



**HAL**  
open science

# Un logiciel de stockage, de traitement et de visualisation graphique et cartographique des données géologiques et géotechniques

Mohamed Damak

## ► To cite this version:

Mohamed Damak. Un logiciel de stockage, de traitement et de visualisation graphique et cartographique des données géologiques et géotechniques. Géologie appliquée. Université Joseph-Fourier - Grenoble I, 1990. Français. NNT : . tel-00785637

**HAL Id: tel-00785637**

**<https://theses.hal.science/tel-00785637>**

Submitted on 6 Feb 2013

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

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



# THESE

présentée par

**DAMAK Mohamed**

Pour obtenir le titre de DOCTEUR de

**L'UNIVERSITE JOSEPH FOURIER - GRENOBLE I**

Spécialité : Mécanique

Univ. J. Fourier - O.S.U.G.  
MAISON DES GEOSCIENCES  
DOCUMENTATION  
B.P. 53

F. 38041 GRENOBLE CEDEX

Tél. 04 76 63 54 27 - Fax 04 76 51 40 58

Mail : pjalour@ujf-grenoble.fr

## Un logiciel de stockage, de traitement et de visualisation graphique et cartographique des données géologiques et géotechniques

17 SEP. 2003

Date de soutenance : 29 juin 1990

Composition du jury :

M.	R. NEGRE	Président
M.	G. BALLIVY	Rapporteurs
Mme	J. TROCCAZ	
M.	M. DAYRE	Directeur de Thèse
M.	H. BEN DHIA	Examineur

Thèse préparée au sein de l'Institut de Recherches  
Interdisciplinaires de Géologie et de Mécanique

# THESE

présentée par

**DAMAK Mohamed**

Pour obtenir le titre de DOCTEUR de

**l'UNIVERSITE JOSEPH FOURIER - GRENOBLE I**

Spécialité : Mécanique

---

## **Un logiciel de stockage, de traitement et de visualisation graphique et cartographique des données géologiques et géotechniques**

---

Date de soutenance : 29 juin 1990

Composition du jury :

M.	R. NEGRE	Président
M.	G. BALLIVY	Rapporteurs
Mme	J. TROCCAZ	
M.	M. DAYRE	Directeur de Thèse
M.	H. BEN DHIA	Examineur

Thèse préparée au sein de l'Institut de Recherches  
Interdisciplinaires de Géologie et de Mécanique

## Avant Propos

Les travaux présentés dans ce mémoire ont été réalisés à l'Institut de Recherche Interdisciplinaire de Géologie et de Mécanique (I.R.I.G.M) de l'Université Joseph Fourier avec la collaboration de L'Ecole Nationale des Ingénieurs de Sfax (E.N.I.S, Tunisie) et de l'Université de Sherbrooke (Quebec).

L'Institut de Recherche Scientifique et Technique de Tunis a financé la première année d'étude . J'exprime pour cela mes sincères remerciements à Monsieur Foued ZARGOUNI . Les autres années d'étude ont été financées dans le cadre d'une coopération franco-tunisienne .

Je tiens à remercier les enseignants de l'E.N.I.S qui ont contribué à ma formation de géologie ainsi que les enseignants de l'Institut de Mathématiques Appliquées de Grenoble qui m'ont permis d'acquérir une formation en informatique .

Monsieur le professeur R. NEIGRE, professeur à l'université Joseph Fourier et directeur de l'I.R.I.G.M m'a fait l'honneur d'accepter la présidence de du jury, qu'il en soit très sincèrement remercié .

A Monsieur le professeur G. BALLIVY professeur à l'université de Sherbrooke, qui m'a invité au sein de son équipe dans le cadre d'un stage et qui a accepté d'être mon rapporteur, j'exprime mes sincères remerciements .

A Madame Jocelyne TROCAZ chercheur au laboratoire d'Informatique Fondamentale et d'Intelligence Artificielle qui m'a prodigué de précieux conseils et qui a accepté d'être mon rapporteur et à Monsieur H. Ben Dhia, professeur à l'E.N.I.S, j'exprime tous mes remerciements .

A monsieur M. DAYRE, maitre de conférence à l'Université Joseph Fourier qui m'a accueilli et a dirigé ces travaux au sein de l'équipe de mécanique des roches, j'exprime toute ma reconnaissance et mes sincères remerciements .

## Abstract

In this thesis, a software specification, conception and partial realisation are presented . This software allows geological and geotechnical data storage, treatment, mapping and tridimensional graphic display .

As much importance is given the analysis and conception method as to the system realisation .

We have used The System Analysis and Design Technic (S.A.D.T) to specify the software . This method allows us to constitute the function analysis file . Furthermore the entity -association model was applied . This lead us to define a set of entity corresponding to geological data .

The object oriented design allowed the system to be more portable and adaptable .

Priority was given to the sub-software allowing geometric modelisation . Tridimensionnal display, triangulation and hidden surface we studied . Chosen algorithm were modified to suit the application .

## I. INTRODUCTION

Il est courant que l'on mène une campagne de reconnaissance géologique et géotechnique d'un terrain qui a déjà été étudié et dont les informations correspondantes ont été perdues . D'autre part les méthodes manuelles classiques de traitement de données notamment celles à buts cartographiques sont complexes à exécuter ; Cela implique la nécessité de développer un système informatique qui ,sans remplacer le géologue , permette le stockage , l'interrogation sélective , le traitement et la visualisation graphique et cartographique des données géologiques et géotechniques .Ce système doit être interactif,évolutif et facilement maintenable.

Depuis une vingtaine d'années ,plusieurs expériences ont été réalisées et plusieurs étapes ont été parcourues . Aujourd'hui , les systèmes les plus répandus sont les systèmes de banques de données (FIDGI: Laboratoire Régionale des Ponts et Chaussées de Rouen, MIISFIIT:L.R.P.C Aix en Provence, SOCRATE :Laboratoire Centrale de Ponts et Chaussées ) . Une banque de données est formée par une base de données et par un système de gestion de base de données (S.G.B.D) . Ces systèmes ont vite évolué vers les systèmes relationnels ce qui a permis d'améliorer la gestion , l'interrogation et le traitement des données . Cependant plusieurs systèmes,souvent non connectés à des banques de données, permettent désormais d'importants traitements numériques et géostatistiques ainsi que des réalisations cartographiques.

La majorité de ces applications est caractérisée par les faits suivants :

- \* Ces systèmes ont souvent été développés sans l'application de méthodes précises de spécification et de conception , ce qui a donné naissance dans la plus part des cas à des logiciels peu modulaires , peu adaptables ,peu évolutifs et très difficilement maintenables .

- \* La majorité de ces systèmes a été développée sur de gros ordinateurs et a un coût très élevé , ce qui limite la diffusions de ces logiciels.

- \* Le plus souvent, les processus d'analyse et de synthèse qui conduisent à la création d'une carte sont basés sur des

traitements numériques (ce qui s'accompagne souvent par une distorsion de l'information géologique) et ne donnent pas au géologue-utilisateur la possibilité de rectifier ou compléter les résultats graphiques (Durand 88). L'utilisation des nouvelles techniques infographiques pourrait améliorer l'interaction avec l'utilisateur.

Les micro-ordinateurs se révèlent de plus en plus intéressants étant donné l'amélioration continue de leurs performances et la baisse de leur prix. Mais la non-application de méthodes précises de développement logiciel a conduit à un phénomène de prolifération de petits programmes "personnels", incompatibles entre eux, non adaptables et non évolutifs.

Ces constatations nous ont poussé à développer tout particulièrement les deux points suivants:

-La méthodologie du développement du logiciel ; c'est à dire la nécessité d'utiliser des méthodes d'analyse et de conception précises afin de produire un logiciel qui soit l'image informatique du monde géologique à modéliser et qui présente les qualités souhaitées de lisibilité, modularité, évolution et de maintenance.

-L'utilisation des techniques de l'infographie : Ces techniques permettront la visualisation des données en deux et en trois dimensions et assureront une meilleure interaction avec l'utilisateur.

Ainsi nous avons progressé dans notre travail selon les étapes suivantes :

\* Nous avons d'abord effectué la spécification et la conception d'un système général. Ce système permet le stockage, le traitement et la visualisation graphique et cartographique des données géologiques et géotechniques. La spécification et la conception ont été basées sur les méthodes suivantes : le "Structured Analysis Design Dechnic" (S.A.D.T), le modèle "entité-association" et la conception "orientée objet".

\*La deuxième étape a consisté en la réalisation modulaire du sous logiciel graphique (une partie du logiciel général) permettant la modélisation et la visualisation tridimensionnelle des données d'un site, avec élimination des parties cachées.

Nous n'avons pas réalisé ce travail dans le but de fournir un système qui concurrence les systèmes existants. Notre objectif était d'abord de mettre au point une méthodologie de développement correspondant aux problèmes et aux données géologiques. Nous avons démontré aux géologues et géotechniciens intéressés par le développement logiciel l'utilité des méthodes (notamment la méthode orientée objet) en les appliquant à notre cas précis : un logiciel qui répond aux besoins de stockage, de traitement et de visualisation des données.

La conception orientée objet fait correspondre à chacune des entités du monde réel, un objet logiciel qui le modélise dans la machine. Cette conception a montré ces dernières années, sa capacité à produire, dans plusieurs domaines scientifiques, des logiciels modulaires évolutifs, maintenables et qui représente l'image fidèle du monde réel à modéliser.

Jusqu'ici, peu de logiciels dans le domaine géologique, ont été conçus selon la méthode orientée objet.

Cependant, la priorité de réalisation a été donnée au sous-logiciel graphique présenté sous forme d'une bibliothèque de modules de modélisation et de visualisation. Dans ce cadre nous avons été amené à faire un certain nombre de choix concernant les algorithmes et les structures de données pour la modélisation géométrique des objets, l'élimination des surfaces cachées, la triangulation et la cartographie.

## II. LE CONTEXTE HISTORIQUE

En géologie, en géotechnique et en cartographie, l'utilisation de l'informatique est devenue une solution courante pour réaliser divers applications, notamment de stockage et de traitement numérique et cartographique des données.

Historiquement plusieurs étapes ont été parcourues. Chacune de ces étapes est généralement conditionnée par le niveau des moyens matériels et logiciels dont on dispose. On peut distinguer les étapes suivantes :

### *\* L'ETAPE PREINFORMATIQUE : MICROFILMAGE (1970):*

Il s'agit d'un simple stockage de données par les techniques de microfilmage. Cela permettait de disposer, sur une zone déterminée d'un stock d'informations qui peuvent être consultées assez rapidement. Ces informations ont pour base les dossiers d'étude issus d'une campagne de reconnaissance. Les techniques de microfilmage ne répondaient pas à un des objectifs initiaux à savoir le traitement automatique des données.

### *\* L'ETAPE DU FICHER INFORMATIQUE :*

C'est le premier niveau qui répond au double objectif de stockage et de traitement de données. Mais "le fichier informatique" reste une appellation vague qui correspond à une grande variété de réalisations. "Ce sont néanmoins toujours des collections d'enregistrements de même modèle qu'on distinguera selon les critères suivants:

- La formalisation de l'information,
- l'organisation du stockage,
- les possibilités d'interrogation,
- La spécificité ou au contraire la généralité du

logiciel" (Buisson 76).

On peut classer les systèmes les plus connus par ordre de perfectionnement comme suit:

+Les systèmes dérivés des chaînes de dessins automatiques de feuilles de sondage (par exemple celui réalisé au Laboratoire Régional des Ponts et Chaussées de Bordeaux : "COUP") (Sanejouard 75).

Au laboratoire d'Autun un système "GEOLE" a été mis au point. "Ces systèmes sont caractérisés par une absence de formalisation des données et un stockage 'en vrac' ménageant des possibilités restreintes d'interrogation" (Sanejouard 75).

+Les systèmes orientés vers des programmes de traitement définis à l'avance et pour lesquels on ne stockera que l'information sélectionnée et formalisée correspondante : Par exemple, la chaîne de programmes "JACARD" du service géologique régional Jura-Alpes. JACARD a été utilisé par la suite par les L.R.P.C d'Autun et de Lyon dans le cadre d'études de ressources alluvionnaires (GUILLEMIN 75).

-+Les systèmes assurant au moyen d'un logiciel spécifique, le stockage, la gestion et l'interrogation -généralement sur critères limités - de l'ensemble de l'information structurée et formalisée : Exemple le système "FIDGI" (Buisson 76),. Ce système sera détaillé par la suite.

-+Les systèmes comparables à "FIDGI" mais utilisant une chaîne de programmes non spécifiques de l'application géotechnique qui procurera des possibilités d'interrogation beaucoup plus importantes.

### *\* L'ETAPE DE BANQUE DE DONNEES :*

Dans l'étape précédente, le stockage s'effectuait dans des fichiers de formes et de structures différentes contenant des informations redondantes et traitées par des programmes différents. Dans les systèmes de banques de données, la (ou les) base(s) de données contiennent des enregistrements de modèles ou de types différents reliés entre eux selon des modalités variées.

La notion de fichier correspond à la nécessité de gérer des données relative à un domaine d'activité restreint, alors que la notion de base de données découle de l'idée de partage de ressources qui s'impose dès lors qu'on admet la possibilité d'utiliser des données en dehors de leur domaine de création.

La base de données est donc un ensemble cohérent de ressources communes mises à la disposition de nombreux utilisateurs au sein d'une même organisation pour des usages multiples et différents en nature .

Une banque de données est formée par une base de données et d'un Système de Gestion de Base de Données (S.G.B.D) . Le S.G.B.D permet la définition , la gestion et l'exploitation de la base de données et il est totalement transparent pour l'utilisateur .

Ces systèmes représentent les avantages suivants:

- + La suppression de la redondance comme conséquence de l'établissement d'inter-relation entre données,
- + l'indépendances entre l'enregistrement des données et leurs utilisation ce qui permet une certaine sécurité,
- + l'amélioration des performances lors des accès

Un des systèmes de banques de données géologiques et géotechniques les plus connus est "MIISFIT" réalisé au L.R.P.C d' Aix-en-Provence (Guinnet 76).

Les systèmes de banques de données ont vite évolué par la suite vers l'utilisation des S.G.B.D relationnels, ce qui a permis d'améliorer la gestion, l'interrogation et le traitement des données .

Les systèmes de gestion de base de données cités ci-dessus permettent essentiellement le stockage et l'interrogation sélective des données géologiques et géotechniques récupérées à partir de forages, de données superficielles (topographie, lithologie de surface ...) et de données des essais sur échantillons ( aux laboratoires ou in situ ). Les applications cartographiques liées à ces bases de données sont très rares. Ces systèmes ont caractérisés les années soixante dix.

#### \* LA CARTOGRAPHIE AUTOMATIQUE

Cependant, et durant la même période , certaines applications cartographiques ont été développées . Ce sont des systèmes de cartographie automatiques . ces systèmes avaient pour but de produire par l'intermédiaire de l'ordinateur, des cartes que le géologue produisait manuellement .

La cartographie automatique a été définie (Visvalingum 90) comme "l'utilisation ,pour de buts de productivités, de l'ordinateur afin de produire des cartes, cela inclue le remplacement de plusieurs activités manuelles telles que la définition et les changements d'échelles , l'édition, la séparation des couleurs , la symbolisation et la définition des légendes , par des processus informatiques.

Dans ces applications ,l'informatisation des processus d'analyse et de synthèses qui conduisent à la création d'une carte géologique se rapportent essentiellement à des traitements de données numériques . Il est donc nécessaire avant tout traitement de transformer les données de type géologiques en données de types numériques. Le géologue doit donc codifier ces données avant de les rentrer dans la base de données.

On ne faisant passer dans le processus que les paramètres qui ont pu être codifiés sur les fiches de terrains ,puis entrés dans une base de données et finalement traités par le programmes ; on laisse de coté les éléments qui sont plus spécifiques à la géologie et qui ne rentrent pas dans les capacités de traitement des programmes.Cela se traduit nécessairement par une perte et une distorsion de l'information.

En effet ,l'analyse en référence spatiale sur une carte fait souvent appel à l'intuition et l'expérience du géologue. La saisie intuitive de relations spatiales entre les informations portées sur une cartes constitue un des éléments les plus importants dans l'élaboration de la carte géologique.

D'autre part ces systèmes sont peu interactives et ne donne aucune possibilité au géologue d'introduire ces concepts pour guider ou corriger le traitement .

Pour ces raisons " la création des cartes géologique font encore très peu appel à ces techniques "(Durand 89). Et les géologues deviennent de plus en plus orientés vers les systèmes interactives qui font remplacer la cartographie automatique par la cartographie assistée par ordinateur .

#### \*LA CARTOGRAPHIE ASSISTEE PAR ORDINATEUR .

L'objectif de la cartographie assistée par ordinateur est fondamentalement différent de celui de la cartographie automatique. En effet cette nouvelle approche ne demande à l'informatique qu'une

assistance qui exclue l'interprétation automatique . Cette approche remplace le coté automatique par un coté interactif . cette interactivité a fait le succès de ces systèmes car elle permet à l'utilisateur géologue de récupérer son rôle essentiel dans le processus .

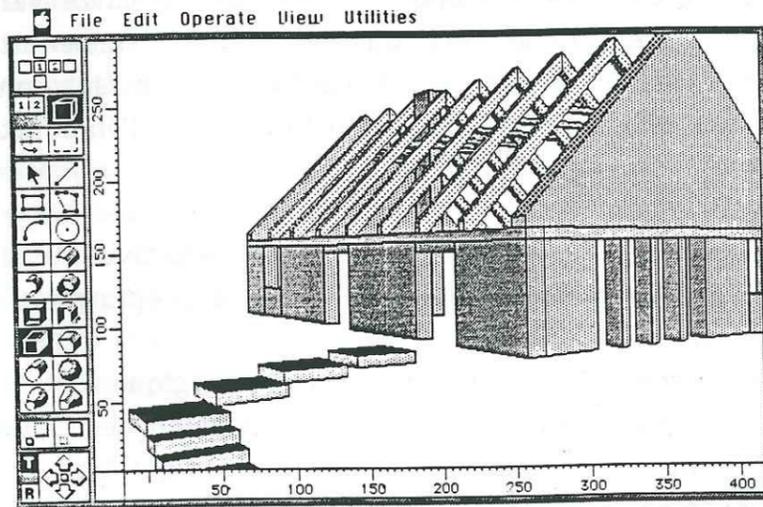
Le développement de ces systèmes est du à l'évolution spectaculaire que les techniques de l'infographie et de la conception assistée par ordinateurs ont connu.

On peut classer ces applications en deux classes :

- les logiciels de dessin assisté par ordinateur
- les logiciels de conception assistée par ordinateur.

