le Chapiteau
 - 5.2 Le Fût
 - 5.3 La base
6. Des Considérations Techniques
7. Conclusion

1. Introduction.

Ce rapport est rédigé en conclusion d'un stage pratique au Groupe de Recherche en Conception Assistée par Ordinateur, à la Faculté de l'aménagement, à l'Université de Montréal, du 16 Août au 17 septembre 1999.

Ce stage se situe dans la continuité des travaux entrepris dans le cadre de ma maîtrise sur la Représentation des Connaissances architecturales et ses outils numériques, et portant sur les langages de programmation fonctionnelle, en particulier Scheme, ainsi que SGDLsoft comme interface de modélisation volumique.

2. Les Objectifs:

Dans le cadre de ce cours, j'ai défini les objectifs suivants:

- 1- Connaître les concepts de base de la modélisation volumique;
- 2- Se familiariser avec la programmation en Scheme;
- 3- Se familiariser avec l'interface de SGDLsoft;
- 4- Modéliser une colonne égyptienne papyriforme, comme application pratique.

Il s'agissait d'établir, dans un temps défini, un niveau d'apprentissage élémentaire, qui me servira comme une base pour un perfectionnement ultérieur. La manipulation de nouveaux outils de la CAO est un processus d'essais et d'erreurs en fonction des problèmes envisagés et de l'optimisation des approches utilisées. Ainsi, pour être efficace, ce processus doit profiter des acquisitions et des compétences antérieures, et les adapter à ces problèmes.



3. La Méthodologie d'étude:

J'ai suivi le cheminement suivant:

- 3.1. Une recherche bibliographique pour identifier les ressources disponibles concernant l'architecture et l'archéologie égyptienne (les références sont citées à la fin de ce rapport).
- 3.2. Une lecture des ouvrages qui traitent des méthodes de constructions en Égypte ancienne et surtout des règles géométriques de conception architecturales, afin d'identifier les propriétés inhérentes à ce style de colonne, ainsi que les considérations esthétiques en jeu.
- 3.3. Des exercices de programmation en Scheme pour me familiariser avec ce langage et adapter sa grammaire et sa syntaxe aux problèmes de modélisation posés dans SGDLsoft.
- 3.4. Des applications pratiques en SGDLsoft tirées du manuel de l'utilisateur pour me familiariser avec cette interface.

4. L'Approche de la modélisation :

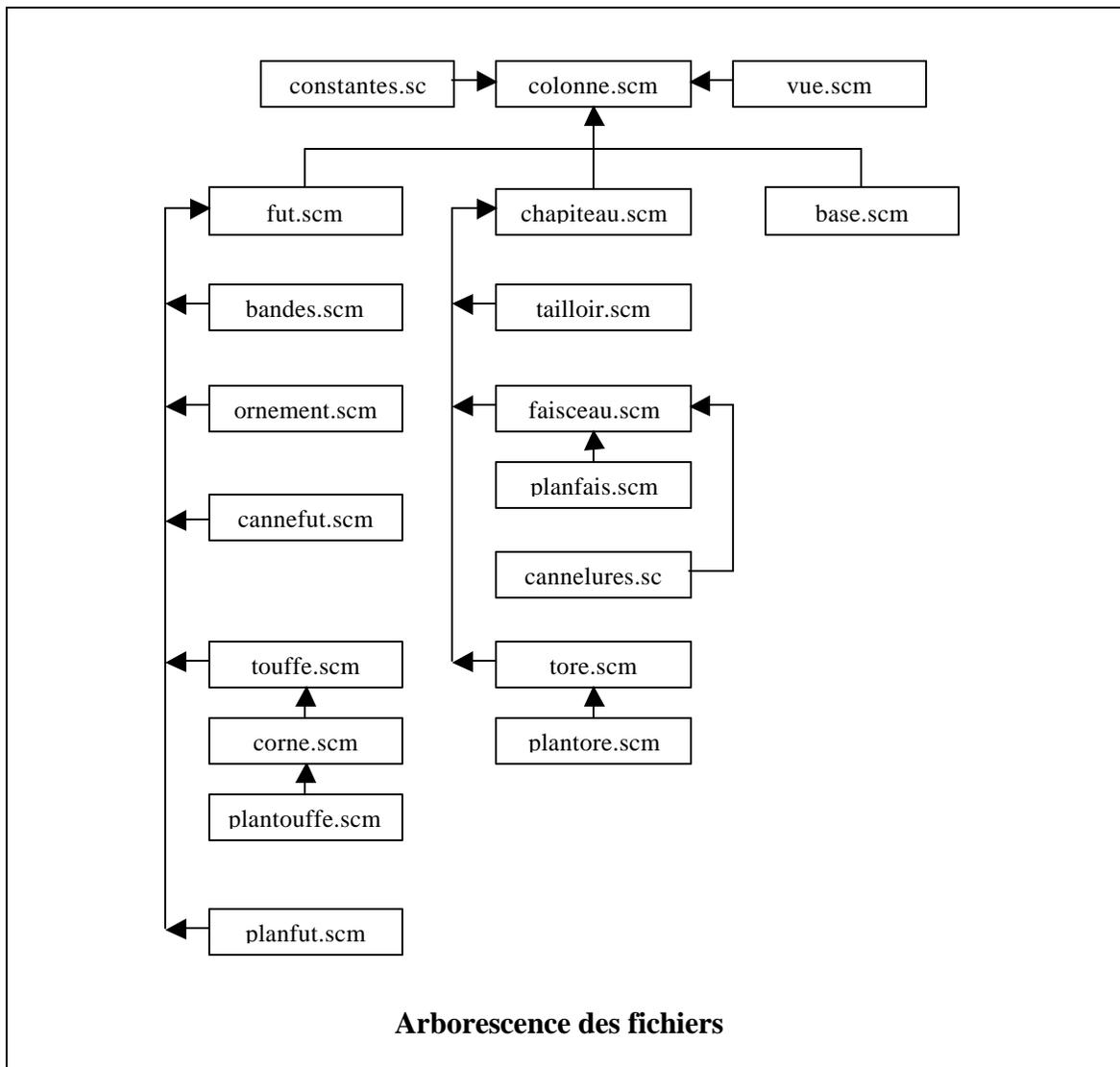
Ma stratégie de modélisation s'articule autour de trois axes principaux : l'ouverture des modules géométriques, la réutilisation des fonctions, l'arborescence acyclique des fichiers.

4.1. Objectifs de l'approche :

Cette approche a les objectifs suivants :

4.1.1. Faciliter la compréhension de l'interface entre la colonne et sa représentation numérique.

4.1.2. Intégrer une flexibilité dans le code qui appuiera les rectifications et les développements ultérieurs.



4.2. Directives de l'approche :

- J'ai utilisé deux types de mise en relation des fichiers afin d'expérimenter avec les possibilités de Scheme et SGDLsoft. La première est l'intégration du fichier enfant dans le fichier parent et par la suite l'intégration de ce dernier dans le grand-parent (voir l'exemple de touffe.scm dans l'arborescence). La seconde est l'intégration de tous les fichiers enfants dans le parent à un seul niveau hiérarchique (voir l'exemple de faisceau.scm).

- Mon fichier maître est colonne.scm (voir l'arborescence ci-dessus) qui incorpore les fichiers de génération des composantes ainsi que ceux de la visualisation et des paramètres constants et globaux. Tous les fichiers furent testés et les procédures furent déboguées individuellement. Les relations exprimées par l'organigramme ci-dessus représentent des inclusions dans le sens des flèches : par exemple le fichier bandes.scm est inclus dans fut.scm, qui à son tour est inclus dans colonne.



- Le fichier «constantes.scm» contient toutes les variables et les constantes globales utilisées dans les autres fichiers. En plus, dans chacun de ces derniers, les variables et les constantes nécessaires au fonctionnement des procédures sont insérées mais commentées par un «;». Cette approche permet la vérification de chaque fichiers indépendamment des autres. De même pour les paramètres de visualisation propres à chaque fichier.

- Le fichier «vue.scm» contient les paramètres de visualisation de la colonne, qui pourraient être modifiés afin de restituer les différents points de vue relatifs aux images prises dans ce rapport.

- Sur un niveau fonctionnel, j'ai identifié deux niveaux hiérarchiques de la structure de la colonne : les fonctions singulières et les fonctions récurrentes. Les premières sont celles qui ne se répètent pas et s'appliquent à des objets essentiellement différents, les secondes sont celles qui se trouvent dans plusieurs éléments architecturaux. Cette identification facilite la réutilisation du code.

- Tous les objets ont été générés centrés sur le point (0,0,0,1) puis déplacés à leur emplacement par des opérateurs de translation et/ou rotation.

5. La Modélisation.

5.1. Analyse systémique de la colonne :

j'ai considéré la colonne comme un système de connaissances matérialisé dans un agencement de formes. Les techniques de construction chez les Égyptiens sont un facteur déterminant dans la perception des modules de ce système (Plommer 1956). A travers les lectures et les recherches sur le web, j'ai cerné deux autres facteurs : les connaissances mathématiques et en particulier la géométrie, et les règles d'esthétique et d'harmonie qui ordonnent l'application de ces premières. Mon étude des proportions et mes graphiques en découlent.

5.2. Définition des composantes:

La colonne est composée de 3 parties principales (Choisy 1983) :

- 1- le chapiteau qui comprend un tailloir, un faisceau de cannelures, et une base.
- 2- le fût constitue par un cône tronqué de bandes horizontales, un cône d'ornementations verticales, le fût proprement dit, et une base qui représente une touffe de tiges de papyrus. La considération de la touffe comme faisant partie propre du fût est contestée (Lloyd 1986 et Fletcher 1961) par le rapport de la hauteur totale des trois composantes précédentes a celle de la colonne, qui est : $\text{hauteur_fut} / \text{hauteur_colonne} = 5/8$.
- 3- la base de la colonne qui est composée par un volume cylindrique.



5.3. Proportions :

Le rapport de la hauteur de la colonne au grand diamètre du fût, varie entre 1/6 et 1/7 (Benoît 1911); j'ai choisi le premier rapport car il est le plus usité. En plus, le facteur déterminant dans la conception d'une colonne est sa hauteur, et toutes les autres dimensions doivent en découler. (Voir figure 1)