#### *\*LE DESSIN ASSISTE PAR ORDINATEUR*

Les logiciels de dessin assisté par ordinateur(D.A.O) fournissent interactivement par l'intermédiaire des menus un ensemble d'icônes ,de symboles de graphiques(droites,cercles ,ellipses...) et de trames que l'utilisateur pourrait utiliser(par assemblage et diverses manipulations) pour former son dessin . le graphique suivant illustre un ensemble de symboles et de trames que le menu d'un logiciel de D.A.O peut présenter.



Edition Graphique sur Micro-ordinateur  
(Schweizer 87)

Ces menus interactifs permettent de dessiner plus rapidement, et corriger facilement les erreurs au lieu de devoir recommencer le schéma depuis le début.

Ces systèmes offrent les mêmes avantages que le traitement de texte : facilité de correction, précision et qualité d'impression . Un dessin produit peut être stocké et peut être sujet de correction et de manipulation dans une nouvelle session de travail.

Ces logiciels peuvent être utilisés pour la cartographie puisque ils fournissent les outils de dessins nécessaires pour produire une carte. de stocker ce produit , de le corriger et de le manipuler (changement d'échelle ,...).

Nous présentons ici un Un exemple d'utilisation de ce type de logiciel .Il s'agit d'un système de Cartographie géologique assistée par ordinateur (C.G.A.O). Cet exemple a été effectué au département des sciences de la terre de l'université de Montréal (Durant 89).

Dans cet exemple un micro-ordinateur a été utilisé , ce micro est équipé d'un écran graphique matriciel "bit map display" .

Les étapes de production du document CGAO sont les suivantes:

#### *a) création des fiches et du fond de la cartes*

La carte comporte normalement en arrière-plan un fond de carte topographique , qui peut être créer en numérisant un document existant ou en dessinant les contours principaux à l'aide de la souris du micro-ordinateur ou avec une table de numérisation. Le dessin principal d'affichage qui pour le géologue est la carte ,sert de mode d'accès privilégié dans la base de données .

#### *b) mise en place des objets graphiques*

Une fois le fond de la carte est créé ,on indique la position des objets géologiques qu'on veut dessiner et stocker.

Fichier Éditer Types Accès Texte Symboles Format Trames

Fiche de données de Struc.Géol.: (37 fiches)

ep. (en m): - Direction (0-360°) 65° Pendage 0°

Commentaires	Illustration (Photo ou schéma)
Deux anticlinaux et un synclinal dans un épaisseur de 8 m de strates Trenton disloquées par la glaciotectonique.	
Ref.	
Dossier Métro Pie IX à l'Assomption	
Diapos(D):	
BOY-D1 à D11	
photo(P)	
20	
Ech(E)	
	Stratigraphie
	Argile grise massive sur till argileux sur calcaire de la ftn de Tétreauxville en lits de 10 à 15 cm, disloqués par

Vue du plissement anticlinal au mur Ouest de la tranchée au chaînage

Préc. Suiv. Revenir Dessin Lien

Exemple de fiche d'objet; la fiche de chaque objet contient toutes les données qui se rapportent à un élément du dessin de la carte. (Durand 89)

Un objet graphique peut être un forage, un profil géologique, des mesures d'inclinaison de couches, etc. On remplit par la suite une fiche par objets dans le quel on dessine l'objet, on met des commentaires, Le graphique suivant présente une fiche d'objet (graphique et texte).

Une position X,Y à l'écran peut recevoir plusieurs objets superposés (ex données topographique, hydrogéologique, géologique, etc). Cela tient au fait que le système fonctionne comme une base de donnée à référence spatiale.

#### \*LA CONCEPTION ASSISTEE PAR ORDINATEUR

Le développement et le raffinement des techniques de dessin assisté par ordinateur ont conduit naturellement à informatiser toutes les phases de la conception dans de nombreux domaines. Ce qui est important à signaler ici, c'est la différence entre le D.A.O et la conception assisté par ordinateur (C.A.O). Alors que les logiciels de

D.A.O ne fournissent que des outils de production, de modification et de stockage de graphiques, les logiciels de C.A.O permettent -en plus des utilitaires graphiques- la modélisation et la gestion des structures de données à l'intérieur des programmes.

"En résumé la CAO, c'est la création et la modification d'une structure de donnée, représentant divers modèles de l'objet en conception par interaction graphique, et l'usage d'un ensemble de facilités et d'automatisme" (Schweizer 87)

Pour les applications cartographiques l'utilisation des systèmes de CAO sur micro-ordinateur est devenue très courante à la fin des années quatre vingt (Shepherd 90)

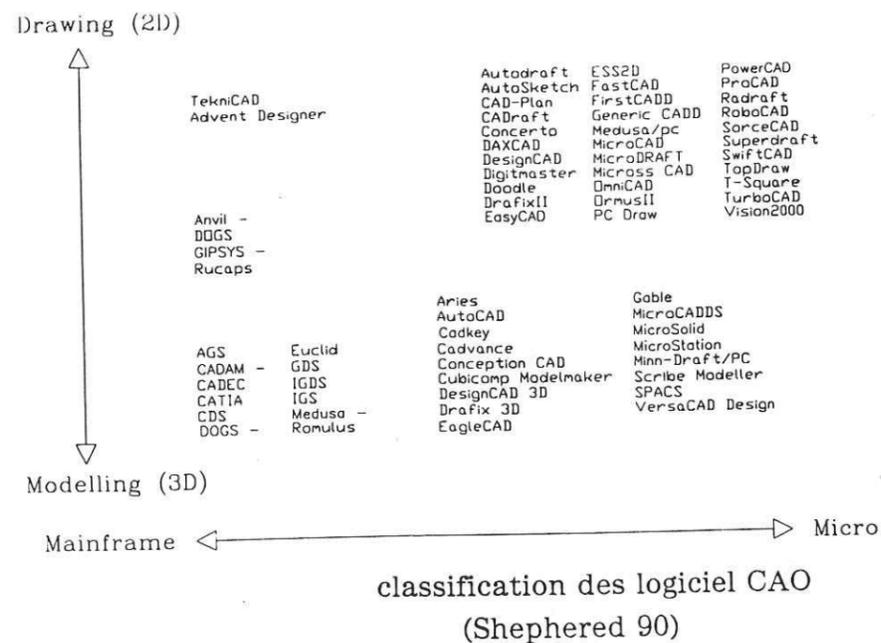
On peut distinguer deux sortes de logiciel de CAO

-2D-CAO : ce sont les logiciels qui travaillent en deux dimensions et fournissent essentiellement des outils de dessin. On peut les considérer plutôt comme des logiciels de D.A.O

-3D-CAO : ces logiciels concernent le dessin et la modélisation des objets tri-dimensionnels. Un objet tri-dimensionnel peut être modéliser en terme de modèle fil de fer, un modèle de volume défini par ses faces ou modèle solide.

Entre les deux extrêmes 2D et 3D-CAO ce sont les systèmes appelés les deux et demi-dimensionnels (2.5D)

La figure suivante illustre une classification des logiciels de CAO existant. L'axe horizontal représente la capacité de l'ordinateur sur lequel ils fonctionnent, alors que l'axe vertical représente leurs capacités de modélisation.



Le logiciel de CAO le plus utilisé dans la cartographie est le AUTOCAD (Hobbs 90). Ce logiciel offre les fonctionnalités suivantes :

-Acquisition de données: en premier stade les données sont acquises essentiellement par digitalisation en utilisant une tablette appropriée, cela concerne plutôt le fond de la carte.

Les éléments géométriques utilisés pour fabriquer la carte sont manipulés par des commandes de dessin, cela concerne les points, les droites, les cercles, les arcs, les ellipses, etc.

-L'édition : AUTOCAD offre plusieurs commandes d'édition interactives telles que l'agrandissement, la réduction aussi bien que des commandes de correction, de destruction, de déplacement d'éléments.

-La conception : il est possible d'essayer plusieurs combinaisons de couleurs, de formes de ligne, de symboles et de textes pour concevoir la carte finale.

-Les sorties graphiques : l'état final de la qualité graphique de la carte dépend en général du périphérique utilisé.

L'utilisation des logiciels de CAO dans la cartographie présente les avantages suivants :

- \*les possibilités de l'interactions graphiques.
- \*la précision du dessin

\*la qualité graphique du produit final est généralement bonne

\*Le coût des logiciels n'est pas excessif.

\*beaucoup de logiciels fonctionnent sur micro-ordinateur.

Les inconvénients:

- la digitalisation : les données manipulées proviennent de la digitalisation. Cela se traduit par la limitation de ne traiter que des données de surfaces telles que les données topographiques ou au mieux la lithologie de surfaces. Ces systèmes n'acceptent pas des données géologiques de profondeurs telles que les forages.

- La limitation de la modélisation : la modélisation des objets tridimensionnels n'est possible que dans des cas simples comme en électronique ou en mécanique où l'utilisateur définit interactivement les éléments de l'objet. Par contre il est impossible pour ces logiciels de modéliser un volume géologique (bloc-diagramme) à partir des données de forages.

En résumé, on peut considérer ces systèmes sont plutôt des outils généraux de dessin assisté par ordinateur. En général, ces systèmes ne sont pas connectés sur des bases de données. La visualisation cartographique y est assurée à partir des données numériques (digitalisation) et ne prend pas en compte des données géologiques (données de forages, natures lithologiques, etc). C'est ce qui rend ses systèmes plutôt utiles pour les traitements des données topographiques et la visualisation des modèles numériques. Par conséquent, ces systèmes ne peuvent pas être vus comme des systèmes généraux qui répondent aux besoins de stockage, de traitement, et de visualisation graphiques et cartographiques des données géologiques et géotechniques d'un site.

#### \* L'ORIENTATION DE NOTRE TRAVAIL

D'après l'aperçu général que nous venons de présenter, on peut dresser les constatations suivantes :

-Les systèmes existants peuvent être globalement classés les catégories suivantes:

-Les systèmes orientés banques de données géologiques et géotechniques, ces systèmes ne permettent que le stockage et l'interrogation sélective des données

-les systèmes de cartographie géologique automatique, ces systèmes n'ont pas connu beaucoup de succès, essentiellement à cause de l'absence d'interactivité.

- les systèmes de dessin et de conception assistés par ordinateur, ces systèmes sont basés sur l'interactivité mais ne traitent que les données numériques (à partir de la digitalisation) et n'acceptent pas des données de type géologiques.

Ceci peut être résumé par la figure 2-2 :

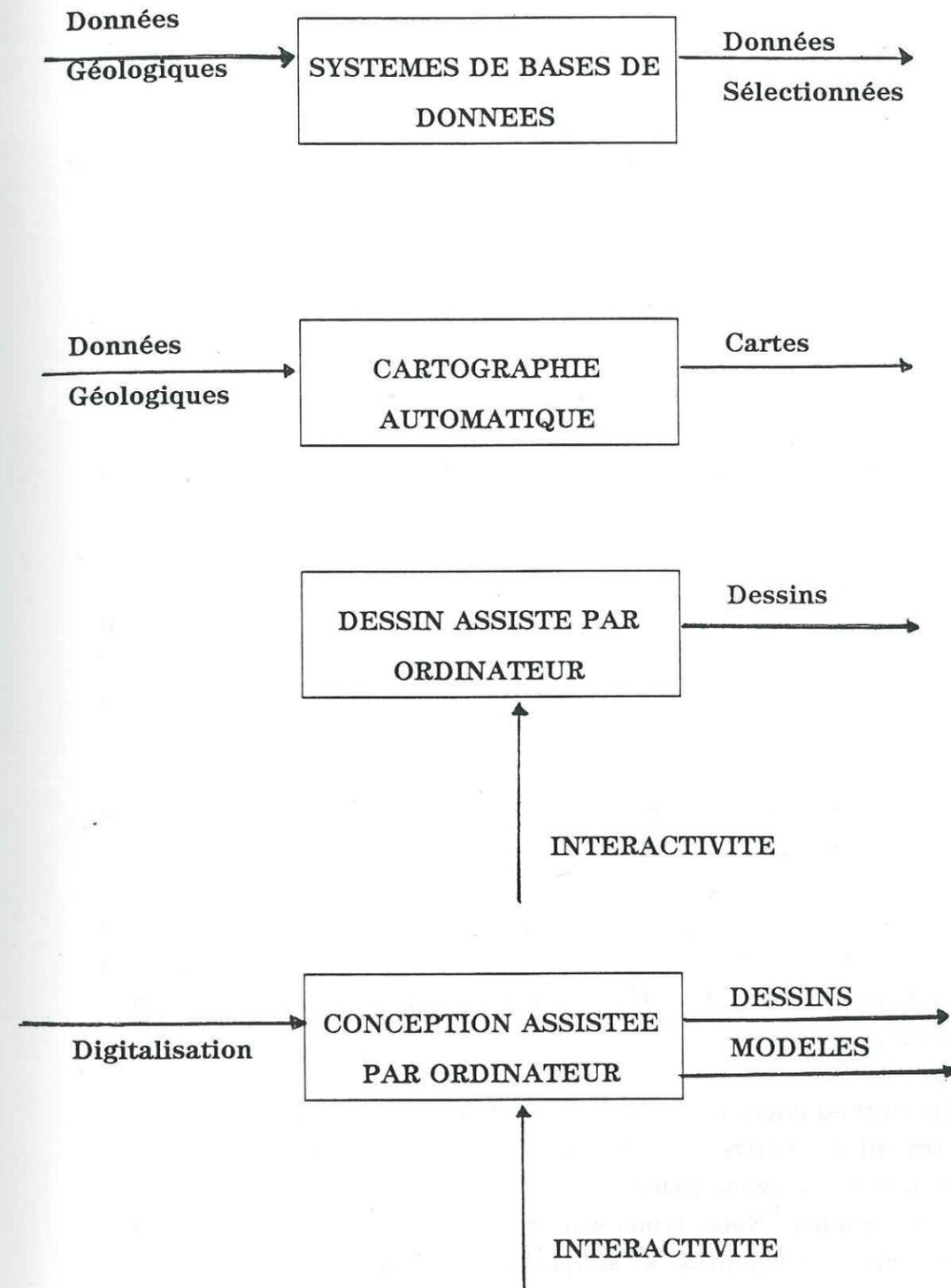


Figure 2-2

Il existe donc peu de systèmes généraux qui répondent à la fois aux besoins de stockage, de traitement et de visualisation graphique et cartographique des données géologique et géotechniques.

-La majorité des applications existantes ont été développées sans application de méthodes précises de développement logiciel . C'est ce qui rend la plupart des systèmes peu modulaires, peu adaptables et par conséquent difficilement maintenable . La difficulté de la maintenance n'est pas spécifique aux logiciels de géologie ,mais c'est un phénomène général considéré comme un symptôme de ce que certains spécialistes de génie logiciel appellent "la crise logicielle"(Booch 88).

-La plupart des applications ont été confrontées aux problèmes de la formalisation et de la modélisation des données géologiques et géotechniques . En effet l'information géologique est peu modélisable . Le traitement numérique des données géologiques qui conduit à des cartes ou des graphiques amène souvent à une distorsion de l'information (Durand 88) . Ceci a orienté la recherche vers des systèmes graphiques et non exclusivement numériques . L'amélioration de l'interactivité est essentielle pour permettre au géologue-utilisateur de corriger et de compléter, selon son expérience les résultats graphiques .

Les constatations précédentes ont orienté notre travail de façon à mettre l'accent tout particulièrement sur les points suivants :

1-La généralité du système : La spécification que nous avons effectuée concerne un système général qui répond à la fois aux besoins de stockage, de traitement et visualisation graphique et cartographique des données géologiques .

2-La méthodologie de développement logiciel : Nous avons essayé dans ce travail de mettre au point une méthodologie en appliquant des méthodes que nous avons jugées compatibles avec les problèmes et les données géologiques . Nous avons utilisé la conception orientée objet qui permet une meilleure évolution et adaptabilité au logiciel.

A notre connaissance il existe peu de systèmes géologique dont la conception est orientée objet.

3- L'interactivité : nous avons jugé utile de donner la priorité à la réalisation des fonctionnalités graphiques du système vu l'importance de l'interactivité et de la visualisation .C'est pour cette raison que nous

avons orienté notre travail de réalisation vers l'utilisation des techniques infographiques . De même nous avons donné beaucoup d'importance aux choix de structures de données lors de l'élaboration des algorithmes et de la modélisation géométrique .

### III. LA METHODOLOGIE DE DEVELOPPEMENT LOGICIEL.

#### A. INTRODUCTION

Depuis quelques années les spécialistes du logiciel ont commencé à parler d'une "crise logicielle". Les symptômes de cette crise ont été définis (Booch 88) comme suit:

- la non-correspondance entre les logiciels et les besoins de l'utilisateur ,
- la non-fiabilité des systèmes ,
- le prix excessif des logiciels: souvent le logiciel coûte plus que le matériel,
- la complexité des modifications et le coût élevé de la maintenance ,
- le temps de réalisation est souvent plus important que prévu
- les logiciels ne sont pas transportables d'un système à un autre .

Ces symptômes ne sont pas spécifiques aux logiciels dans le domaine des sciences de la terre , ils ont été définis d'une façon plus générale.

La cause de cette crise est vraisemblablement due à la difficulté de gérer simultanément les relations et les ramifications des situations complexes, cependant la solution n'est pas de réduire la complexité des problèmes mais d'améliorer les outils nécessaires à leur résolution.

Nous devons ainsi utiliser des méthodes de développement logiciel pour vaincre la complexité des problèmes(Bailly 87). Les méthodes de conception les plus connues sont :

- la méthode de conception hiérarchique ,
- la méthode de conception "flots de données",
- la méthode de décomposition fonctionnelle descendante,
- la méthode de conception "orientée objet"

Nous allons détailler cette dernière méthode parce que c'est celle a donné le plus de résultats [Bailly 87] ces dernières années .

#### B. LA METHODE ORIENTE OBJET

Ce que nous désirons , est une méthode permettant de construire un système logiciel qui soit l'image la plus fidèle possible du monde réel à modéliser, et qui présente de bonnes qualités d'adaptation et d'évolution .

La méthode la plus efficace consiste à faire correspondre à chacune des entités du monde réel un objet logiciel qui le modélise dans la machine . La structure du système logiciel est alors définie par la structure du monde physique ,et non pas par les tâches à exécuter: c'est la conception orientée objet

Cette méthode a été simultanément découverte par les spécialistes de la simulation, par les scientifiques de l'intelligence artificielle et par les psychologues dans leurs travaux de modélisation des systèmes physiques , artificiels ou humains. Elle est encore en pleine évolution,"Le style orienté objet évolue encore et présente des formes variées ... C'est une lame de fond qui déborde des applications usuelles de l'intelligence artificielle . Elle va modifier , à terme , les habitude de programmation de tous les secteurs informatiques" [Bailly 87].