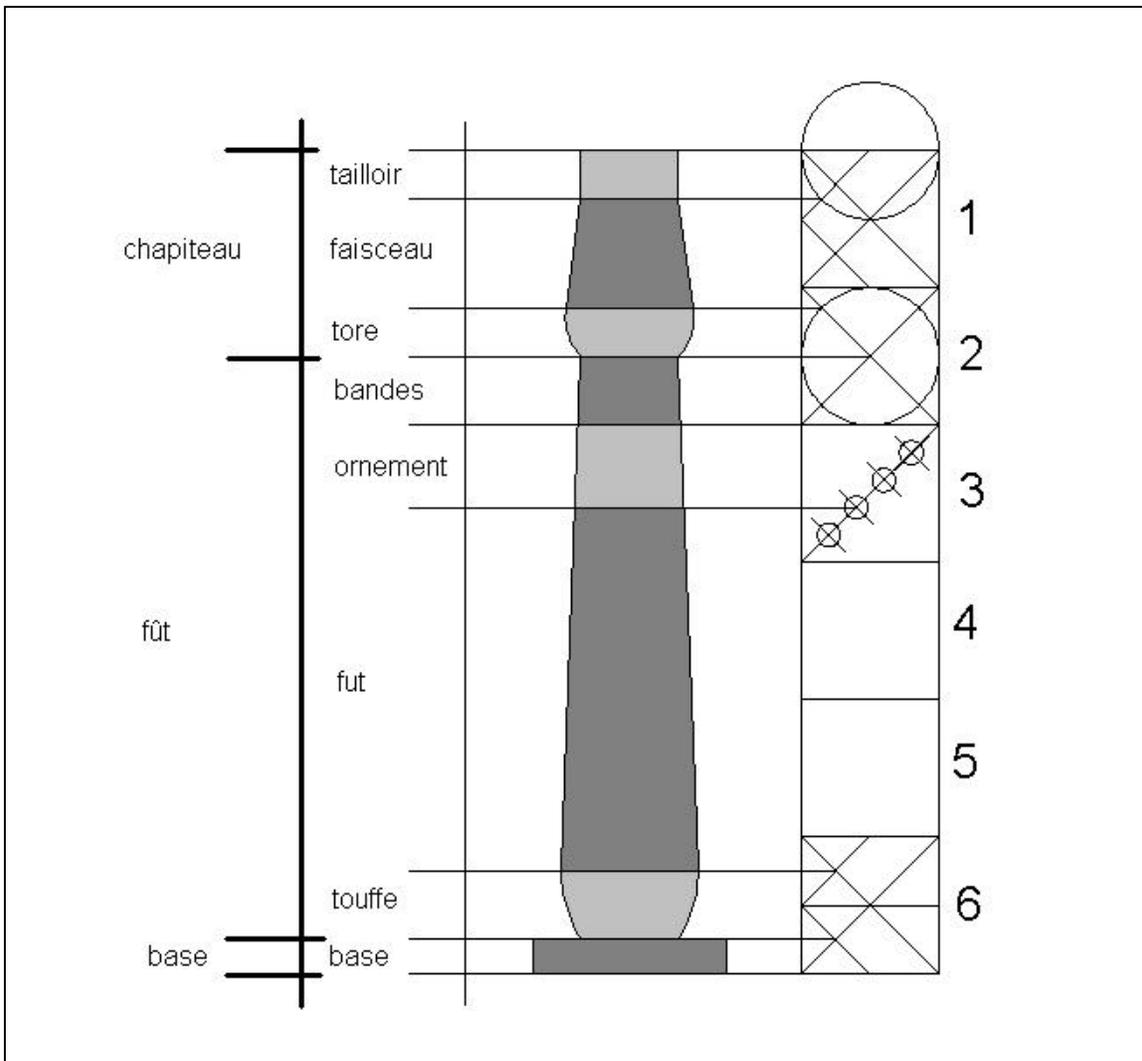


Fig.1. Proportions des composantes

L'agencement des composantes en plan s'inscrit dans trois cercles concentriques : de la base, du grand périmètre et du petit périmètre de la colonne. Le rapport des rayons (ou diamètres) pris consécutivement deux à deux est de l'ordre de : $\text{grand_diametre}/\text{petit_diametre} = \sqrt{2}$. (voir Fig.2)

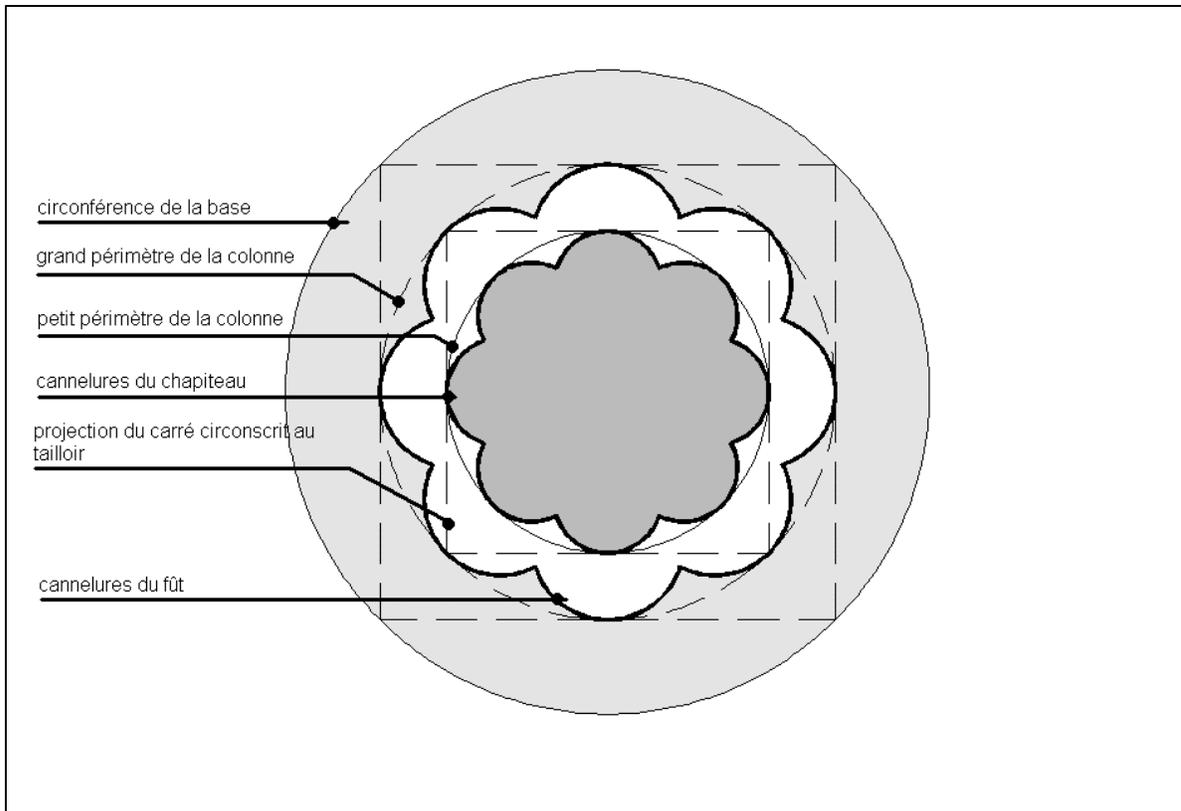
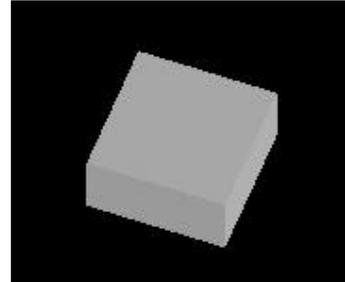