Un objet est une entité logicielle qui modélise une entité physique(FIGURE 3-1) ; c'est donc un module élémentaire "encapsulant" un ensemble cohérent de données qui lui sont propres, et des procédures qui manipulent ces données (FIGURE 3-2). Un logiciel est alors une collection d'objets différents(Booch 88)

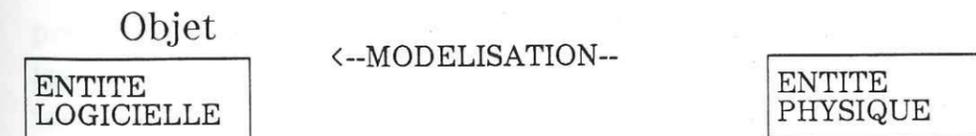
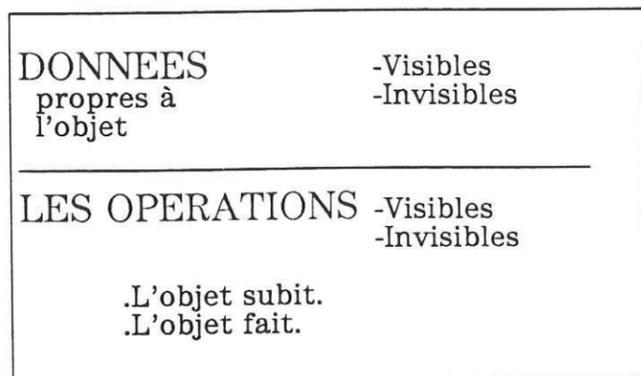


FIGURE 3-1



Savoir  
Declaratif  
(Le nom)

Savoir  
Procédural  
(Le verbe)

FIGURE 3-2

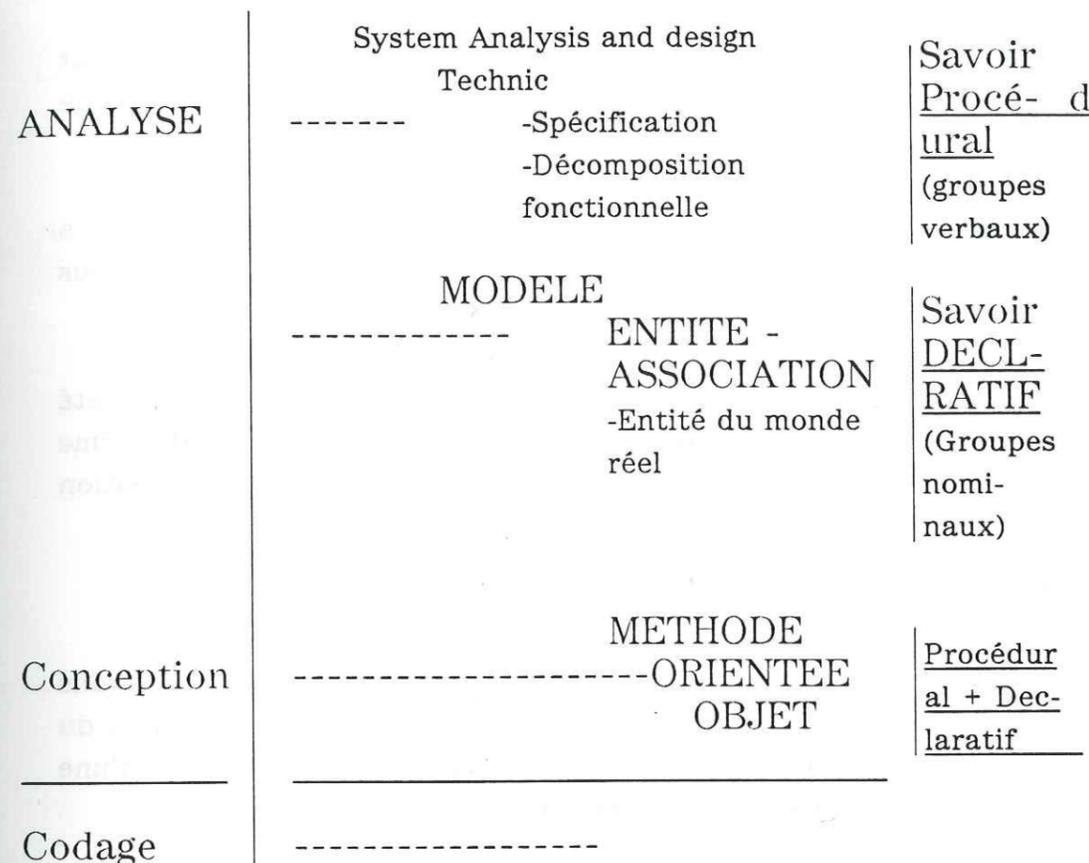
Les objets sont des boites noires utilisables dans des contextes divers . C'est ce qui rend le logiciel modulaire , lisible et évolutif . La modification à l'intérieur d'un objet n'affecte pas le reste des objets.

C. "LA METHODOLOGIE ORIENTEE OBJET"

La difficulté dans une application est en fait de déterminer quels sont les sortes d'objets à créer . Dans notre travail nous avons suivi une méthodologie qu'on appellera "méthodologie orientée objet" . Nous avons suggéré cette appellation car le but final est de définir les objets logiciels selon le concept orienté objet . Cette méthodologie présente les étapes suivantes ::

- 1- Définition du problème,
- 2- Définition des entités du monde à modéliser
- 3- Définition des objets logiciels (la conception orientée objet proprement dite)

Le cycle de vie du logiciel selon cette méthodologie est représenté par la Figure 3-3



CYCLE DE VIE DU DEVELOPPEMENT : METHODOLOGIE  
FIGURE 3-3

1. DEFINIR LE PROBLEME : spécification et analyse (s.a.d.t)

Nous avons commencé notre méthodologie en définissant le problème ;c'est la phase de spécification et d'analyse . Nous avons défini les besoins (cahier des charges) d'un système général de stockage , de traitement et de visualisation graphique et cartographique des données géologiques.Nous avons ensuite appliqué la méthode "System Analysis Data Technic" (S.A.D.T) . Cette méthode est basée sur la construction de diagrammes de fonctionnalités en suivant une syntaxe précise: pour chacune des fonctionnalités on définit les données(entrées) les

résultats(sorties) , le contrôle qui déclenche l'activité et le mécanisme selon lequel l'exécution a lieu .

Cette méthode permet de définir les fonctionnalités du système et de décomposer chacune de ces fonctionnalités en un ensemble de fonctionnalités plus élémentaires.

En plus de la décomposition fonctionnelle, cette méthode nous a permis de définir les données manipulées par le système, ce qui nous prépare déjà à la définition des entités et des objets.

La syntaxe, l'intérêt et la justification de cette méthode ont été détaillés dans le chapitre "analyse générale des fonctionnalités". De même nous avons présenté dans ce chapitre le résultat de la décomposition fonctionnelle .

## 2. LA DEFINITION DES ENTITES DU MONDE REEL

C'est la deuxième phase de notre méthodologie , ici nous avons appliqué le modèle entité-association . IL s'agit de définir les entités du monde réel qui correspondent aux informations collectées lors d'une campagne de reconnaissance d'un terrain.

Dans ce modèle, on définit les associations existantes entre les différentes entités ou les ensembles d'entités . Ces relations reflètent la structure physique du domaine à modéliser, elles peuvent être illustrées dans un diagramme appelé "schéma-Z généralisé" .

L'application de ce modèle nous permet de contribuer à la spécification formelle des données géologiques et géotechniques et nous prépare à la conception orientée objet.

## 3. LA CONCEPTION ORIENTEE OBJET

Il s'agit de définir les objets logiciels correspondant aux entités du monde réel, de définir leurs attributs ainsi que leurs opérations qui les manipulent et enfin l'interface de chacun des objets avec l'extérieur . L'interface d'un objet est sa partie accessible, elle est définie par les

attributs et les opérations visibles, tout le reste est propre à l'objet et reste caché.

#### IV. DEFINITION D'UN SYSTEME GENERAL

##### A. SPECIFICATION GENERALE DU SYSTEME:CAHIER DES CHARGES

###### 1. CONTEXTE SCIENTIFIQUE ET ENVIRONNEMENT DU SYSTEME

###### a) PRINCIPES GENERAUX DE LA CARTOGRAPHIE GEOTECHNIQUE

###### (1) définition

La réalisation d'un système de gestion et de traitement graphique et cartographique des données géologiques et géotechniques réclame au préalable une connaissance, aussi sommaire soit elle, de la cartographie géotechnique, de sa conception, de sa relation avec l'urbanisme, de sa finalité, de ses exigences et de son contenu.

Précisons que dans l'ensemble de ce travail, le terme "géotechnique" est utilisé avec le sens suivant : "La géotechnique est destinée à regrouper les données les plus significatives des domaines de la mécanique et de la géologie"... "ces deux domaines sont si étroitement liés qu'il serait vain de demander lequel est le plus important" (Terzaghi)

###### (2) utilité

**Le choix du site:** L'objectif fondamental des auteurs des documents cartographiques semble être, et doit rester, la réalisation d'une ou de plusieurs cartes comportant les renseignements nécessaires et suffisants pour qu'un projecteur puisse sélectionner à priori, dans une région, les zones d'implantation possible d'une grande voie de communication et les zones globalement favorables à l'industrialisation et à tout projet d'urbanisme.

**Un moyen de communication :** Les cartes géotechniques sont considérées comme un instrument destiné aux hommes de bureaux d'études et de chantiers et visant à les sensibiliser aux techniques de la mécanique des sols et de la géologie ; en effet, ces cartes présentent un moyen de communication entre géotechniciens, géologues, réalisateur et constructeurs.

###### (3) contenu d'un dossier géotechnique

Un dossier géotechnique contient les documents suivants:

**\*Une carte lithologique :** C'est une carte qui présente la nature des couches géologiques affleurantes et leurs limites. Cette carte doit être accompagnée d'une légende.

**\*Les cartes des couches compressibles:** Une couche compressible peut - être la cause d'un tassement des ouvrages, il est nécessaire de désigner et de délimiter ces couches. On peut présenter les documents suivant:

-Une carte de limitation des couches compressibles affleurant.

-Une carte d'isoprofondeur des couches compressibles: Dans cette carte on relie par des courbes les points ayant la même profondeur de couche compressible.

-Une carte isopaque (d'isoépaisseur) : C'est une carte dont les courbes relient les points d'égale épaisseur de couches compressibles.

**\*Une carte du substratum :** Le substratum est la couche la plus dure du sol qu'on peut rencontrer en creusant. On peut produire des cartes d'isoprofondeur de substratum.

**\*Une carte de nappe :** Il s'agit d'une carte d'isoprofondeur de la nappe d'eau.

**\*Une carte de situation :** Il s'agit d'une carte qui indique tous les points de sondages, d'échantillonnages et d'essais mécaniques effectués sur le terrain.

###### b) DESCRIPTION DE L'INFORMATION BRUTE NECESSAIRE A LA PRODUCTION D'UN DOSSIER GEOTECHNIQUE

Dans les projets d'urbanisation, certaines informations concernant le terrain doivent être nécessairement disponibles.

Une campagne de reconnaissance peut nous fournir les principales informations dont on a besoin, il s'agit de :

\* **La lithologie du terrain** : Le géologue fournit un document illustrant la succession des couches géologiques rencontrées lors d'un forage (sondage), ce document est la coupe lithologique .

\* **L'information hydrogéologique** : Il s'agit souvent de la profondeur de la nappe d'eau dans le sous-sol en un point donné et à une date donnée.

\* **L'information granulométrique** : Celle-ci concerne les dimensions et la nature des grains d'un échantillon, prélevé soit en surface soit en profondeur .

\* **L'information mécanique** :

\* Les essais en laboratoires : certaines informations concernant les propriétés mécaniques ne peuvent être recueillies qu' à partir des essais mécaniques faits en laboratoire sur des échantillons.

\* Les essais in situ : Ce sont des manipulations qu'on peut exécuter sur le terrain et qui sont capables de nous donner une idée du comportement mécanique des différentes couches du terrain.

\* **L'information géophysique** : Le comportement du terrain vis à vis de la propagation des ondes ou de l'électricité est très important car il nous permet de déduire les caractéristiques des couches du sous-sol.

### c) *L'INFORMATISATION DU PROCESSUS*

Il est courant que l'on mène une campagne de reconnaissance d'un terrain qui a déjà été étudiée et dont les informations correspondantes ont été perdues; d'autant plus que la recherche manuelle d'une information à partir d'un tas de documents mal organisé est une opération fastidieuse.

D'où la nécessité pour une entreprise de génie civil de se procurer un système de stockage de toute l'information géotechnique disponible, sous une forme qui permet l'accès le plus simple possible à cette information en utilisant un système de gestion de base de données adéquat . L'informatisation ne concerne pas que le stockage des données mais aussi leur exploitation graphique et cartographique..

## 2. PRINCIPALES FONCTIONNALITES DU SYSTEME

### a) *LA REALISATION D'UNE BASE DE DONNEES*

Il s'agit du stockage de l'information géologique et géotechnique . Pour réaliser cette base de données on doit utiliser un système de gestion de base de données (S.G.B.D) du type S.Q.L ou dbase-4 . Ces systèmes ont les caractéristiques suivantes :

- Ce sont des systèmes relationnels : Un système relationnel est défini par un ensemble de schémas de relations et dont les opérations de gestion s'effectuent par l'intermédiaire des opérations de l'algèbre relationnel

- Ils sont destinés à des micro-ordinateurs compatibles avec "I.B.M pc".

Ces système de gestion de base de donnée permettent de :

- Saisir les données géologiques et géotechniques brutes  
- Saisir les dessins ( cartes ,coupes ,bloc-diagrammes) produits par le système ,sur demande de l'utilisateur

- Interroger la base de données.

### b) *L'ESTIMATION ET LE CALCUL*

#### (1) l'interpolation et l'estimation

Le système doit être capable d'estimer la valeur -ou les valeurs- d'une propriété (soit en un point déterminé par l'utilisateur, soit aux noeuds d'un réseau régulier ) à partir de valeurs mesurées aux points voisins sélectionnés de la base de données.

#### (2) le calcul

Il s'agit de formuler une bibliothèque de procédures de calcul qui utilisent des méthodes choisies pour la résolution des équations différentielles et pour le calcul matriciel. Ces procédures sont essentiellement utiles pour l'estimation et l'interpolation .

### c) LE DESSIN ET LA PRODUCTION DES GRAPHIQUES

Le système doit permettre de produire des graphiques (cartes, profils géologiques, coupe de sondage et blocs-diagrammes) à partir des données sélectionnées de la base. Il s'agit de restituer sous forme graphique une synthèse de données, en utilisant les résultats des programmes d'estimation et de calcul.

#### (1) les documents cartographiques

Il s'agit des cartes qui peuvent être contenues dans un dossier géotechnique et qui sont:

- les cartes lithologiques
- les cartes des couches compressibles
- les cartes du substratum
- les cartes de nappes d'eau
- les cartes de situation

On peut classer toutes ces cartes en trois catégories:

\* **Les cartes de données ponctuelles** : C'est la classe des cartes de situation dans lesquelles on dessine simplement des points correspondants aux endroits de forages, de prélèvement d'échantillons, de mesure ou d'acquisition des données. Ces cartes peuvent présenter les valeurs des mesures, en plus de leur localisation.

\* **Les cartes de "zoning"** : C'est la classe des cartes où on divise le terrain en plusieurs zones, selon un critère choisi. Par exemple, les cartes lithologiques divisent le terrain en plusieurs zones selon la nature des couches : calcaire, sable, argile, etc.

\* **Les cartes d'isovaleur** : C'est la classe des cartes où l'on dessine des courbes qui relient les points ayant les mêmes valeurs vis à vis d'une propriété choisie. Par exemple:

- les cartes topographiques
- les cartes d'isoprofondeur
- les cartes d'isoépaisseur

#### (2) les coupes graphiques de sondage

Le système doit pouvoir illustrer la description des couches rencontrées successivement dans le sous-sol (description fournie par le géologue) sous forme d'un schéma légendé.

#### (3) les profils géologiques

Il s'agit de visualiser en deux dimensions la succession des couches géologiques en profondeur, en reliant plusieurs forages entre eux.

#### (4) les blocs diagrammes

Le système dessine en trois dimensions, à partir de plusieurs sondages, la composition et la structure du sous-sol. Un bloc-diagramme met en relief la topographie et illustre la succession des couches en profondeur.

### 3. L'INFORMATION MANIPULEE PAR LE SYSTEME

Le système manipule trois classes d'information :

#### a) L'INFORMATION BRUTE

C'est l'information géologique et géotechnique collectée lors d'une campagne de reconnaissance. Ces informations doivent être stockées dans la base de données.

Il s'agit de:

- données lithologiques et géologiques
- données hydrologiques
- données granulométriques
- données mécaniques
- données géophysiques

### b) LES DONNEES GRAPHIQUES

Il s'agit des dessins produits par le système sur écran, sur imprimante ou sur table traçante.

L'information graphique peut être :

- un point
- un segment
- une courbe
- une carte
- un profil géologique
- un bloc-diagramme

### c) L'INFORMATION INTERMEDIAIRE

Pour pouvoir produire les sorties graphiques, le système doit créer une classe d'informations intermédiaires qui permet le passage des données géologiques brutes aux données graphiques .

## 4. CONTRAINTES ET EXIGENCES

### a) LES CONTRAINTES

#### (1) la nature sémantique de l'information

L'expression des propriétés d'un terrain est le plus souvent ,soit numérique ,soit sémantique

- information numérique : exemple ,les résultats d'un essai mécanique.

- Information sémantique : l'exemple suivant illustre une description d'une couche géologique:

"Une couche marnocalcaire -passant à - calcaire massif gris, contenant marne gris alternant avec calcaire gris-bleu très fin à nodule calcaire".

Pour ce genre d'information sémantique,il est nécessaire de mettre au point une nomenclature formalisée.

Toute formalisation s'accompagne d'une distorsion de l'information formalisée et d'une perte d'information . Il s'agit de retenir la formalisation qui minimise la distorsion et la perte d'information.

#### (2) l'utilisation d'un micro-ordinateur de type PC

En matière de géotechnique et de géologie, plusieurs logiciels sur grands ordinateurs ont été réalisés en France et dans le monde ;ils ont eu pour but de stocker les données d'un sous-sol et de produire selon le cas ,soit des cartes d'aptitude liées aux problèmes d'urbanisme ,soit des cartes de glissement de terrain.