Fig.2. Proportions en plan

5.1. Le chapiteau :

Le chapiteau se compose de trois éléments : un tailloir, un faisceau de cannelures, et une base.

5.1.1. Le tailloir (tailloir.scn): formé par un volume parallélépipédique, de plan carré inscrit dans un cercle de diamètre équivalent à celui de la colonne (Benoît 1911) . A son tour, ce plan définit un cercle qui lui est inscrit (voir Fig. 2). Sa hauteur varie entre $\frac{1}{2}$ et 1 rayon de ce cercle; j'ai choisi la dernière.



```
(define tailloir
  (lambda ()
    (DLatt
      (SDcolRGB (vector 255 255 255 255))
      (SDmatrep (SGmattrl (vector 0 (-(/ hauteur_col 2)/ petit_ray 2)) 0 1)))
      (GDhxahxa
        (vector petit_ray (/ petit_ray 2) petit_ray 1)
        (vector
          (vector (- petit_ray petit_ray) (- petit_ray petit_ray) (- petit_ray petit_ray) 1)
          (vector 1 0 0 0)
          (vector 0 0 1 0)
          (vector 0 1 0 0) ) ) ) ) ) )
```

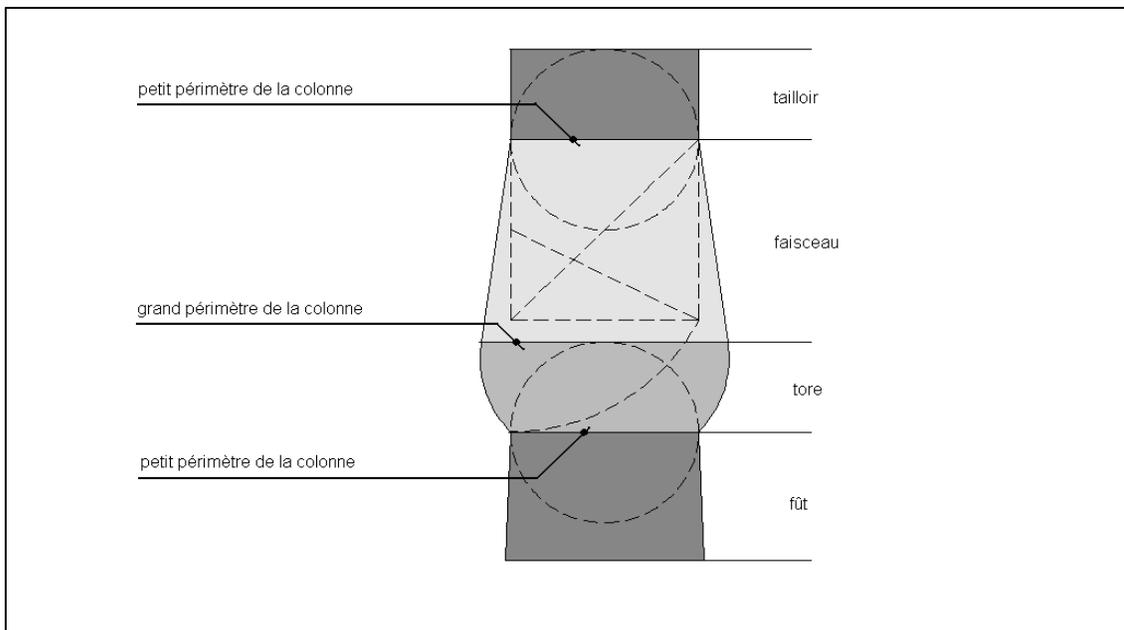
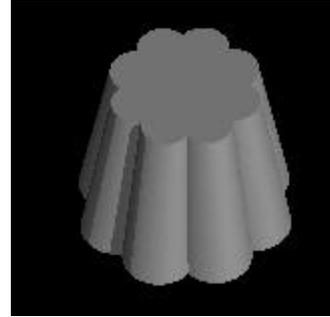