La difficulté d'usage de ces logiciels ,s'ajoutant aux contraintes économiques a fait de ces logiciels des produits peu diffusés.

L'orientation vers la micro-informatique ne représente pas nécessairement une limitation ,le bon usage de la micro-informatique peut conduire plutôt à une amélioration par rapport à l'usage des grands ordinateurs .

Nous avons choisi de développer notre système sur un micro-ordinateur pour les raisons suivantes :

- Le coût peu élevé des micro-ordinateurs.
- La possibilité d'utilisation des micro-ordinateurs dans l'acquisition automatique des mesures dans les laboratoires.
- La portabilité des micro-ordinateurs permet l'acquisition des mesures sur le terrain.
- La facilité à se procurer des logiciels de géologie et de géotechnique sur le marché et qui fonctionnent sur micro-ordinateur.

Le micro-ordinateur disponible est de type compatible pc Cela facilite la portabilité, et donc la diffusion du logiciel produit.

## (3) l'utilisation du turbo pascal

Le choix du pascal comme langage est justifié par les raisons suivantes:

- Pascal est un langage procédural : Cela permet une programmation modulaire qui facilite la maintenance et l'amélioration du logiciel et sa modification en cas de changement de matériel .

On peut donc subdiviser le logiciel en plusieurs modules indépendants mais qui communiquent entre eux par l'intermédiaire des interfaces.

## (4) l'utilisation d'un S.G.B.D relationnel

L'utilisation d'un S.G.B.D relationnel permet une meilleure gestion des ressources . la gestion de la base de données est prise en charge par le s.g.b.d ,elle est donc transparente aux programmeurs , mais il faut auparavant définir les relations entre les différentes entités d'une façon algébrique.

Dbase-4 est un système de gestion de base de données relationnel qui fonctionne sur micro-ordinateur . La compatibilité entre le d-base4 et le pascal dans un même logiciel est assurée par un compilateur délivré avec le logiciel dbase4. Cependant il est nécessaire de définir un interface entre la banque de données et le logiciel de traitement.

## (5) contrainte de place mémoire

Une campagne de reconnaissance produit une quantité très importante de données ,l'utilisation du micro-ordinateur ,si elle présente un avantage économique , pose le problème de la capacité de la mémoire. Il faut donc résoudre ce problème à l'étape de l'analyse et de la réalisation.

*b) LES EXIGENCES*

## (1) la transparence du logiciel

Le logiciel produit doit être complètement transparent à l'utilisateur ,celui-ci doit complètement ignorer la façon dont le logiciel a

été fait, et d'autre part,le logiciel doit être très facile à utiliser pour un utilisateur non-initié à l'informatique.

## (2) la portabilité du système

Dans le cas de changement de matériel , le logiciel doit toujours fonctionner au prix de quelques modifications peu importantes .

## (3) la maintenance et l'adaptabilité du système

L'amélioration du logiciel doit être possible et facile Cela concernera les possibilités suivantes :

-La rectification de certaines fonctionnalités.

-L'intégration de nouvelles fonctionnalités:

-L'intégration des corrélations entre les paramètres géotechniques.

-L'intégration des fonctionnalités géostatistiques.

-L'intégration de la saisie automatique des données en laboratoires et in situ.

-L'élimination de certaines fonctionnalités .

-La facilité de changement du système de gestion de base de données utilisée.

## (4) l'interactivité du logiciel

Une des caractéristiques essentielles du logiciel est l'interactivité . Cela concerne la saisie et l'interrogation sélective de la base de données mais aussi la production des documents graphiques .

Le système permet l'intervention du géologue utilisateur pour compléter et corriger un dessin selon son expérience dans le domaine qui échappe au système informatique .

En ce qui concerne les cartes , les coupes géologiques et les bloc-diagrammes ,le choix des légendes et des couleurs correspondant aux couches illustrées se fait d'une manière interactive.

(5) l'ergonomie

Le logiciel doit présenter une qualité ergonomique importante , un menu doit être présenté .Le choix doit être effectué par l'intermédiaire de la souris . Ce même outil doit être utilisé pour corriger et compléter un dessin.

#### 5. LA PRIORITE DE REALISATION

Cependant la priorité a été donnée à la réalisation des fonctionnalités de la cartographie et à la réalisation de la visualisation tridimensionnelle des données d'un site en passant par la réalisation d'une bibliothèque graphique tridimensionnelle .

## B. ANALYSE GENERALE DES FONCTIONNALITES

### 1. PRESENTATION DE L'ANALYSE FONCTIONNELLE

#### a) INTRODUCTION

L'analyse du système de gestion et de traitement graphique des données géotechniques a été effectuée par la méthode Structured Analysis Design Technic (S.A.D.T) .

Le résultat de cette analyse se traduit dans ce chapitre par une décomposition fonctionnelle .

Le point de vue sélectionné dans cette analyse est celui de l'utilisateur . Ce choix se justifie par le fait qu'il a été nécessaire de concevoir ce système en fonction des fonctionnalités offertes à l'utilisateur plutôt que du point de vue du constructeur ou de la maintenance . On a donc adopté le vocabulaire géotechnique.

#### b) LA METHODE S.A.D.T

##### (1) présentation

Une première description du système reste trop générale et peu précise . Pour comprendre en détails les fonctionnalités du système et leurs interactions , il nous a paru intéressant d'adapter cette méthode .

##### (2) justification

Cette méthode permet de :

- Placer dans leur contexte toutes les fonctionnalités du système .
- Décomposer les fonctionnalités en sous-fonctionnalités.
- Dégager une première hiérarchie.
- Simuler sur le papier le fonctionnement du système.
- Définir les principales données véhiculées .

## (3) La syntaxe

La syntaxe de la méthode S.A.D.T est définie par les règles suivantes :

1- Dans un diagramme chaque fonctionnalité est représentée par une boîte . Moins de trois boîtes serait insuffisant , plus de six boîtes serait trop complexe .

2- Les boîtes sont interconnectées par des flèches qui montrent les contraintes et les relations .

3- Une flèche peut représenter une entrée , une sortie, un contrôle ou un mécanisme .

4- Une entrée est convertie par l'activité en une sortie . L'activité transforme l'information apportée par l'entrée , change sa position ou change son état .

5- Un contrôle n'est pas modifié par l'activité . Il se traduit généralement par un ordre qui déclenche l'activité ou la contraint . Une boîte d'activité doit avoir au moins un contrôle .

6- Une sortie est générée par l'activité . Chaque boîte doit avoir au moins une sortie .

7- Un mécanisme est un outil utilisé pour réaliser ou aider à réaliser l'activité .

8- Un diagramme est la décomposition d'une des boîtes du diagramme parent , c'est ce qui permet la décomposition fonctionnelle du système.

2. DECOMPOSITION FONCTIONNELLE DU SYSTEME

Le système a été décomposé au plus haut niveau en quatre groupes de fonctionnalités qui sont :

- Saisir ,interroger et gérer la base de données
- Dessiner
- estimer
- Interpoler ,lisser ,et calculer

Chacune de ces fonctionnalités est la racine d'une arborescence de fonctions et fera l'objet d'un des paragraphes ci-dessous

*a) MANIPULER LA BASE DE DONNEES*

(1) but

Il s'agit d'exécuter les tâches suivantes :

**- La saisie :**

- La saisie des données géotechniques brutes :

- La saisie des données lithologiques
- La saisie des données hydrologiques
- La saisie des données mécaniques
- La saisie des données géophysiques.

- La saisie des données traitées.

- La saisie des données graphiques.

**- L'interrogation sélective :**

- La récupération des données brutes :

- La récupération des données en un point défini

par ses coordonnées ,

- la récupération des données concernant une zone définie par ses coordonnées.

Ces données peuvent être récupérées sur écran ou sur imprimante .

- La récupération d'un dessin :

- La récupération d'un dessin illustrant la succession des couches en un point (coupe lithologique correspondant à un sondage en un point donné )

- La récupération d'une carte ou d'un bloc-diagramme concernant une zone définie par les coordonnées des points de sommets.

Les dessins peuvent être récupérés sur l'écran , sur l'imprimante ou sur la table traçante .

## (2) principe

### (a) Les besoins de l'utilisateur

Le système présente à l'utilisateur un menu des fonctionnalités de la manipulation de la base de données . Pour chacune de ces fonctionnalités , le système présente un sous menu . Le choix effectué par l'utilisateur correspond à la fois à un ordre d'exécution et à une requête formulée selon le choix de l'utilisateur .

Il peut s'agir :

- D'une requête de saisie

\_ D'une requête d'interrogation sélective de données ou de dessin . L'utilisateur doit alors mentionner les coordonnées d'un point s'il s'agit de données ponctuelles , ou de l'ensemble des points repérés délimitant les zones ou les dessins concernés.

- D'une requête d'effacement de données . L'utilisateur doit mentionner les coordonnées et les clefs.

### (b) besoin du système

ces requêtes peuvent être formulées également sur ordre interne du système , en effet quand l'utilisateur demande l'exécution d'un dessin ou d'une estimation concernant une zone déterminée par ces coordonnées, le système se charge de sélectionner les données nécessaires pour exécuter les tâches demandées par l'utilisateur .

### (3) contrôle

Avant de stocker les données brutes ou les dessins , le système vérifie si l'information à stocker n'existe pas déjà dans la base et produit un code selon les résultats de son contrôle .

Un contrôle similaire est effectué pour l'effacement de l'information : Le système s'assure d'abord de son existence dans la base.

L'exécution de chacune de ces fonctionnalités de manipulation de la base de données est déclenchée par un ordre. Chacun de ces ordres est formulé sous forme d'une requête de manipulation de base de données . Un ordre peut être déclenché par l'utilisateur ou par le système lui-même.

Un des paramètres les plus importants de la fonctionnalité de l'interrogation sélective concerne les coordonnées. Il peut s'agir des types de coordonnées suivants:

- Les coordonnées d'un point :C'est le cas lorsque l'utilisateur demande l'exécution d'un dessin de la coupe lithologique (succession des couches ) à partir d' un forage exécuté en ce point . Cela peut être aussi le cas d'une estimation en un point de la valeur d'une caractéristique .

\_ Les coordonnées de quatre points délimitant une zone : C'est le cas où l'utilisateur demande l'exécution d'une des tâches suivantes :

- La récupération de toutes les données ou d'une catégorie de données concernant cette zone ,

- la production d'une carte de zoning ou d'isovaleur recouvrant cette zone ,

- la production d'un bloc diagramme de la zone.

#### (4) mécanismes

- La syntaxe du s.g.b.d : le système de gestion de base de données propose un langage de programmation pour l'écriture des requêtes ce qui correspond à une syntaxe spécifique .

- Les relations algébriques : On utilise un système de base de données relationnel . Les requêtes et les fichiers doivent être d'abord formulés sous forme de relations algébriques.

#### b) DESSINER

La fonctionnalité "dessiner" a trois objectifs distincts:

- Produire un dessin : Il s'agit de dessiner une coupe lithologique d'un sondage , une carte ,un profil géologique ou un bloc-diagramme .

- Manipuler un dessin : faire subir à un dessin une des transformations géométriques suivantes:

- translation,
- rotation,
- changement d'échelle,
- changement de point de vue.

- Corriger et modifier interactivement un dessin.

#### (1) produire un dessin

##### (a) but

Il s'agit d'afficher à l'écran ,imprimer sur une imprimante ou sur une table traçante un des graphiques suivants:

-Un log lithologique : Ceci illustre la succession des couches rencontrées lors d'un forage en un point donné.

-Un profil géologique: celui-ci est une coupe en profondeur reliant plusieurs forages entre eux . Elle illustre la succession , les relations entre les couches ,leurs continuités et leurs déformations.

-Une carte : c'est un document qui illustre la variation de valeur d'une caractéristique du sol ou du sous-sol dans une zone déterminée. Le système peut produire deux sortes de cartes:les cartes d'isovaleurs et les cartes de zoning .

#### (b) principe

Un dessin sur l'écran est le produit d'une combinaison d'un modèle géométrique représentant les données du terrain et d'une matrice représentant la transformation de visualisation.

Le terrain est assimilé à l'espace utilisateur ou l'espace de travail.

la matrice est définie par l'emplacement du dessin à l'écran et par l'échelle de représentation , elle est modifiée à chaque transformation géométrique telle que: rotation, zoom, réduction et changement de point de vue .

#### (c) dessiner une carte

Il s'agit d'effectuer une des tâches suivantes :

-Dessiner une carte d'isovaleur qui peut correspondre selon la propriété étudiée à une des cartes suivantes :

- Une carte topographique,
- une carte d'iso-épaisseur d'argile superficielle,
- une carte d'isoprofondeur de la nappe,
- une carte du substratum.

-Dessiner une carte de zoning : le système peut fournir pour une zone donnée une des cartes suivantes :

- Une carte lithologique de surface,
- une carte lithologique à une profondeur donnée .

i) dessiner une carte d'isovaleur

Le principe du dessin de ce type de carte est le suivant:

-L'utilisateur définit la caractéristique à cartographier .

-Le système récupère ,d'après cette caractéristique et selon la zone concernée ,les données brutes de la base de données et les transforme en un réseau régulier de données ou en une triangulation.

-L'utilisateur peut proposer des valeurs des courbes à tracer ou laisser le choix au système le soin de les définir systématiquement une à une ,à partir du réseau ou de la triangulation.

-Le système définit la suite des mailles ou des triangles par lesquels une courbe de valeur Z doit passer; puis ,il donne un ordre aux fonctionnalités d'interpolation ,de lissage et de calcul pour que celles-ci lui fournissent les points par lesquels la courbe doit passer, et les tangentes de la courbe en ces points.

ii) dessiner une carte lithologique

C'est une fonctionnalité différente de la précédente car dans ce genre de carte , le terrain étant divisé en domaines d'égal-valeur d'une caractéristique, les courbes correspondent aux limites de ces plateaux

(d) dessiner un profil géologique

Pour le dessin d'un profil géologique ou celui d'un simple sondage on a besoin des coupes lithologiques . Le système les récupère à partir de la base .

L'exécution de cette fonctionnalité doit être interactive . En effet, elle doit offrir à l'utilisateur les possibilités suivantes :

-Le système s'occupe simplement de dessiner la succession des couches pour chaque sondage .L'utilisateur doit donc dessiner d'une façon complète les liaisons entre coupe lithologique .

-L'utilisateur corrige et modifie le dessin :C'est le cas ou le système s'occupe de dessiner les liaisons ,la continuité et les déformations des couches à partir d'un mini-système expert . On détaillera la possibilité de correction dans le paragraphe concernant la fonctionnalité de la modification .

(e) réaliser un bloc-diagramme

La réalisation de cette fonctionnalité passe par une fonctionnalité de modélisation géométrique du volume à afficher.Cette modélisation s'effectue à partir des profils géologiques reliant les forages périphériques et à partir de la modélisation de la surface du terrain.

(2) manipuler un dessin

-La manipulation d'un dessin se résume en une des transformations géométriques suivantes : rotation , changement d'échelle ou changement de point de vue .

-L'utilisateur doit fournir les paramètres de cette transformation .

-Le système rectifie d'abord la matrice géométrique de la visualisation selon les nouveaux paramètres .

-Par la suite le système applique la matrice géométrique sur les données de terrain (modèle défini par des coordonnées espace-utilisateur).

-La manipulation d'un dessin ne présente pas une nouvelle fonctionnalité . En effet pour transformer un dessin on réexécute la fonctionnalité de production d'un dessin mais avec les nouveaux paramètres.

(3) modifier un dessin

(a) but

Il s'agit d'offrir à l'utilisateur les possibilités suivantes:

-Effacer partiellement un dessin et le rectifier par l'intermédiaire de l'outil de dessin interactif qui peut être la souris

-choisir les légendes et les couleurs.

(b) principe

La correction concernera surtout les courbes et les limites de couches . Cela affectera nécessairement le modèle géométrique choisi pour représenter le terrain .

Le choix de légende et de couleur consiste à sélectionner une couleur pour une zone correspondante à une couche dans une carte ou dans un bloc-diagramme.

Le stockage d'un dessin modifié ne peut pas se faire en stockant la matrice de transformation mais se fait en stockant l'intégrité des pixels de l'image.

c) *ESTIMER*

(1) but

Il s'agit de calculer les valeurs d'une caractéristique étudiée aux noeuds d'un réseau régulier, à partir d'un amas de valeurs mesurées en un ensemble de points irrégulièrement espacés et recouvrants la même zone.

(2) principe

-L'estimation géostatistique est une estimation basée sur le krigeage universel. Il s'agit d'une approche probabiliste qui permet d'estimer la valeur d'une caractéristique en un point à partir des points voisins; ces calculs se basent sur la théorie des variables régionalisées et fournissent pour chaque valeur estimée une précision sur le calcul.

-L'estimation analytique (numérique) : à partir des valeurs mesurées d'une caractéristique, on définit une fonction continue d'une variable  $z=f(x,y)$  . Cela permet d'estimer les valeurs aux noeuds d'un réseau régulier mais sans donner une précision sur le calcul.

d) *INTERPOLER,LISSER,CALCULER*

(1) but

Il s'agit d'effectuer les fonctionnalités suivantes :

-Définir l'image d'un point à partir de son abscisse et de l'équation d'une fonction .

-Définir la tangente d'une courbe en un point donné.

-Définir l'intersection entre deux courbes , deux surfaces,une surface et une courbe ...

-résoudre les équations différentielles et effectuer des calculs matriciels

(2) principe

Cette fonctionnalité correspond à un ensemble de sous fonctionnalités indépendantes .C'est une sorte de bibliothèque de calcul et d'analyse numérique dont le système a besoin pour exécuter d'autres fonctionnalités, notamment les fonctionnalités de dessin et d'estimation . Le choix des techniques appropriées dépend des problèmes à résoudre et de la capacité de mémoire disponible.

### C. LA DEFINITION DES ENTITES REELLES : LE MODELE ENTITE-ASSOCIATION

C'est la deuxième phase de notre méthodologie . Dans le chapitre précédent nous avons défini les fonctionnalités du système et nous les avons décomposés en fonctionnalités élémentaires . Dans ce chapitre ,on s'occupe d'abord de la définition des entités constituant le domaine que nous avons à modéliser . Ces entités représentent l'information collectées lors d'une campagne de reconnaissance . Nous allons ensuite définir les liaisons (association) qui existent entre les entités ou les ensembles d'entités . Le schéma-Z résume d'une façon simplifiée la structure qui relie les entités entre elles.