Fig. 3. Proportions du Chapiteau

5.1.2. Le faisceau (faisceau.scm): se compose d'un nombre pair de cannelures, 8 dans le cas étudié (Benoît 1911). Le faisceau a une forme conique tronquée, de section circulaire, large dans sa partie inférieure et plus réduite dans la supérieure. Le petit diamètre est égal a celui inscrit dans le tailloir et le grand est celui de la colonne. Le rapport de ces deux diamètres est égal a : $\text{grand_diametre}/\text{petit_diametre} = \sqrt{2}$. Ce faisceau est défini par une multiplication de l'ensemble des cannelures par un plan horizontal infini.



```
(define faisceau
  (lambda ()
    (DLatt
      (SDcolRGB (vector 255 255 255 255))
      (SDmatrep
        (SGmattrl
          (vector 0 (-(/ hauteur_col 2)(+/ hauteur_faisceau 2) petit_ray)) 0 1) ) )
      (DLmul
        (ensemble canne) (planfais) ) ) ) )
```

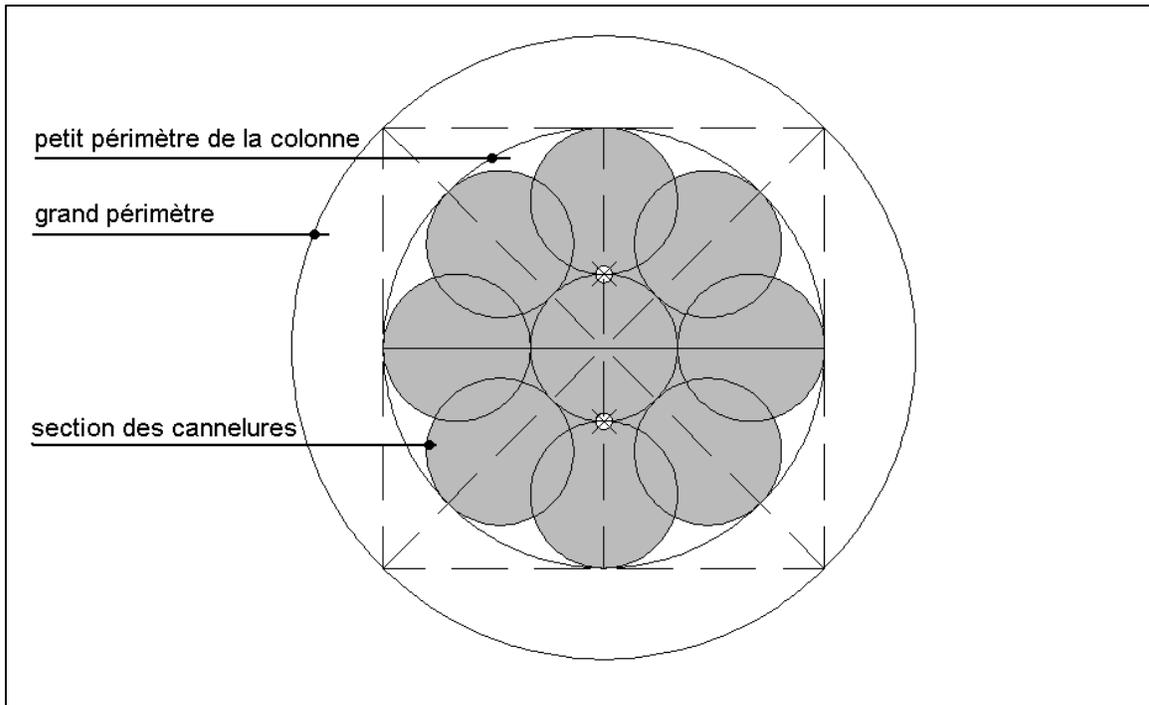
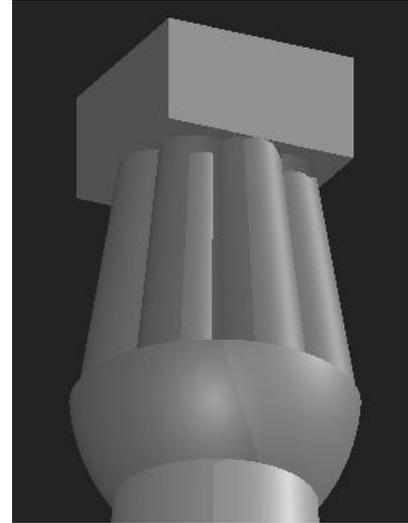


Fig. 4. Agencement en plan des sections des cannelures

Les cannelures sont des coniques obliques tronquées (cannelures.scm), de section circulaire, dont le diamètre supérieur est égal au tiers du petit diamètre de la colonne (voir Fig. 4). Le diamètre inférieur est approximativement égal tiers du grand diamètre. Cette section a évolué d'un tracé presque triangulaire vers un tracé circulaire (Cenival 1964). Les conséquences de cette évolution furent la disparition de l'arête verticale aiguë de la cannelure et l'accentuation de l'effet visuel courbe et fluide.

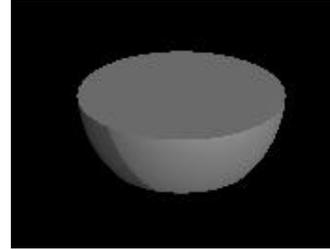


Ce fichier génère les huit cannelures par un appel récursif à une fonction (canne) qui produit une conique, et l'angle de rotation «ang» est incrémenté à chaque appel. L'emplacement des 8 sections circulaires est tangent à la circonférence des deux cercles définis par leurs diamètres respectifs (voir Fig. 4).

```
(define ensemble ; c'est l'ensemble des cannelures
(lambda (canne)
(DLatt
(SDcolRGB (vector 255 255 255 255))
(SDmatrep (SGmattrl (vector 0 (/ hauteur_faisceau 2) 0 1)))
(DLuni (DLformz2 ; definition du cone central de remplissage
(...
(let f
((ls(list))
(acc 0)
(ang 0))
(if(= acc nb)
(apply DLuni ls)
(f
(cons (canne ang)ls)
(+ acc 1)
(+ ang/( * pi 2)nb)) ))))))))
```

La hauteur du faisceau est fonction de son petit diamètre et le nombre d'or (1,6180333...): hauteur_faisceau = (petit_diametre * nb_d_or) - (petit_diametre / 2). Cette partie étant vide à l'intérieur, j'y ai intégré un cône central afin d'avoir une homogénéité de la densité. (voir plus loin, les cannelures du fût)

5.1.3. La base du chapiteau(tore.scm): c'est une forme hybride qui varie d'une période stylistique a une autre, mais qui se rapproche d'une calotte sphérique tronquée dont le diamètre supérieur est égal au grand diamètre de la colonne, et l'inférieur est égal au petit.