#### 1. LA PRESENTATION DU MODELE ENTITE-ASSOCIATION

##### a) LA DEFINITION DU MODELE

Un modèle est un ensemble de concepts pour décrire une certaine perception de la réalité . Dans ce modèle ,on essaie de définir un certain nombre d'entités à partir des objets du monde réel. Cela représente déjà un premier niveau d'abstraction .

Le deuxième niveau contient la définition des ensembles abstraits d'entités déjà définies.

Les relations qui peuvent exister entre les différents objets du monde réel seront traduites d'une façon abstraite par des associations entre entités .

Toutes les associations ne sont pas permises. , Cela implique qu'il faut définir des conditions d'associations selon le problème réel à traiter.

Les associations entre les différents ensembles d'entités sont regroupées et exprimées selon un schéma appelé "schéma-Z"

##### b) LA JUSTIFICATION

###### (1) approche relationnelle de base de données

Les associations définies avec le modèle entité association vont pouvoir être exprimées d'une façon algébrique, d'où l'approche relationnelle du modèle .

Le modèle entité-association utilisé en phase de conception nous prépare pour une phase ultérieure ; c'est la phase de construction et de réalisation de la base de données relationnelles.

En effet , on peut déduire un schéma de dépendance fonctionnelle à partir du schéma -Z produit dans la phase de conception, pour pouvoir formuler en langage relationnel les différentes requêtes de saisie et d'interrogation sélective des données

###### (2) passage à la méthode orientée objet

La définition des entités abstraites et des ensembles d'entités dans le modèle entité- association permettra de définir plus facilement les objets pendant l'utilisation de la méthode orientée objet dans la phase de conception qui prépare la réalisation.

##### c) L'APPLICATION SELECTIVE

Le modèle entité association ne va être appliqué que pour une partie de l'information manipulée par le système ; c'est l'information brute ,c'est à dire l'information collectée lors d'une campagne de reconnaissance

#### 2. L'ETUDE DE L'INFORMATION COLLECTEE LORS D'UNE CAMPAGNE DE RECONNAISSANCE

Un dossier peut regrouper toute l'information collectée lors d'une campagne de reconnaissance d'un terrain ou lors des travaux concernant un projet.

Un dossier est caractérisé par :

- Un numéro
- la date des travaux de reconnaissance

-des renseignements généraux sur les entreprises et les laboratoires qui ont participé aux travaux.

-un ou plusieurs groupes d'information concernant les travaux suivants:

\*Des travaux donnant suite à une coupe géologique décrivant la succession des couches rencontrées dans le sous-sol. Il peut s'agir de :

- forages
- puits d'exploration
- carrières ou gravières (extraction de matériaux)

\*Des essais qui se font sur le terrain (in situ):

-sondages pressiométriques : on effectue d'abord un forage et on exerce sur une partie du sous-sol une pression, par l'intermédiaire d'un instrument gonflable, situé dans le trou provoqué par le forage. Cette opération nous donne une idée sur la résistance mécanique de la couche concernée.

-sondages pénétrométriques : il s'agit d'enfoncer une pointe dans le sol et mesurer la résistance du sol à l'enfoncement de la pointe du pénétromètre. Cette résistance reflète la nature mécanique du sol. Il existe deux types de pénétromètres :

+le pénétromètre dynamique : l'enfoncement se fait grâce au poids d'un mouton qu'on soulève et qu'on lâche un certain nombre de fois (cela varie selon le sol) afin d'enfoncer la pointe à une profondeur déterminée.

+le pénétromètre statique : l'enfoncement se fait grâce à un moteur et la résistance se mesure grâce à un baromètre.

-le sondage scissométrique : on essaie de cisailier le sol à une certaine profondeur et on mesure sa résistance au cisaillement.

\*Des études géophysiques :

-La sismique : Il s'agit d'étudier la propagation des ondes dans le sol, cette propagation donne une idée sur la nature du sol.

-La résistivité électrique : Il s'agit de mesurer la résistivité électrique du sol.

\*Essai aux laboratoires : On prélève des échantillons auxquels on fait subir les essais suivants :

-Les essais granulométriques : C'est l'étude de l'éventail de diamètres et de la nature des grains qui constituent l'échantillon.

-Les essais de cisaillement : On essaie de cisailier le sol sous plusieurs pressions. La variation du comportement de l'échantillon en fonction de la variation de la pression indique la nature mécanique du sol.

-Les essais de compressibilité : Il s'agit de compresser l'échantillon. La variation de la valeur de la compressibilité selon la pression exercée, reflète les propriétés du sol et donne une indication sur les possibilités de tassement du terrain.

### 3. L'APPLICATION DU MODELE

#### *a) LA DETERMINATION DES ENSEMBLES D'ENTITES COMPOSANT LE SYSTEME D'INFORMATION*

\*Dossier : C'est l'ensemble de dossier qui renferment les données, chaque dossier correspond à une campagne de reconnaissance ou à un projet.

\*Date : C'est l'ensemble des dates correspondant aux travaux réalisés.

\*Renseignements : c'est l'ensemble des indications géographiques et administratives concernant les municipalités, les centralisateurs et les entreprises qui ont fait les travaux.

\*Zone : C'est l'ensemble des zones concernées par les reconnaissances . Une zone est déterminée par les coordonnées de quatre points du terrain.

\*Forage : C'est l'ensemble des forages qui ont été effectués.

\*Numéro : C'est un ensemble d'entiers . Cela peut correspondre à :

- Un numéro de zone,
- un numéro de dossier ,
- un numéro de forage,
- un numéro d'échantillon

\*Coor-Lambert : C'est un ensemble de couples de réels qui peuvent correspondre :

- à des coordonnées de forage
- aux limites des zones

\*Degrés : C'est l'ensemble de réels correspondant aux directions des forages et à leurs pendages .

\*Métrage : C'est un ensemble de réels exprimés en unité de mètre . Cela peut correspondre :

- aux cotes de forages,
- aux profondeurs des couches ,
- aux profondeurs des nappes,
- aux épaisseurs des couches ,
- aux épaisseurs des échantillons
- aux diamètres des échantillons,
- aux diamètres des forages

\*Bool : C'est un ensemble de valeurs logiques définies par extension "vrai ou faux " .

\*Coup-géol : C'est l'ensemble des coupes géologiques effectuées .

\*T-définition : C'est un ensemble de termes permettant de définir la lithologie d'une couche . Il n'est pas nécessaire à présent de définir cet ensemble en extension . On peut le définir donc par un ensemble de chaînes de caractères .

\*T-description : c'est un ensemble de termes permettant de décrire une composante d'une couche ; chaque terme est représenté par une chaîne de caractères .

\*Niveau-desc :C'est un ensemble d'entiers qui définissent le niveau de description . Un entier indique par exemple, si le terme de description concerne la composante principale ou la secondaire .

\*T-coordination : C'est un ensemble d'entiers décrivant la nature de la relation entre deux composantes d'une même couche. Il peut s'agir :

- d'une inclusion
- d'une juxtaposition
- d'une alternance

\*Echantillon : C'est l'ensemble des échantillons qui peuvent être prélevés et qui peuvent subir des essais .

\*C-échantillon : C'est l'ensemble des chaînes de caractères indiquant l'état et la nature de l'échantillon .

\*Pourcentage : C'est un ensemble de valeurs entre 0 et 100 qui peuvent correspondre à :

- un pourcentage de récupération
- un pourcentage de composition minéralogique
- une valeur tamisat
- un R.Q.D

\*Carac-tech : C'est l'ensemble de chaînes de caractères décrivant les caractéristiques techniques du forage .Cela pourra concerner :

- Le type de machine
- l'outil utilisé

- la méthode de forage
- le lubrifiant
- la nature de la descente

\*N-classe : C'est un ensemble de chaînes de caractères pour désigner les normes suivies pour la classification des sols : ucsc, afnor, lcpc ...

\*Classe -sol : C'est l'ensemble des classes possibles pour un sol, selon les normes choisies.

\*Densité: c'est un ensemble de réels exprimés en unité de g/centimètre cube . Cela peut correspondre à :

- La densité sèche
- la densité solide

\*T-eau : Il s'agit de la teneur en eau des échantillons . On la définit par un ensemble de réels qui s'expriment en pourcentage ,mais qui peuvent dépasser la valeur de 100 , Cela peut correspondre à :  $w, w_{opt}, w_s, l, l_p$

\*Pression -cont : C'est un ensemble de valeurs exprimées en pascal. Cela peut correspondre :

- au module de Young,
- à la cohésion ,
- à la résistance de pointe p, p+m
- à la pression,
- au module pressiométrique ,
- à la pression limite , à la pression de fluage...

\*Indi-coef : C'est un ensemble de réels qui indiquent des mesures sans unité , Cela peut concerner :

- l'indice des vides
- l'indice de consolidation
- le degré de saturation

\*Cisaillement : C'est l'ensemble des essais de cisaillement

\*compressibilité: C'est l'ensemble des tests ou des essais de compressibilité .

\*Granulométrie : C'est l'ensemble des essais granulométriques.

\*E-scissométriques : C'est l'ensemble des essais scissométriques.

\*Pénétro-stat: C'est l'ensemble des essais de pénétromètre statique.

\*Pénétro-dyna : C'est l'ensemble des essais de pénétromètre dynamique

\*E-résistivité : C'est l'ensemble des essais de résistivité

\*V-résistivité : C'est l'ensemble des valeurs de résistivité .

#### b) LES ASSOCIATIONS ENTRE ENTITES

La définition des entités et des ensembles d'entités n'est pas suffisante pour décrire le domaine à modéliser . La définition des associations entre ces entités est indispensable pour traduire la structure physique qui reliant ,entre elles, les entités du monde réel.

##### (1) définition

Dans ce chapitre on représente les schémas des relations entre les entités . Une association est la relation (fonction) entre deux ensembles d'entités

Une association est caractérisée par le type de chaque fonction :

- monovaluée ou multivaluée,
- partielle ou totale,
- la cardinalité minimale et maximale .

Une fonction est monovaluée si son évaluation délivre une seule valeur, par opposition à multivaluée où son évolution fournit un ensemble de valeurs .

Une fonction est partielle s'il existe des éléments dans l'ensemble de départ qui n'ont pas d'image et une fonction est totale si tous les éléments de l'ensemble de départ ont une image .

### (2) notation

Pour les notations concernant les noms d'entités, les noms d'ensembles d'entités, les associations et les fonctions , nous avons adopté les conventions suivantes :

- les noms d'entités commencent par une lettre minuscule ou un chiffre;
- les noms d'ensemble commencent par une lettre majuscule ;
- une fonction monovaluée commence par une lettre minuscule et symbolisée par une flèche simple (<- ou -->);
- une fonction multivaluée commence par une lettre majuscule et symbolisée par une flèche double (<<- ou -->>);
- une fonction partielle est symbolisé par la barre incliné(-/-) .

### (3) présentation

-Dossier	num <-/------>	numéro
	Centralisateur <<-/------/->>	renseignement
	Entreprise <<-/------/->>	
-Zone	num <-/------>	numéro
	Coordonnées <-/------>>	coor-Lambert

-forage	zonage <<-/------>	zone
	coordonnées <-/------>	coor-lambert
	num <-/------/->	numéro
	municipalité <<-/------/->	renseignement
	Centralisateur <<-/------/->>	
	Entreprise <<-/------/->>	
	direction <<-/------/->	degrés
	pendage <<-/------/->	
	élévation(N.G.F) <<-/------>	métrage
	profondeur <<-/------>	
	toit du substrat <<-/------>	
	nappe <<-/------>	

	Echantillon	<<-/->>	échantillon
	coupe verticale	<<-/->	coup-géo
	coupe technique	<<-/->	description-tech
-Descri- ption-tech	prof-sup	<<-/->	métrage
	prof-inf	<<-/->	
	diamètre	<<-/->	
	machine	<<-/->	carac-tech
	outil	<<-/->	
	méthode	<<-/->	
	lubrifiant	<<-/->	
	descente	<<-/->	
	tubage	<<-/->	
coupe-géol	Couches	<<-/->>	couche géologique
*couche géologique	Desc-couche	<<-/->>	(t-définition* t-description* niveau-desc* t-coordination age*formation )

	Description	<<-/->>	c-échantillon
*échantillon	coordonnées	<<-/->	coor-Lambert
	profondeur	<<-/->	métrage
	diamètre	<<-/->	
	épaisseur	<<-/->	
	% récupération	<<-/->	pourcentage
	r.q.d	<<-/->	
	dossier	<<-/->	numéro
	num ds le dossier	<<-/->	
*essai- identification	sujet	<<-/->	échantillon
	num ds le dossier	<<-/->	numéro
	norme	<<-/->	n-classe
	d-sèche	<<-/->	densité
	d-solide	<<-/->	
	ll	<<-/->	t-eau
	lp	<<-/->	
	w	<<-/->	

% mo	<<-/-/-----/->	pourcentage
n-perméa	<<-/-/-----/->	norme-p
v-perméa	<<-/-/-----/->	k
sujet	<<-/-/-----/->	échantillon
*granulométrie	<<-/-/-----/->	
num ds le dossier	<<-/-/-----/->	numéro
diamètre-max	<<-/-/-----/->	métrage
20 mm	<<-/-/-----/->	
5mm	<<-/-/-----/->	pourcentage
2mm	<<-/-/-----/->	
0.4mm	<<-/-/-----/->	
0.2mm	<<-/-/-----/->	
0.08mm	<<-/-/-----/->	
<0.01mm	<<-/-/-----/->	
<0.002mm	<<-/-/-----/->	
autre	<<-/-/-----/->>	minéralogie

cisaillement	<<-/-/-----/->	échantillon
sujet	<<-/-/-----/->	
num ds le dossier	<<-/-/-----/->	numéro
résis-remaniée	<<-/-/-----/->	pression-cont
résis-intact	<<-/-/-----/->	
module-théo	<<-/-/-----/->	
modul-expér	<<-/-/-----/->	
cohésion-drainée	<<-/-/-----/->	
cohésion-n-d	<<-/-/-----/->	
frottement-d	<<-/-/-----/->	angle
frottement-n-d	<<-/-/-----/->	
nature	<<-/-/-----/->	
*compressibilité	<<-/-/-----/->	type
sujet	<<-/-/-----/->	échantillon
num ds le dossier	<<-/-/-----/->	numéro
indice-vide	<<-/-/-----/->	indi-coef
i-vinitiale	<<-/-/-----/->	
i-compression	<<-/-/-----/->	
coef-consolidation	<<-/-/-----/->	

	p-eff-en place	
<<-/-/-----/->		pression-cont
	p-consolidation	
<<-/-/-----/->		
	p-eff-fin	
<<-/-/-----/->		
	degrés-sat	
<<-/-/-----/->		pourcentage
	rapport-comp	
<<-/-/-----/->		bool *pressiomètre
	dossier	
<<-/-/-----/->		numéro
	coordonnées	
<<-/-/-----/->		coord-Lambert
	forage	
<<-/-/-----/->		forage
	nature	
<<-/-/-----/->		type-pressi
	Niveau	
<<-/-/-----/->>		(prof*module*pl*pf)
	dossier	
*relevé piézo <<-/-/-----/->		numéro
	num ds le dossier	
<<-/-/-----/->		
	coordonnées	
<<-/-/----->		coord-Lambert
	forage	
<<-/-/-----/->		forage
	type	
<<-/-/-----/->		type-piézo
	limit-sup	
<<-/-/----->		métrage
	limit-inf	
<<-/-/----->		
	Relevé	
<<-/-/----->>		(date*métrage)

	dossier	
*pompage <<-/-/-----/->		numéro
	num ds le dossier	
<<-/-/-----/->		
	coordonnées	
<<-/-/----->		coord-Lambert
	forage	
<<-/-/-----/->		forage
	nature	
<<-/-/-----/->		type-pompe
	Relevé	
<<-/-/----->>		(métrage*k)
	dossier	
*pénétro-stat <<-/-/-----/->		numéro
	num ds le dossier	
<<-/-/-----/->		
	coordonnées	
<<-/-/----->		coord-Lambert
	forage	
<<-/-/-----/->		forage
	nature	
<<-/-/-----/->		type-pénétro-stat
	Relevé	
<<-/-/----->>		(métrage*p*p+m)
	dossier	
*pénétro-dyna <<-/-/-----/->		numéro
	num ds le dossier	
<<-/-/-----/->		
	coordonnées	
<<-/-/----->		coord-Lambert
	forage	
<<-/-/-----/->		forage
	nature	
<<-/-/-----/->		type-pénétro-dyna
	Relevé	
<<-/-/----->>		(métrage*nb de coups)