Elle est définie par la multiplication d'un plan horizontal infini (plantore.scm) par une quadrique sphérique. Sa hauteur est égale au petit rayon (c'est la hauteur du plan horizontal). La variable « psi » (voir fichier constantes.scm) sert à ajuster la valeur du vecteur de "SGmattrl" de la procédure "elli", afin d'avoir un profile qui représente le plus exactement possible l'objet réel.

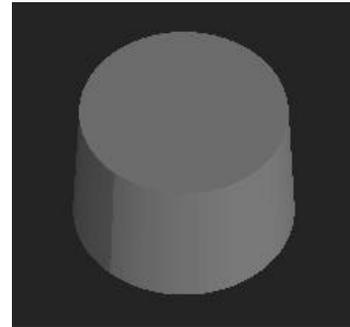
```
(define elli
  (lambda ()
    (DLatt
     (SDmatrep
      (SGmattrl
       (vector 0 (- petit_ray grand_ray psi) 0 1) ) )
      (DLformz2 ... )

(define tore
  (lambda ()
    (Dlatt ...
     (DLMul
      (elli)
      (plantore) ) ) ) )
```

5.2. Le fût :

Le fût est subdivisé en quatre composantes (voir Fig. 1): la composante supérieure formée par un ensemble de bandes horizontales, en dessous de laquelle se trouve la composante médiane qui représente des motifs d'ornementation verticaux, la troisième partie est le fût proprement dit, et la quatrième est la touffe. Le fût comprend 8 cannelures qui en forment le corps principal, et qui se prolongent conceptuellement dans le chapiteau (ou vice-versa).

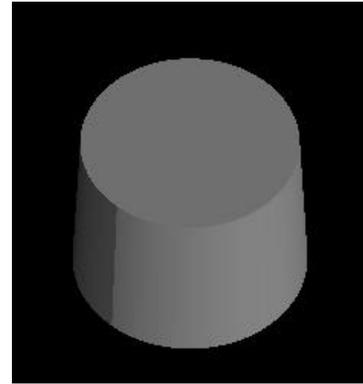
5.2.1. Les bandes (bandes.scm): c'est un cône tronqué dont le diamètre de la section supérieure est égal au petit diamètre de la colonne. Sa hauteur est égale à la moitié du grand diamètre. Les bandes sont gravées dans la pierre en un bas-reliefs et ne constituent pas des entités architecturales indépendantes.



Dans le code ci-dessous la coordonnée fuite_fut sert à ajuster l'angle de l'inclinaison des parois du cône.

```
(define bandeau
(lambda ()
(DLatt
(SDcolRGB (vector 255 0 255 255))
(SDmatrep
(SGmattrl
(vector 0 (-(/ hauteur_fut 2) (/ hauteur_band 2)) 0 1)
)
)
)
(GDcylhxa
(vector petit_ray (/ hauteur_band 2) petit_ray 1)
(vector
(vector 0 (-(/ hauteur_band 2) (/ hauteur_band 2)) 0 1)
(vector 4 0 0 0)
(vector 0 0 4 0)
(vector 0 fuite_fut 0 1) ) ) ) )
```

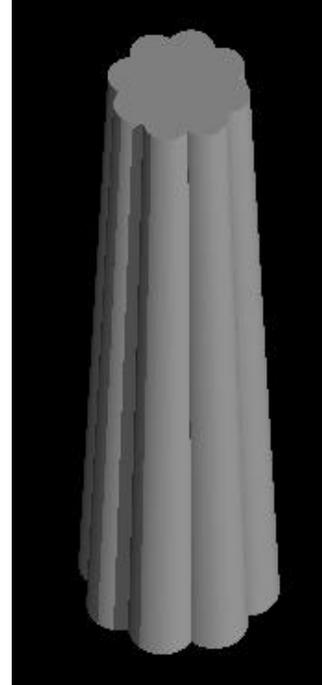
5.2.2. Les ornements (ornement.scm): c'est un cône tronqué dont les diamètres consécutifs des sections supérieure et inférieure sont en rapport homothétique avec celui des bandeaux. Sa hauteur est égale au $\frac{3}{5}$ du grand diamètre. Comme les bandes, les ornements sont des motifs taillés en bas-reliefs.



```
(define ornement
(lambda ()
(DLatt
(SDcolRGB (vector 255 255 255 255))
(SDmatrep
(SGmattrl
(vector 0 (- (/ hauteur_fut 2) (+ grand_ray (/ hauteur_orne 2))) 0 1)
)
)
(GDcylhxa
(vector orne_ray (/ hauteur_orne 2) orne_ray 1)
(vector
(vector 0 (- (/ hauteur_orne 2) (/ hauteur_orne 2)) 0 1)
(vector 4 0 0 0)
(vector 0 0 4 0)
(vector 0 fuite_fut 0 1) ) ) ) ) ) )
```