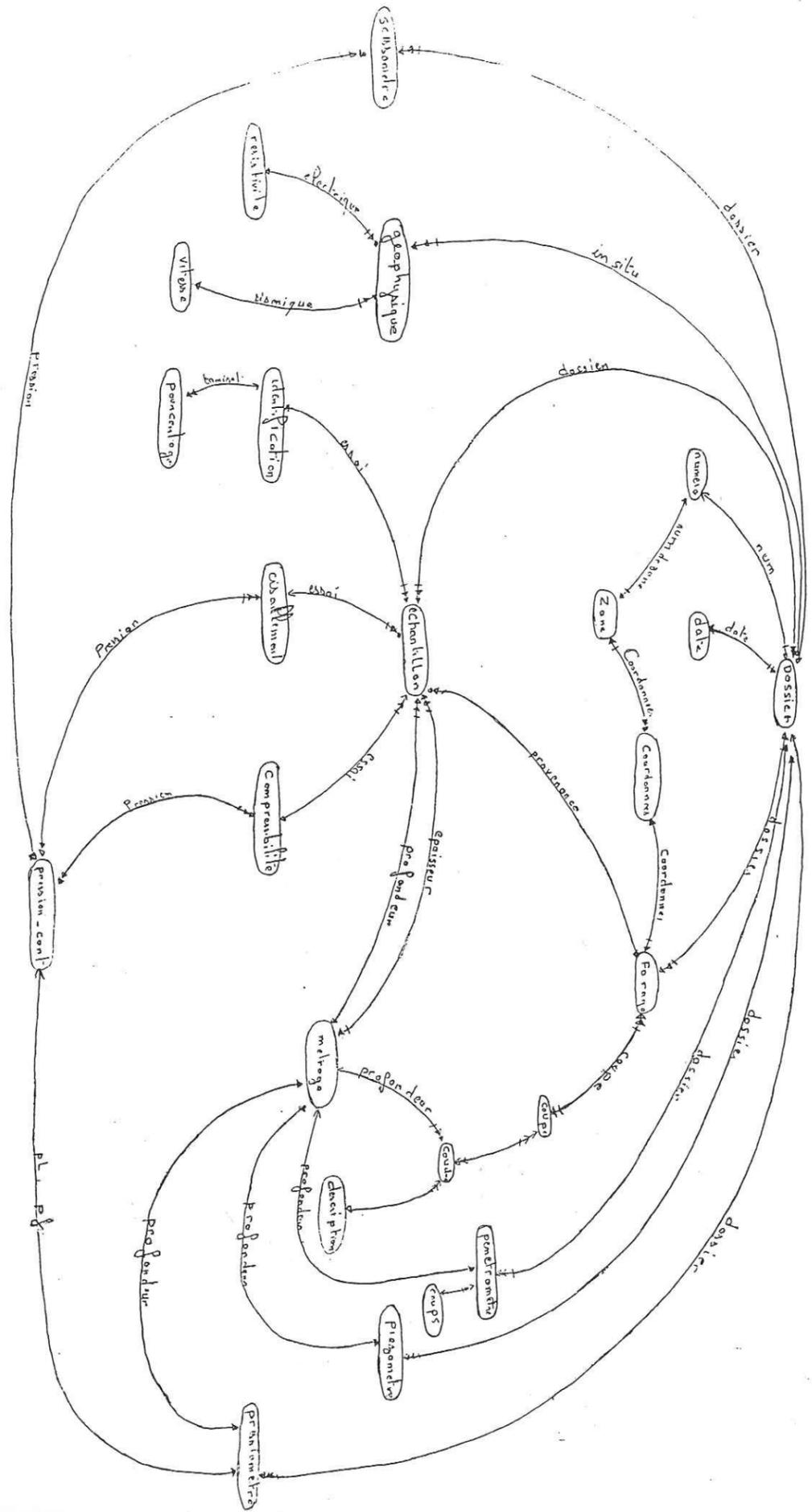
*scissomètre	dossier	
	<<-/-----/->	numéro
	num ds le dossier	
	<<-/-----/->	
	coordonnées	
	<<-/----->	coord-Lambert
	forage	
	<<-/-----/->	forage
nature		
<<-/-----/->	type-scissomètre	
Relevé		
<<-/----->>	(métrage*ri*rr)	

*sismique	dossier	
	<<-/-----/->	numéro
	num ds le dossier	
	<<-/-----/->	
	zone	
	<<-/----->	
	prem-tir	
	<<-/----->	coord-Lambert
deux-tir		
<<-/----->		
Relevé		
<<-/----->>	(métrage*vitesse)	

*résistivité	dossier	
	<<-/-----/->	numéro
	num ds le dossier	
	<<-/-----/->	
	zone	
	<<-/----->	
Relevé		
<<-/----->>	(métrage*vrésistivité* chaîne de caractères)	

c) LE SCHEMAS-Z GENERALISE

Le schéma-Z illustre sur un même document les associations qui existent entre les ensembles d'entités. Le schéma, présenté par la suite, a été simplifié pour des raisons de lisibilité



SCHEMA 5 - Z : simplifié

V. LA METHODE ORIENTEE OBJET : DOSSIER DE CONCEPTION

A. PRESENTATION DE LA METHODE ORIENTEE OBJET

C'est la phase finale de notre méthodologie appelée "méthodologie orienté objet". A partir d'une première liste de données suggérées par l'application de la méthode S.A.D.T(dans la première phase), et à partir des entités et les ensembles d'entités définis par le modèle entité-association (dans la deuxième phase), on applique la conception orientée objet.

Il s'agit d'énumérer et de définir les différents objets concrets ou informatiques qui entrent dans la composition du système.

Pour la définition des objets, nous avons suivi plutôt une démarche pragmatique. En effet nous les avons définis selon leur appartenance à l'une des catégories suivantes :

- Les objets concrets du monde réel,
- les objets définis par une structure de données
- les objets définis par une fonctionnalité ou par un ensemble d'opérations.

Chacun de ces objets doit être caractérisé par ses attributs et par un ensemble d'opérations prédéfinies appelées fonctions d'accès.

Ces fonctions d'accès définissent l'interface de l'objet, elles seules permettent de :

- Accéder aux attributs visibles de l'objet
- réaliser des opérations sur les objets

nous rappelons que la méthode orientée objet a été choisie pour:

-Permettre une meilleure indépendance entre les différentes parties du logiciel,

-assurer une meilleure capacité d'adaptation et de portabilité du logiciel.

Cette méthode nous permettra de définir une architecture logicielle basée sur la décomposition modulaire. Chaque objet définit un module dont l'interface est l'ensemble des fonctions d'accès de l'objet. C'est la partie accessible aux autres modules et qui permet la communication entre les différentes parties du système.

Dans ce chapitre on a va définir des objets graphiques, des objets de modélisation et les objets géologiques.

## B. SPECIFICATION DES OBJETS GRAPHIQUES

### 1. LES DEFINITIONS DE BASE

On présente ici quatre concepts généraux concernant la région, l'image, la transformation graphique et la structure hiérarchique de l'image. Chaque concept est traité comme une collection d'objets accessibles uniquement par certaines fonctions d'accès.

#### a) LA REGION

Une région correspond à une surface en deux dimensions ou un volume en trois dimensions. Les données comme fenêtre, clôture et volume normalisé de découpage (voir présentation des techniques graphiques) appartiennent au concept général de "région".

Une région est donc un ensemble de point dans un espace E. Exemple : Le carré unité à l'origine peut être considéré comme une région définie comme suit:

$$R_1 = \{(x_1, x_2) / 0 < x_1 < 1 \text{ et } 0 < x_2 < 1\}$$

La définition de la région comme ensemble permet d'utiliser les opérations sur les ensembles.

Les opérations importantes sur les régions sont les opérations utilisées sur les ensembles c'est à dire, l'intersection, l'union, la différence et le prédicat (déterminant si une région est vide ou si elle contient une autre région).

#### b) L'IMAGE

Une image est une fonction partielle  $I$  dont le domaine  $D$  est un sous-ensemble de l'espace choisi.

On peut définir de deux façons l'image correspondant au segment ayant pour origine le point (0,0) et pour extrémité le point (1,1).

La première définition (figure 1.a) est:

$$p_1 = \{(x_1, x_2) / 0 < x_1 < 1, 0 < x_2 < 1, x_1 = x_2\}$$

La deuxième définition (b) est  $p_2 = \{(x_1, x_2) / 0 < x_1 < 1, 0 < x_2 < 1 \text{ ou } f(x_1, x_2) = 1 \text{ si } x_1 = x_2 \text{ et } f(x_1, x_2) = 0 \text{ si } x_1 \neq x_2\}$

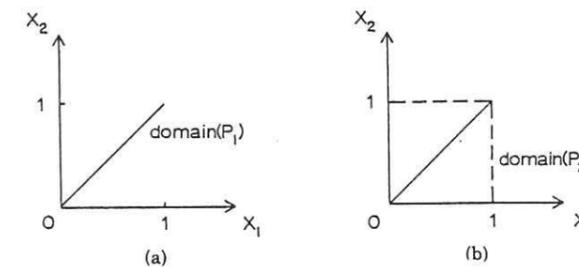


figure 4.1: définition de l'image correspondant à un segment (William 82)

On retient dans le cas (a) le fait que la forme de l'image est définie par le domaine.

On peut citer deux importantes opérations sur les images :

-La somme : C'est une opération qui traduit le recouvrement des images "overlying". La somme  $I_0 = I_1 + I_2 =$

$$\{(p, I_0(p)) / p \in D_1 \cup D_2\} \text{ avec}$$

$$I_0(q) = I_1(q) \text{ pour tout } q \in D_1 - D_2$$

$$I_0(q) = I_2(q) \text{ pour tout } q \in D_2 - D_1$$

$$I_0(q) = I_1(q) + I_2(q) \text{ pour tout } q \in D_1 \cap D_2$$

La couleur de l'image résultante de  $I_1 + I_2$  dans la région de recouvrement est déterminée par la somme des couleurs des deux images. De plus, on peut définir la somme des couleurs de plusieurs façons, par exemple:  $C_1 + C_2 = \text{maximum}(C_1, C_2)$ . Pour l'élimination des parties cachées la somme des deux images est définie selon d'autres critères.

-Domaine : La fonction  $\text{domaine}(I)$  retourne -comme résultat- la région correspondant au domaine de l'image  $I$

### c) LA TRANSFORMATION GRAPHIQUE

-DEFINITION : Une transformation graphique est une fonction qui transforme une image en une autre image. Les translations, changements d'échelles, rotations, projections et découpages (clipping) sont des exemples de transformations graphiques.

Les transformations graphiques sont des opérations sur les images certes, mais elles peuvent également être des opérations sur elles-mêmes. Il est donc justifié de traiter les transformations graphiques comme des objets.

-OPERATIONS :

-Application sur les images

-composition de deux transformations graphiques

$T_1 \circ T_2(I) = T_1(T_2(I))$  pour tout image  $I$ .

cependant, on note deux catégories de transformations graphiques:

*-Les transformations géométriques :*

Ce sont des transformations graphiques qui déplacent les points d'une image sans changer leur couleur.

Une transformation géométrique est donc une fonction entre des points  $G : E \rightarrow E$ . L'effet de la transformation géométrique  $T^g$  sur une image  $I$  est défini comme suit:

$$T^g(I) = \{ (G(p), I(p)) / p \in \text{domaine}(I) \}$$

$G$  est la fonction géométrique qui caractérise  $T^g$ . Une fonction géométrique  $G$  est appliquée à un point  $q$  par  $G(q)$  et à une région  $R$  par  $G(R) = \{ G(q) / q \in R \}$ .

*-Les transformations de restriction:*

Elles sélectionnent une partie de l'image. C'est le cas du découpage "clipping", qui rejette la partie de l'image située en dehors de la fenêtre ou de l'écran.

Une transformation de restriction est caractérisée par une région. Quand on applique la transformation de restriction sur une image, tous les points situés en dehors de la région sont éliminés.

L'effet de la transformation de restriction  $T^r$  sur une image  $I$  est défini comme suit :

$T^r(I) = \{ (p, I(p)) / p \in (R \cap \text{Domaine}(I)) \}$  avec  $R$ , la région caractéristique de  $T^r$ .

### d) LA STRUCTURE HIERARCHIQUE DE L'IMAGE

On définit une structure hiérarchique  $H$  comme un ensemble de noeuds qui peuvent être décomposés en une racine  $n$  et en une collection de sous-ensembles  $H_1, H_2, \dots, H_k$   $k \geq 0$ , tels que chaque  $H_i$  est une structure hiérarchique et tels que l'union de tous les  $H_i$  est  $H - \{n\}$ .  $n$  est le parent de chaque  $H_i$ , et chaque  $H_i$  est un descendant de  $n$ . Un noeud sans descendant est appelé feuille.

Quand les descendants  $H_i$  de chaque noeud  $n$  sont des ensembles disjoints, on parle de structure d'arbre; si en plus les  $H_i$  forment toujours une séquence d'ordre total, le résultat est un arbre d'ordre. La plus grande différence entre un arbre et une structure hiérarchique, est qu'aucun noeud de l'arbre ne peut avoir plus d'un parent.

On définit ainsi un arbre image , comme un arbre d'ordre dans lequel:

- Chaque noeud est associé à une image ,
- chacun des descendants du noeud (si il en a ) est associé à une transformation graphique .

Pour spécifier l'opération de l'affichage d'une image d'arbre H on définit l'image résultante de H comme:

$P+$  la somme des  $(T_i(p_i))$  ,  $i$  varie de 1 jusqu' à  $k$  , avec  $P$  est le noeud associé au noeud H,  $T_i$  est la transformation graphique associée avec le descendant  $H_i$  ,  $p_i$  est le résultat image de  $h_i$

## 2. LA SPECIFICATION DES OBJETS GRAPHIQUES

Chacun des concepts analysés précédemment permet de faire abstraction des données . On spécifie une collection d'objets auxquels on ne peut accéder qu'à travers un ensemble d'opérations "fonctions d'accès" bien défini pour chacun des objets.

Dans ce chapitre, on énumérera les objets graphiques manipulés par le système , on présentera leurs définitions et leurs fonctions d'accès

### *-POINT 2D :*

Définition : C'est un couple de réel

Fonction d'accès :

Point2d (réel,réel)-----> point2d :c'est  
la fonction génératrice d'un point du plan.

### *-POINT 3D :*

Définition : C'est un triplet de réel .

Fonction d'accès :

Point3d (réel,réel,réel)-----> point3d :c'est  
la fonction génératrice d'un point de l'espace.

### *-REGION :*

Définition: C'est un ensemble de points .

Fonctions d'accès:

Région vide ----->région : c'est  
la fonction génératrice de la région vide.

Ecran----->région :c'est  
la fonction génératrice de la région écran.

Segment2d(point2d,point2d)----->région:crée  
l'ensemble des points du segment dans le plan.

Segment3d(point3d,point3d)----->région : crée  
l'ensemble des points du segment dans l'espace.

Rectangle (point2,point2d)----->région: crée la  
région rectangle définie par les extrémités d'une diagonale.

Contour(ensemble de point)----->région:crée  
l'ensemble des segments qui forment le contour.

Tache (ensemble de points)----->région : crée  
l'ensemble des points contenus dans le contour.

### *-FCTG2D :*

Définition : C'est une transformation géométrique dans  
le plan.

Attribut : Une matrice carrée d'ordre 3.

Fonctions d'accès :

Echelle2d (réel, réel) ----->fctg2d : crée la matrice carrée correspondant à la transformation de changement d'échelle.

Rotation (réel)----->fctg2d : crée la matrice carrée correspondant à la rotation d'un angle défini par le réel.

Translation(réel,réel) ----->fctg2d : crée la matrice carrée correspondant à la translation dont les déplacements sont définis par les deux paramètres réels.

Compose(fctg2d,fctg2d)----->fctg2d : produit la transformation, résultat de la composition de deux transformations.

Prisevue2D(région,région,fctg2d)-->fctg2d : produit la transformation de visualisation dans le plan à partir de la région fenêtre ,de la région clôture et d'une transformation fctg2d

*-FCTG3D :*

Définition : C'est une transformation géométrique dans l'espace .

Attribut : Une matrice carrée d'ordre 4.

Fonctions d'accès:

Translat3d(réel,réel,réel)----->fctg3d :Crée la matrice carrée d'ordre 4 correspondant à la translation dans l'espace.

Echelle3d(réel,réel,réel)----->fctg3d : crée la matrice carrée d'ordre 4 correspondant au changement d'échelle dans l'espace.

Rotation3d,x (réel)----->fctg3d : crée la matrice carrée d'ordre 4 correspondant à la rotation selon l'axe des X.

Rotation3d,y (réel)----->fctg3d : crée la matrice carrée d'ordre 4 correspondant à la rotation selon l'axe des Y.

Rotation3d,z (réel)----->fctg3d : crée la matrice carrée d'ordre 4 correspondant à la rotation selon l'axe des Z.

Prisevue3d(point3d, point3d,réel ,réel, région )  
----->fctg3d : crée la matrice correspondante à la projection perspective définie par les paramètres suivants :

la position de l'observateur (point3d ),  
le point de visé (point3d),  
l'angle d'ouverture (réel),  
l'angle de rotation selon l'axe de visé (réel) et  
la fenêtre (région).

Vuedeface----->fctg3d : crée la matrice carrée d'ordre quatre correspondant à la projection orthogonale selon l'axe des x.

Vuedecoté----->fctg3d : crée la matrice carrée d'ordre quatre correspondant à la projection orthogonale selon l'axe des y.

Vuededessus----->fctg3d : crée la matrice carré d'ordre quatre correspondant à la projection orthogonale selon l'axe des z.

Compose3d(fctg3d,fctg3d) ----->fctg3d : détermine la matrice résultat de la composition de deux transformations géométriques en trois dimensions;

*-DECOUPAGE2D:*

Définition : C'est une transformation de restriction dans le plan.

Attributs : fenêtre : une région  
segment : une région

Fonction d'accès :

Découpage2d(région,région)----->région,logique

Cette fonction découpe la première région (un segment) par la deuxième (fenêtre) et produit une région découpée quand la valeur du paramètre logique est égale à vrai

### *-DECOUPAGE3D*

Définition : C'est une transformation de restriction dans l'espace

Attributs : volume : région ,  
segment3d: région.

Fonctions d'accès:

Découpage3d(segment3d,volume)->segment,logique

cette fonction découpe le segment3d en fonction du volume de visualisation , le segment produit est la partie visible si le logique = vrai

### *-ELIMINE CACHEE*

Définition : C'est une transformation de restriction qui pour un objet tridimensionnelle définit les faces qui sont cachées par d'autres . Le critère de visibilité des faces peut varier selon le point de vue.

Attribut : Un ensemble de faces

Fonction d'accès:

Elimine cachée (ensemble de faces)-->ensemble de

faces

### *-IMAGE CONTINUE*

Définition : C'est un ensemble de points sur l'écran

Fonctions d'accès :

Affpoint(point2d,couleur)----->image continue

Lineseg (point2d,point2d,couleur)-->image  
continue

Text(string,point2d,couleur)---->image continue

## C. SPECIFICATION DES OBJETS DE MODELISATION

### 1. UN DOMAINE

Définition : On définit un domaine polygonal D par:

-un ensemble de points extérieurs Vext

-un ensemble d'arêtes extérieures Eext

-et un ensemble des points intérieurs Vint

(Vext,Eext) définit le contour du domaine D

On pose  $V = Vext + Vint$ , V est l'ensemble des points du domaine D

### 2. LA TRIANGULATION

Définition : Une triangulation T d'un domaine D est un ensemble de triangles  $T_i$  qui doivent vérifier les conditions suivantes :

-Les sommets de  $T_i$  sont des éléments de V

-La somme des triangles  $T_i$  appartenant à  $T = D$

-L'intersection de deux éléments différents  $T_i$  et  $T_j$  doit être

-vide

-ou réduite à un point

-ou égale à une arête

Fonction d'accès :

DéfinirT(D:contour, Vint)----->Triangulation

### 3. LE MAILLAGE

Définition : Soit  $X$  la matrice des coordonnées des sommets de  $V$  alors  $M = (D, V, E_{ext}, T, X)$  définit un maillage de topologie  $T$  sur le domaine  $D$

Cependant, on peut définir un maillage selon une topologie autre qu'une triangulation

### 4. L'ENVELOPPE CONVEXE

Définition: L'enveloppe convexe d'un domaine  $D$  - dont l'ensemble des points est  $V$  - est le polygone convexe qui englobe l'ensemble des points  $V$

Attributs:

$V$  : un ensemble de points définissant un domaine

Fonctions d'accès :

$-ext-int(V) \rightarrow V_{int}, V_{ext}$  : cette fonction répartit l'ensemble des points  $V$  en deux ensembles, les points externes et les points internes. L'ensemble des points externes ordonnés définit un polygone convexe, c'est l'enveloppe convexe du domaine  $D$ .