5.2.3. Les cannelures du fût (cannefut.scm): cette partie est topologiquement identique au faisceau du chapiteau (Choisy 1983) et n'en diffère que par sa hauteur. Cette partie étant vide à l'intérieur, j'y ai intégré un cône central (la quadrique Diformz2 dans le cadre ci-dessous) afin d'avoir une homogénéité de la densité.

```
(define ensemblefut ; c'est l'ensemble des cannelures
(lambda ...
(DLformz2 ; cette quadrique produit un cone pour combler
l'interieur de l'ensemble des cannelures
(vector
(vector (- (/ petit_ray 2)) 0 0 1)
(vector (/ petit_ray 2) 0 0 1)
(vector (- petit_ray petit_ray) fuite_fut 0 1) ; la
modification de x permet de deplacer le sommet
(vector 0 0 1 0)
(vector (- petit_ray petit_ray) fuite_fut 0 1) ; la
modification de x permet de deplacer le sommet
(vector 0 0 (/ petit_ray 2) 1)...))
```



5.2.4. La touffe (touffe.scm): cette partie est en continuité avec les cannelures du fût. Son module de base est un ellipsoïde oblique tronqué (corne.scm) qui subit 8 rotations et instantiations . L'usage de 8 modules au lieu de 4 fut utilisé pour reproduire par

```
(define touffe ; c'est l'ensemble des cornes
(lambda (corne)
(Dlatt ...
(let f
(
(ls(list))
(acc 0)
(ang 0)
)
(if(= acc nb)
(apply DLuni ls)
(f
(cons (corne ang)ls)
(+ acc 1)
(+ ang/( * pi 2)nb),....))
```



l'asymétrie de la corne, le profile exact de cette partie du fût et raccorder avec la plus grande précision possible la touffe avec le fût. L'exactitude est possible sur une petite partie de la circonférence de la section commune aux deux parties. (Voir Considérations Techniques).

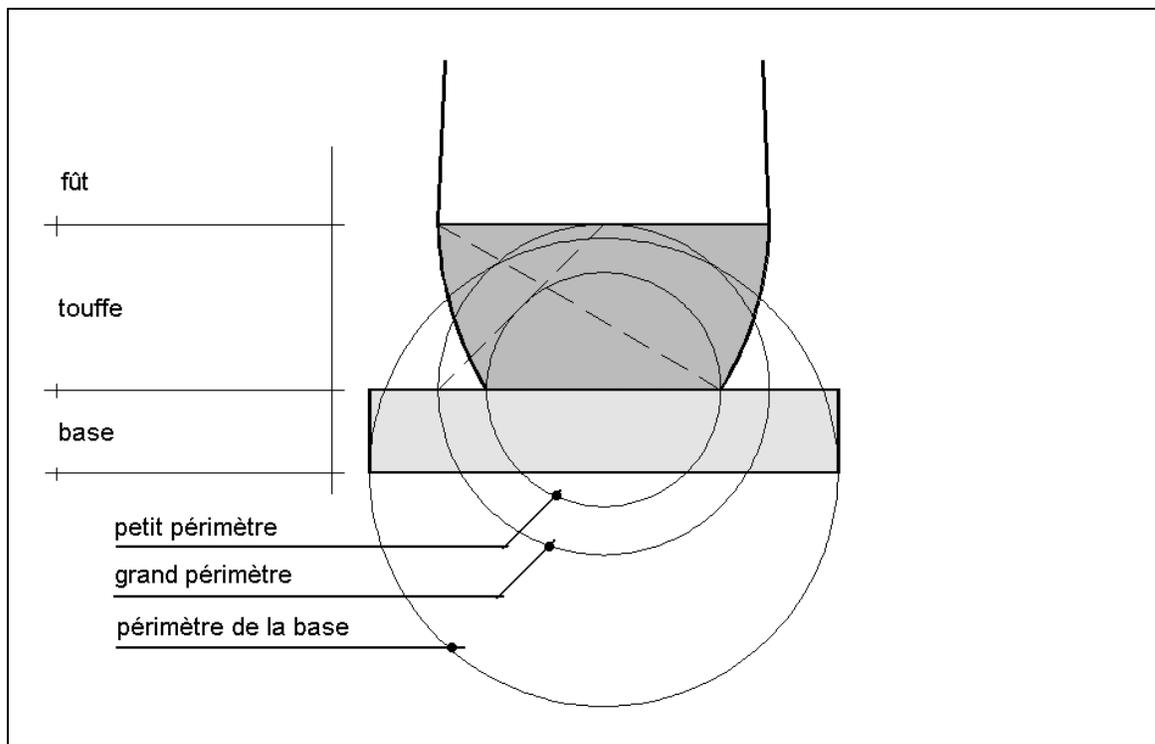
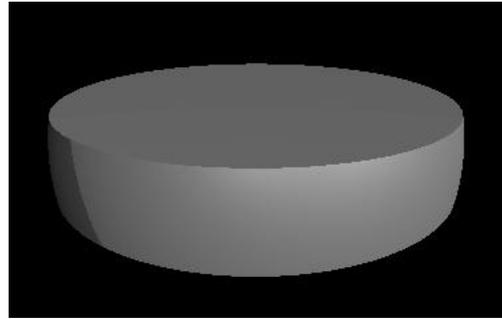


Fig. 5. Tracé de la base et de la touffe

5.3. La base :

La base (base.scm): elle est généralement cylindrique et de hauteur variable (voir Fig. 5). Certains exemples montrent des versions avec les bords supérieurs arrondis, d'autres montrent des côtés embossés. Le diamètre et la hauteur de la base varient sans un rapport défini, selon le site, le style et la fonction de l'édifice architectural, mais en général sa hauteur est égale à la moitié du grand rayon de la colonne.



Son diamètre est égal au grand_diametre * $\sqrt{2}$. Cet élément est utilisé par les architectes et/ou les maître-tailleurs pour corriger les incidences des irrégularités du terrain sur la colonne (Gryse 1984).

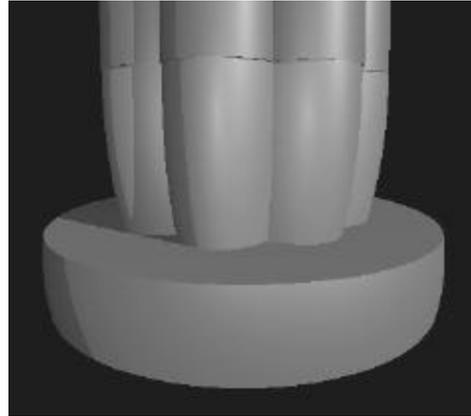
Sa représentation numérique est faite par une multiplication d'un plan horizontal infini avec une sphère.

```
(define plan
(lambda ()
(DLatt
(SDcolRGB (vector 255 255 255 255))
(SDmatrep (SGmat001 (vector (/ pi 2) 1)))
(DLformz2
(vector
(vector (-(/ hauteur_base 2)) 0 0 1) ;p0
(vector (/ hauteur_base 2) 0 0 1) ;p1
(vector 0 (/ hauteur_base 2) 0 0) ;p2
(vector 0 0 (/ hauteur_base 2) 0) ;p3
(vector 0 (/ hauteur_base 2) 0 0) ;p4
(vector 0 0 (/ hauteur_base 2) 0) ;p5 ))))

(define base
(lambda ()
(DLatt
(SDcolRGB (vector 255 255 255 255))
(SDmatrep (SGmatr1 (vector 0 (-(/ hauteur_col 2) (/ hauteur_base 2)) 0 1)))
(DLmul
(GDsph000 (vector rayon_base 1))
(plan))))))
```

6. Considérations techniques :

Une des difficultés auxquelles j'ai fait face fut la modélisation de la touffe d'une manière qui aurait permis d'avoir un raccordement exact avec le corps du fût. Ce problème découle de la particularité des deux éléments concernés : le module de base de la touffe est un ellipsoïde tandis que celui du fût est une conique avec une section circulaire. La difficulté réside que dans la version actuelle de SGDLsoft la manipulation de la quadrique ne permet pas un contrôle assez flexible pour que ce raccordement soit exact.



J'ai réussi à avoir un agencement exact sur le point d'intersection qui correspond au quadrant de la section circulaire de la cannelure et au sommet de la section elliptique de la corne.

8. Conclusion:

Pendant ces 4 semaines, ce fut une expérience enrichissante en termes de développement de ma perception personnelle de la CAO et de ses concepts, de l'exploration de nouvelles venues et problématiques de ce domaine, et de l'acquisition de nouvelles compétences en programmation. Cette période m'a permis, en particulier, d'explorer en une première approche, les possibilités de modélisation par Scheme et SGDLsoft.

La représentation des connaissances et du savoir-faire architectural est influencée par les outils en usage. Le degré de pertinence de cette représentation et de son exactitude obéissent à des principes qui doivent être inclus dans la modélisation de l'objet architectural. Les fondements géométriques de SGDLsoft (surtout ses propriétés affines et projectives) confèrent une plus grande maniabilité à la modélisation et établissent des niveaux d'intervention (pour l'utilisateur) qui permettent d'explorer plusieurs solutions, et par conséquent, d'ajuster ces dernières aux spécifications d'un problème réel.



J'ai apprécié la puissance de Scheme et sa flexibilité. Sa grammaire compacte et sa syntaxe facile m'ont permis d'embarquer immédiatement dans la programmation. En plus, Scheme a suscité mon intérêt pour approfondir davantage mes compétences en programmation par cet outil, et comme dans tout apprentissage, le facteur temps est important pour l'assimilation du fonctionnement de SGDLsoft, surtout parce que ses propriétés géométriques sortent du cadre usuel de la géométrie métrique. La disponibilité de ce logiciel pour l'apprentissage académique rendrait sa maîtrise et son développement plus adéquats, et lui assurerait une visibilité et une place propre dans le marché des outils de la CAO.

La présence plus fréquente d'un tuteur en programmation aurait été plus bénéfique pour faciliter mon interaction avec les interfaces et aurait accéléré mon apprentissage. La disponibilité d'un seul ouvrage de référence, le manuel de l'utilisateur de SGDLsoft, constitue une introduction élémentaire à ce logiciel qui serait plus avantageuse avec un didacticiel, ainsi qu'avec des exemples réalisés par d'autres étudiants.

Références.

Benoît, François (1911). *L'architecture, Antiquité*. Paris; Librairie Renouard.

Cenival, Jean-Louis De., (1964). *Égypte: Époque Pharaonique*. Fribourg; Office du Livre.

Choisy, Auguste (1983). *Histoire de l'Architecture, tome 1*. Genève; Slatkine Reprints.

Fletcher, Banister (1961). *A History of Architecture on the Comparative Method*. New York; Charles Scribner's Sons.

Gryse, Bob De., (1984). *Karnak: 3000 ans de gloire*. Liège; Éditions du Perron.

Lloyd, Seton Howard Frederick (1986). *Ancient Architecture*. New York; Electa/Rizzoli.

Plommer, Hugh (1956). *Ancient and Classical Architecture*. London; Longmans, Green.

Sturgis, Russell (c1906-15). *A History of Architecture*. New York; The Baker and Taylor C Company.

Wildung, Dietrich (1997). *Egypt: from Prehistory to the Romans*. Koln/London; Taschen.