$-Convex\ hull(V) \rightarrow$  enveloppe convexe  
cette fonction donne l'enveloppe convexe

### 5. LA COURBE

Définition : Une courbe est un ensemble de polygones ouverts ou fermés, reliant des points ayant la même valeur  $Z$  de la propriété étudiée ; c'est à dire la représentation graphique de la fonction  $F(x,y) = \text{constante}$

Attributs:

$T$ : Une triangulation du domaine

$Z_0$ : la valeur constante de la courbe

Fonction d'accès:

courbe-iso ( $T, Z_0$ )  $\rightarrow$  courbe

### 6. LA CARTE ISO

Définition : C'est un ensemble de courbes d'isovaleur, chaque courbe a une valeur constante de la propriété cartographiée.

Attributs:

$T$  : Une triangulation du domaine  $D$

$[Z_i \dots Z_j]$  : Le domaine des valeurs de la propriété à cartographier.

$Pas$  : la différence entre les valeurs de deux courbes successives.

Fonction d'accès:

Carte-isoval( $T, Z_i, Z_j, pas$ )  $\rightarrow$  carte-iso

## **D. SPECIFICATION DES OBJETS GEOLOGIQUES**

### 1. LA COUCHE

Définition : Une couche est définie par sa borne inférieure (profondeur du mur), par sa borne supérieure (profondeur du toit) et par sa nature

Attributs :

Limites : un couple de réels indiquant la profondeur du toit et la profondeur du mur de la couche.

Nature : la nature de la couche peut être indiquée d'une façon composée :

- terme de définition : il définit la lithologie
- terme de description : il décrit une composante
- niveau de description : il indique s'il s'agit de la composante principale, secondaire,...
- terme de coordination : il décrit la nature des relations entre composantes (inclusion, juxtaposition, alternance)

Qualificatif : indique s'il s'agit d'un bedrock, d'une couche compressible, ou de la contenance d'un minéral précis.

Fonction d'accès:

-saisie-couche(limites,nature,qualificatif)  
----->couche:            cette  
fonction définit la couche

-nature-couche(couche)----->nature    :   cette  
fonction récupère la nature de la couche.

-limite-couche(couche)----->limites    :   cette  
fonction récupère les bornes de la couche.

## 2. LE FORAGE

Définition : C'est un ensemble de couches ordonnées selon la profondeur

Attributs :

coord : les coordonnées du forage  
topo : l'altitude du terrain au niveau du forage  
succ-couche : la succession des couches selon la  
profondeur

Fonction d'accès :

Saisir-forage----->forage: c'est la fonction génératrice du forage.

long(F:forage)----->réel : cette fonction renvoie la longueur (profondeur) du forage.

Prof-c(f:forage, n :nature de couche)-->un ensemble de réels : cette fonction renvoie les profondeurs auxquelles on a rencontré la couche de nature n.

épai-c(f,n)----->réel : cette fonction renvoie l'épaisseur cumulée d'une couche.

Bedrock(f)----->logique : donne la valeur logique vrai si le forage est arrêté par le substratum

compressible(f)----->liste des couches compressibles

litho(f,prof:réel)----->nature : cette fonction renvoie la nature de la couche située à une profondeur déterminée.

## 3. LE TERRAIN

Définition: Un terrain est un ensemble de forages situés dans une zone délimitée

Attributs:

F: l'ensemble des sondages  
Zone : définie par les coordonnées de ses sommets

Fonction d'accès:

Domaintopo(F)-----> un ensemble de couples (point2d, réel) : cette fonction définit le graphe de la fonction topographie (p:un point, topo (p): réel) .

Domainlitho-sup(F)-----> un ensemble de couple (point2d, nature de couche) : cette fonction définit le graphe de la

fonction litho (p:un point3d , litho(p): nature)) indiquant la nature du terrain affleurant.

Domainlitho-prof(F,Z:une profondeur)-----> un ensemble de couples (point3d, nature de couche) : les points sont pris à une profondeur z . La fonction permet de récupérer la nature lithologique dans chacun des points

Domainprof-couche(F,N: nature de couche)--> un ensemble de couples (point2d, réel) .cette fonction définit le graphe de la fonction profondeur de la couche de nature N . Les point2d correspondent aux coordonnées des forages , le réel est la profondeur à laquelle on a rencontré la couche de nature N

Dom-épai-com-sup(F,p:profondeur)-----> un ensemble de couples (point2d,réel) : cette fonction donne pour l'ensemble des forages, l'épaisseur des couches compressibles supérieures cumulée sur une profondeur p . C'est donc le graphe de la fonction épaisseur des couches compressibles supérieures.

#### 4. LA REPARTITION LITHOLOGIQUE

Définition : Une répartition lithologique est un ensemble de facettes planes dont chacune appartient à une même couche géologique . Ces facettes présentent une répartition de la surface du terrain étudiée et correspondant au domaine D

Attributs:

-V : l'ensemble des points 3d du domaine indiquants la lithologie.

-T : Une triangulation du domaine

-F: Un ensemble de facettes planes dont chacune appartient à la même couche.

Fonctions d'accès :

-répartition litho(V,T)-----> répartition lithologique

-carte-litho(F) ----->image continue cette fonction produit la carte lithologique à partir de l'ensemble des facettes ,c'est une projection parallèle (vue de dessus du domaine)

-litho3d (F) ----->image continue c'est une projection perspective utilisant l'élimination des parties cachées des facettes

#### 5. UN PROFIL GEOLOGIQUE

Définition : C'est une surface passant par deux ou plusieurs forages géologiques ( les reliant entre eux ) . La surface correspondant à un profil géologique peut être considérée comme un ensemble de facettes planes, définissant les limites des couches.

Attributs:

P: plan de la coupe .

F: l'ensemble des forages ordonnés

Plaquettes :l'ensemble des facettes plane de la coupe . Chacune des facettes appartient à une même couche . Cet ensemble de facettes définit une partition de la surface reliant les forages entre eux.

Fonction d'accès:

Profilgéo(F) ----->plaquettes

#### 6. UN BLOC-DIAGRAMME

Définition : Un bloc-diagramme représente un volume du sous-sol, délimité par un ensemble de profils géologiques correspondant aux forages de contours, et par l'ensemble des facettes

lithologiques supérieures (la répartition lithologique du domaine) . Le bloc-diagramme est donc un volume délimité par ses facettes extérieures.

#### Attributs :

Epg : Un ensemble de profils géologiques

Rep-lith : Une répartition lithologique

#### Fonctions d'accès:

bloc(rep-lith, epg) : bloc-diagramme.

affiche-bloc(bloc-diagramme)---->image continue

### E. CONCLUSION

Dans notre cas , nous avons donné la priorité à la réalisation du sous-logiciel graphique . Nous n'allons donc pas "implanter" tout les objets définis dans ce chapitre , seuls les objets de modélisation et les objets graphiques ont été "implantés" . Sur la base de ces objets , nous avons construit la bibliothèque graphique . Elle est formée par un ensemble d'unités permettant la modélisation géométrique et la visualisation en deux et en trois dimensions avec l'élimination des parties cachées . Certains objets ont été simplement sujet de développement algorithmique, surtout en ce qui concerne le maillage, la triangulation et la cartographie.

Cependant , avec l'application du modèle entité-association et la conception orientée objets , nous avons contribué à la formalisation des données géologiques et géotechniques . Cette formalisation pourrait servir à un travail ultérieur pour construire un système plus intelligent: un système expert ou un système de base de connaissances

Nous avons utilisé pour l'implémentation, le langage Turbo Pascal. Ce langage présente un caractère typé . Il permet aussi de construire des modules , de définir leurs interfaces (fonction d'accès , types et variables visibles) et de les compiler séparément . Cela nous a permis d'associer un module à chaque objet défini

Mais l'utilisation de ce langage présente aussi certaines contraintes dans le contexte orienté objet . En effet, si ce langage permet la construction de modules ,il reste un langage procédural et ne présente pas les avantages que présente un langage orienté objet proprement dit tel que le "Smalltalk" .

Les langages orientés objets permettent par de simples instructions la création d'un objet , la définition de ces attributs et des opérations qui manipulent cet objets. Ces langages présentent donc l'avantage d'être utilisés à partir de de la phase de conception et tout au long du cycle de vie d'un logiciel.

En utilisant un langage orienté objet , le logiciel produit est une base de connaissances formée par l'ensemble des objets du système . Chaque objet représente à la fois un savoir déclaratif (base de données locale) et un savoir procédural (les méthodes) . La connaissance (données et programmes) est distribuée entre chacun des objets .

Par de simples instructions les langages orientés objets permettent aussi de créer des classes et des groupes d'objets. Ils permettent aussi l'instanciation des objets : En effet plusieurs objets de même nature sont regroupés en une classe . La classe est un moule à partir duquel on fabrique autant d'exemplaires que l'on veut [Bailly87] . Ces exemplaires sont appelés instances ou individus , ils peuvent être des objets sous-classes et ils héritent alors les propriétés de la classe mère .

La propriété d'héritage est très importante . Nous donnons l'exemple suivant dans le cas de la géologie: si on définit une faille d'une part par une direction , et si d'autre part on définit une faille normale comme une sous-classe de l'objet faille , il sera inutile de mentionner qu'une faille normale est caractérisée par une direction . Cela nous évite de nombreuses redondances .

### VI. LA REALISATION DES PROCEDURES DE VISUALISATION

Les chapitres précédents ont été consacrés à l'analyse et la conception du système de stockage ,de traitement et de visualisation des données géologiques et géotechniques . La priorité a été donnée à la

réalisation du sous logiciel graphique sous forme de bibliothèques formées par un ensemble de modules de modélisation géométrique et de visualisation en deux et trois dimensions, avec l'élimination des parties cachées. Ces modules ont été réalisés sur la base de certains objets spécifiés dans la première partie de notre travail. La présentation de ces modules, leurs structures et leurs interfaces ne seront présentés que lors du chapitre suivant : Le dossier de réalisation de la bibliothèque graphique.

On ne présente dans ce chapitre que les algorithmes des procédures de visualisation exécutées dans certains modules, nous détaillons tout particulièrement celles qui concernent les objets tridimensionnels : la projection perspective générale, le découpage tridimensionnel et l'élimination des parties cachées. Pour chacun des problèmes, nous avons essayé de choisir l'algorithme adéquat, de justifier le choix, de le simplifier ou l'adapter à notre cas précis. L'implantation de ces procédures repose les structures de données choisies pour la modélisation géométriques des objets, notamment les volumes (bloc-diagrammes). Ce travail de modélisation est exposé dans le chapitre "modélisation géométriques des données".

Pour la réalisation des procédures de visualisations, la connaissance des techniques graphiques de base est nécessaire. Nous avons donc jugé utile de les rappeler. Nous les avons présentées dans l'annexe A. On y a détaillé les transformations graphiques, la représentation matricielle homogène, le fenêtrage, les transformations de visualisations et la classification des projections.

## A. LA REALISATION DE LA PROJECTION PERSPECTIVE GENERALE

### 1. LE PRINCIPE DE L'ALGORITHME

La situation où la direction de vision est confondue avec l'axe z est un cas particulier. Mais il faut savoir déterminer la matrice de la projection perspective dans le cas général c'est à dire quand la direction de vision n'est pas confondue avec l'axe Z. Cela peut être effectué en ramenant le cas général au cas particulier par un changement de repère

On a déjà modélisé dans le repère observateur la perspective (selon l'axe z) par la matrice homogène  $M_{per,z}$  établie dans l'annexe A.

On peut maintenant modéliser la perspective générale dans le repère objet par une matrice  $M_{per,O}$  qu'on obtient par composition du changement de repère (passage du repère objet au repère observateur) et de la perspective d'axe z.

On va établir dans la section suivante La matrice  $M^{-1}$  correspondant au passage du repère objet au repère observateur

Les points d'un objet sont donnés par leurs coordonnées dans le repère objet ; ces coordonnées doivent être exprimées dans le repère de l'observateur pour pouvoir appliquer la formule de la perspective. On a alors :  $M_{per,O} = M_{per,z} * M^{-1}$

Si on s'intéresse à une projection perspective plutôt qu'à une transformation perspective on aura :

$$M_{proj,O} = M_{proj,z} * M^{-1}$$

La perspective est donc une transformation de l'espace 3D dans l'espace 3D, cette transformation doit être suivie d'une projection pour former la projection perspective.

## 2. LE PASSAGE DU REPERE OBJET AU REPERE OBSERVATEUR

Dans ce paragraphe on se livre à un exercice dont le résultat sera surtout utile pour la visualisation en perspective des objets tridimensionnels (voir figure 6.1). Il s'agit d'élaborer la matrice qui permet d'exprimer les coordonnées d'un point dans un repère 2 (repère associé à l'observateur) à partir de ses coordonnées exprimées dans un repère 1 (repère associé à l'objet). Le repère 2 est caractérisé par un axe des  $x$  horizontal et un axe des  $z$  dirigé suivant un vecteur  $ov$  (donné dans le repère 1)

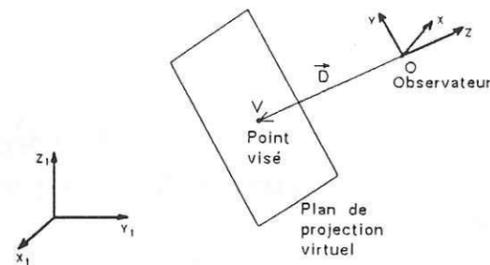


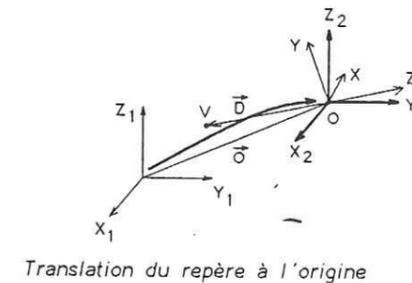
Figure 6.1: repères associés à l'observateur et à l'objet (Schweizer 87)

Le passage du repère 1 au repère 2 se fait par une succession de transformations élémentaires qui sont les transformations de coordonnées.

Les changements de repères élémentaires à effectuer sont dans l'ordre :

### \* La première étape : TRANSLATION

Cette étape (figure 6.2) consiste à translater le repère 1 ( $o_1 x_1 y_1 z_1$ ) à la position du point  $o$  et donne le repère intermédiaire ( $o x_2 y_2 z_2$ )



Translation du repère à l'origine

Figure 6.2 (Schweizer 87)

Cette translation est  $T = o_1 o$ . La matrice de changement de repère sera donc  $T^{-1}$ , l'inverse de la matrice de translation du vecteur  $o_1 o$ .

$$T^{-1} = \begin{vmatrix} 1 & 0 & 0 & -Tx \\ 0 & 1 & 0 & -Ty \\ 0 & 0 & 1 & -Tz \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

### \* La deuxième étape : ROTATION AUTOUR DE Z

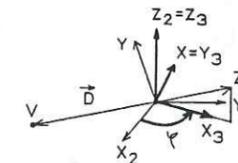


Figure 6.3 (Schweizer 87)

Il s'agit de tourner le repère intermédiaire ( $o x_2 y_2 z_2$ ) autour de  $z_2$  pour inclure le futur axe  $x_3$  dans le plan défini par  $z_2$  et  $ov$  (voir figure 6). Les projections de  $ov$  et de  $x_3$  sur le plan horizontal sont colinéaires. On obtient ainsi un nouveau repère intermédiaire ( $o x_3 y_3 z_3$ ). L'angle  $\theta$  est compté positivement de l'axe  $x$  vers l'axe  $y$ .

Le changement de repère est donné par la matrice  $R^{-1}_{z,\emptyset} =$

$$\begin{vmatrix} \cos \emptyset & -\sin \emptyset & 0 & 0 \\ \sin \emptyset & \cos \emptyset & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Dans le plan vertical défini par  $(z_2, ov)$ , l'axe  $x_3$  fait un angle  $\emptyset$  avec la droite de vecteur directeur  $ov$ . Cela nous permet de définir l'étape suivante.

*\* Troisième étape : ROTATION AROUND DE Y*

Ce troisième changement de repère (voir figure 6.4) a pour but de transformer  $x_3$  en  $x_4$ , tel que  $x_4$  soit colinéaire avec  $ov$ . Comme  $x_3$  fait un angle  $\emptyset$  avec la direction de  $ov$ ,  $\emptyset$  est compté positivement de l'axe  $z$  vers l'axe  $x$ . Ici, on veut ramener le futur  $x_4$  dans la direction de  $z$ , c'est une rotation de  $-\emptyset$ . Comme il s'agit d'un changement de repère, la matrice est  $R^{-1}_{y,\emptyset} =$

$$\begin{vmatrix} \cos \emptyset & 0 & \sin \emptyset & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \emptyset & 0 & \cos \emptyset & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

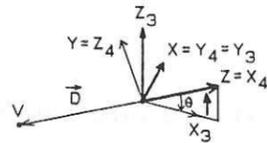


Figure 6.4

Ce repère intermédiaire  $(o x_4 y_4 z_4)$  est peu différent du repère final  $(o x y z)$ , en effet le passage au repère final est assuré par une permutation : c'est la dernière étape

*\* La quatrième étape*

IL s'agit d'échanger les coordonnées de telle façon que l'axe  $x$  soit horizontal et l'axe  $z$  soit orienté selon la direction de vision. La permutation est :

$$x = y_4$$

$$y = z_4$$

$$z = x_4$$

La matrice correspondante est  $E^{-1} =$

$$\begin{vmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

La matrice du changement de repère global  $M^{-1}$  est déterminée par la composition des quatre matrices

$$M^{-1} = E^{-1} * R^{-1}_{y,\emptyset} * R^{-1}_{z,\emptyset} * T^{-1} =$$

$$\begin{vmatrix} -\sin \emptyset & \cos \emptyset & 0 & -Tx \\ -\sin \emptyset * \cos \emptyset & -\sin \emptyset \sin \emptyset & \cos \emptyset & -Ty \\ \cos \emptyset * \cos \emptyset & \cos \emptyset * \sin \emptyset & \sin \emptyset & -Tz \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Le sinus et le cosinus des angles  $\emptyset$  et  $\emptyset$  peuvent être obtenus directement à partir du vecteur directeur  $D = ov$ , de sa projection  $D_{z=0}$  sur le plan  $z = 0$  et de leurs normes :

$$\cos \emptyset = D_x / D_{z=0}$$

$$\sin \emptyset = D_y / D_{z=0}$$

$$\cos \emptyset = D_{z=0} / D$$

$$\sin \emptyset = D_z / D$$

### 3. LA REALISATION DE L'ALGORITHME

L'algorithme de la projection perspective est réalisé sous forme de deux procédures:

*-LA PROCEDURE PRISEVUE3D : (unit transformation)*

*LES DONNEES:* Cette procédure a pour données les paramètres suivant:

-Une caméra : ce paramètre comporte la position de l'observateur et du point de visé, exprimé en coordonnées objet.

-Un angle d'ouverture définissant un facteur d'agrandissement .

-Un angle de pivotement introduisant une rotation selon la direction de vision et permettant de réaliser des effets de roulis en animation

-une fenêtre : Spécifie la partie de l'écran qui correspond à la projection du volume de visualisation

-une profondeur de vision : Avec la fenêtre, ce paramètre va être utile pour la définition des coordonnées écran normalisées 3d

*L'ALGORITHME :*

L'algorithme de la procédure prisevue3d effectue les étapes suivantes:

-Etape 1: Mettre à jour la fenêtre courante à partir de la fenêtre spécifiée dans les paramètres de la procédure.

-Etape 2: Calculer la matrice de la perspective générale qui correspond à la matrice de changement de repère permettant de transformer les coordonnées objet en coordonnées observateur . C'est la matrice  $M^{-1}$ .

-Etape 3: Calculer une deuxième matrice de perspective normalisée correspondant à un deuxième changement de repère permettant de transformer les coordonnées observateur en coordonnées écran normalisée .

-Etape 4: Effectuer la composition des deux matrices : cette opération va produire la matrice finale de la transformation perspective normalisée . Cette matrice ne comporte pas de projection , elle permet simplement le passage des coordonnées objet au coordonnées écran normalisées homogènes

-Etape 5: Mettre à jour la variable globale de la transformation courante 3d en lui affectant la valeur de la matrice de transformation perspective normalisée, calculée à l'étape 4.

*-LA PROCEDURE TSNH3DE (unit point)*

On n'appelle cette procédure que lorsque le point exprimé en coordonnées écran normalisées homogènes appartient au volume de visualisation normalisé

Cette procédure calcule les coordonnées écran d'un point, à partir de ses coordonnées écran normalisées homogènes , on passe ainsi d'un repère 3d à un repère 2d . C'est cette procédure qui effectue la projection . Pour le calcul des coordonnées x et y , exprimées dans le repère écran, on utilise que les coordonnées x et y , exprimées dans le repère écran normalisé homogène , on ne tient pas compte des z . La coordonnée z est conservée pour l'élimination des parties cachées .

### 4. INFLUENCE DE LA POSITION DE L'OBSERVATEUR

La figure 6.5 illustre la projection perspective d'un d'un volume - avec élimination des parties cachées- correspondant à un bloc-diagramme modélisant un terrain formé de couches stratiformes inclinées

C'est une perspective correspondant à un point de vue haut-face-droit:

observateur(40,35,30 )

point de visé(25,25,15)

angle d'ouverture = 60

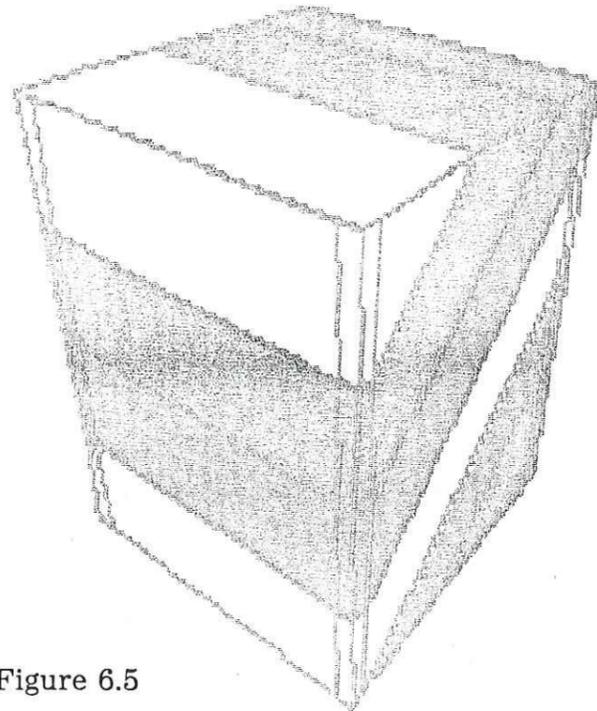


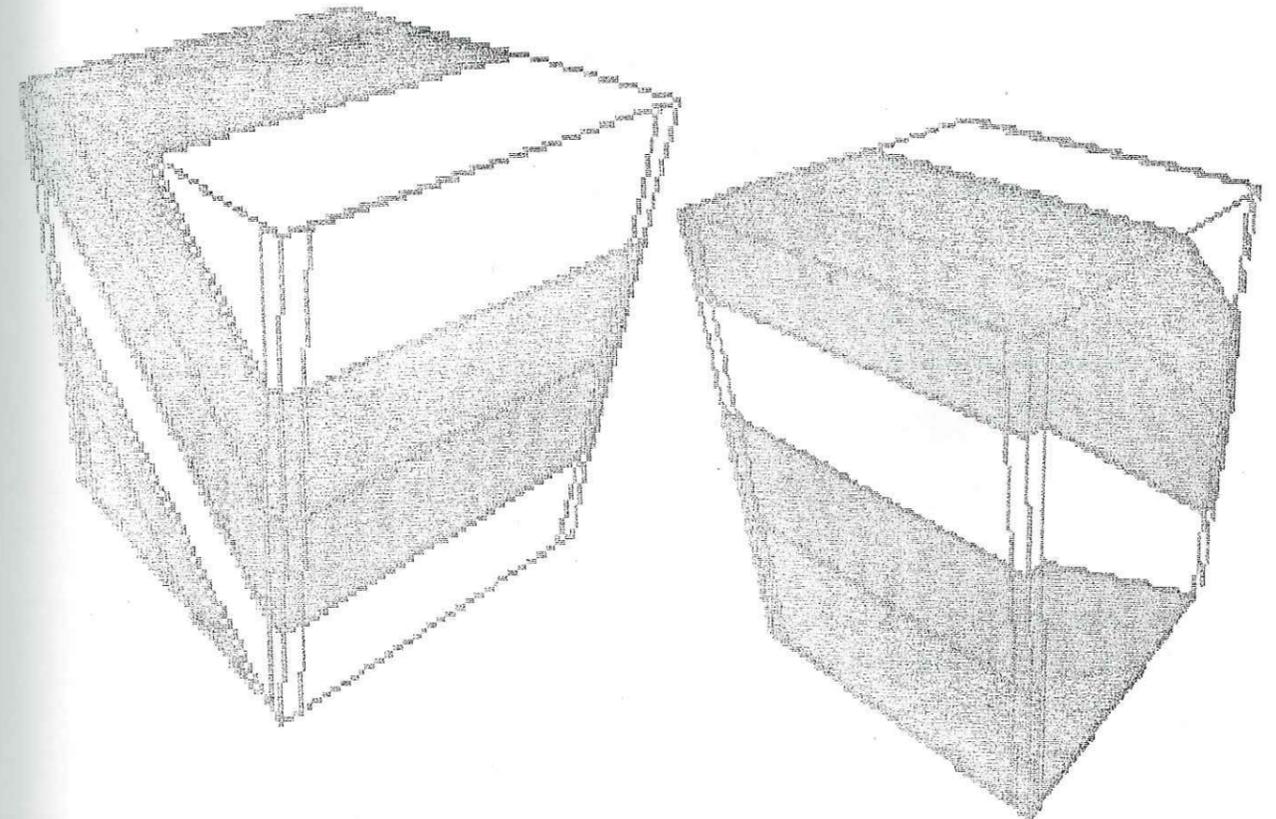
Figure 6.5

L'utilisateur pourra choisir la position de l'observateur et du point de visé, dans le cadre de la définition de la caméra (paramètre de la procédure prisevue3d). Cependant, on peut définir à priori quatre points de vue de perspective pour chaque objet tridimensionnel : haut-face-droit, haut-face-gauche, haut-derrrière-droit, haut-derrrière-gauche. Chacun de ces quatre points de vue correspond à une direction de visée, proche de la trisectrice du quart de l'espace correspondant

Les figures suivantes (6.6,6.7,6.8) représentent le même modèle de terrain selon des points de vue différents:

La figure 6.6 représente un point de vue haut-face-gauche: Observateur (40,-15,30), point de visé (25,-5,15), ouverture=60.

La Figure 6.7 représente un point de vue haut-derrrière-droit: Observateur (-20,-15,30), point de visé (5,-5,15), ouverture = 60.

Figure 6.6: perspective  
haut-derrrière-droiteFigure 6.7: perspective  
haut-face-gauche

La figure 6.8 représente un point de vue haut-derrière-gauche:  
 Observateur  $(-20, 35, 30)$ ,  
 point de visé  $(-5, 25, 15)$ , ouverture = 60

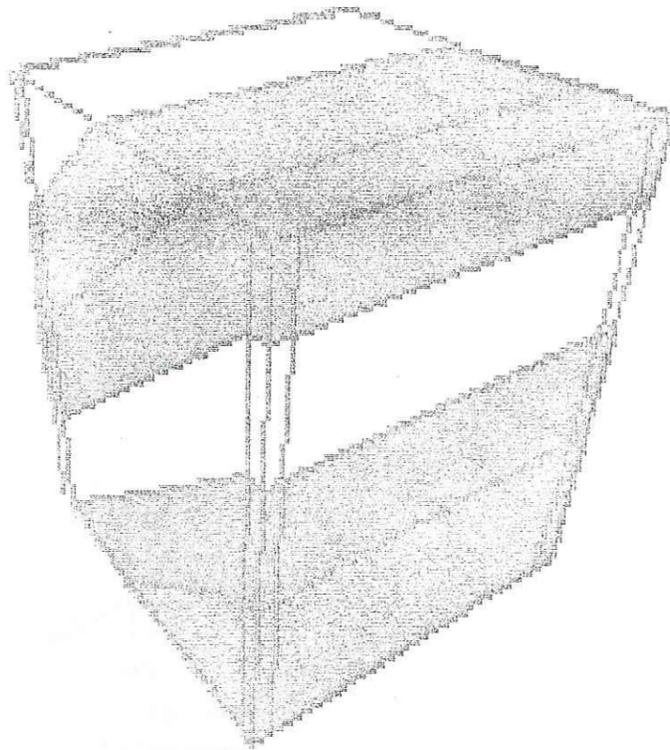


Figure 6.8

## B. LA REALISATION DES PROJECTIONS ORTHOGONALES

La réalisation de ces projections est plus facile que celle des projections perspectives généralisées. Il s'agit des projections parallèles à l'un des axes principaux caractérisées par le fait que l'observateur est placé à l'infini sur l'axe de projection. Ces projections sont définies par une transformation de repère et la suppression d'une coordonnée. Les matrices correspondantes ont été présentées dans l'annexe A.

Nous avons choisi de réaliser indépendamment les transformations orthogonales correspondant aux changements de repères, et les projections correspondantes à la suppression de la coordonnée  $z$ .

Les transformations orthogonales sont réalisées dans le module transformation (unit transformation). Il s'agit des procédures suivantes:

- PROCEDURE VUEDEFACE,
- PROCEDURE VUEDEDERRIERE,
- PROCEDURE VUEDEDROITE,
- PROCEDURE VUEDEGAUCHE,
- PROCEDURE VUEDEDESSOUS.

Toutes ces procédures définissent la matrice du changement de repère; celui-ci est effectué par une simple permutation des axes, de façon que les axes des  $x$  et des  $y$  soient dans le plan de projection. L'axe des  $z$  du nouveau repère est parallèle à la direction de la projection, et dirigé de telle manière que l'observateur soit placé à  $z = -$ . Ce choix a été effectué pour l'élimination des parties cachées; la coordonnée  $z$  sera plus importante quand le point est plus loin de l'observateur. Cette démarche nous a permis d'utiliser les mêmes procédures pour l'élimination des parties cachées à la fois dans le cas de la projection perspective et dans celui de la projection parallèle.

La coordonnée  $z$  ne sera pas utile pour déterminer les coordonnées écran du point. Elle n'a qu'une valeur relative permettant de décider de l'élimination des surfaces cachées.

Pour l'affichage du point on n'a besoin que des coordonnées  $x, y$ , obtenues après l'application de la matrice de la transformation

orthogonale correspondante . La projection consiste simplement en la suppression de la coordonnée z.

la détermination des coordonnées x,y dans le repère écran se fait après la projection moyennant une transformation de visualisation dans le plan (procédure prisevue2d:unit transformation ). Cette transformation 2d est définie en spécifiant une clôture dans le plan de projection et une fenêtre dans l'écran ; elle fait correspondre à chaque point du plan de projection, un point à l'écran .

Les figures suivantes représentent les projections orthogonales du même modèle de terrain(couches stratiformes inclinées) que celui représenté , en projection perspective, dans le paragraphe précédent .

La figure 6.9 représente la vue de face : c'est une projection selon l'axe des x.

La figure 6.10 représente la vue de derrière : c'est une projection selon l'axe des x avec inversion du sens de l'axe.

La figure 6.11 représente la vue de droite : c'est une projection selon l'axe des y.

La figure 6.12 représente la vue de gauche : c'est une projection selon l'axe des y avec inversion du sens de l'axe .

La figure 6.13 représente la vue de dessus : c'est une projection selon l'axe des z.

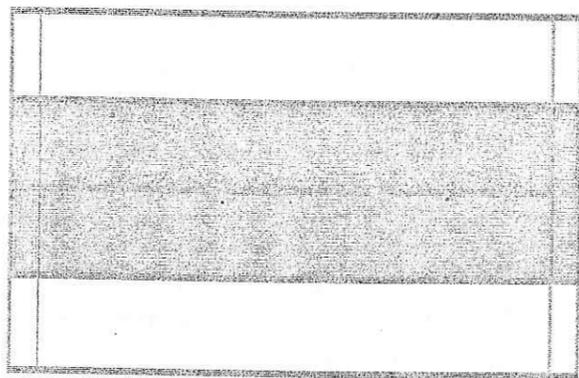


Figure 6.9:vue de face



Figure 6.10 : vue de derrière

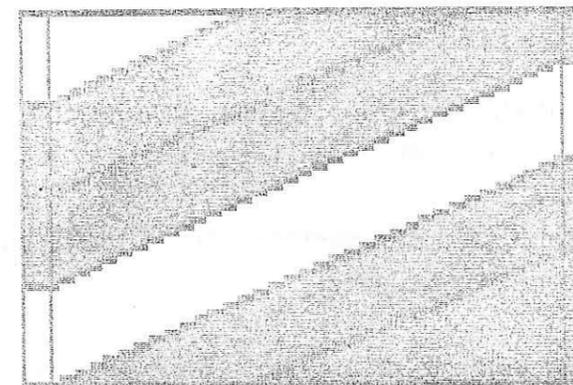


Figure 6.11:vue de droite

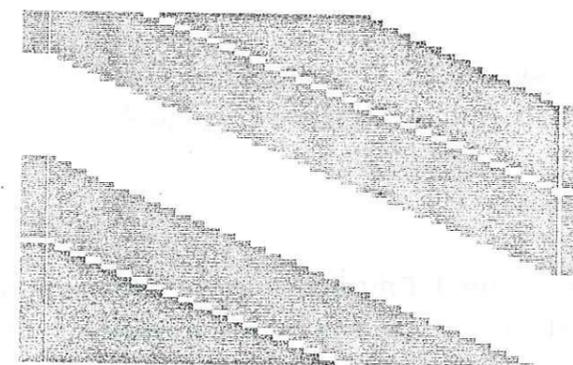


Figure 6.12: vue de gauche

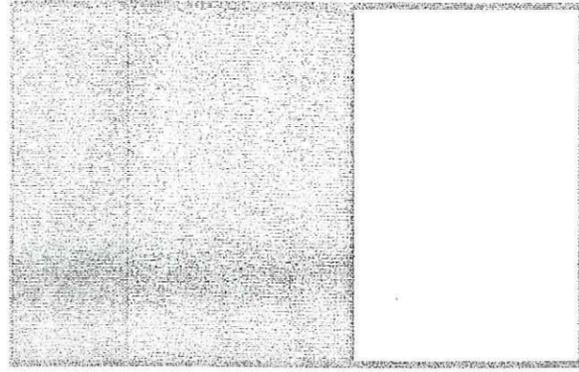


Figure 6.13: Vue de dessus

### C. LES ETAPES DE LA VISUALISATION D'OBJETS TRIDIMENSIONNELS

Pour l'affichage d'un point graphique tridimensionnel les opérations à effectuer sont dans l'ordre :

- Transformation perspective des points caractéristiques en coordonnées écran normalisées

- découpage selon le volume de vision normalisé

- élimination des parties cachées

- projection

- transformation en coordonnées écran (entiers) des points caractéristiques de l'objet découpé et projeté

- tracage

La procédure de définition de la transformation de visualisation 3D effectue le calcul de la matrice homogène, représentative de la perspective généralisée, et permet de spécifier fenêtre

Le principe de cette procédure est de mettre à jour les variables globales que sont la transformation courante et la fenêtre courante, à partir des paramètres de la caméra et de la fenêtre désirée de l'écran .

La matrice déterminée exprime la perspective en coordonnées d'écran normalisées homogènes et se calcule par concaténation des matrices élémentaires correspondant à différentes étapes.

Dans le cas de projection orthogonale les étapes de la visualisation des objets tridimensionnels sont exécutées dans l'ordre suivant:

- La transformation des coordonnées des points en appliquant la matrice transformation orthogonale .

- L'élimination des parties cachées.

- La projection.

- La transformation de visualisation 2d.

- Le découpage par rapport à la fenêtre de l'écran.

- Le tracage

## D. LE DECOUPAGE ET LA NORMALISATION EN TROIS DIMENSIONS

### 1. INTRODUCTION

Si l'on applique directement les formules de perspectives pour obtenir une représentation perspective d'un objet, l'image affichée sur l'écran pourra ne pas être correcte. En effet, il faut éliminer les parties de l'objet qui sont en dehors du champ de vision de l'observateur. L'observateur ne pourra pas voir des objets placés derrière lui. D'où la nécessité de définir un volume de visualisation et de déterminer pour tout objet la partie comprise dans ce champ. Seule cette partie de l'objet sera concernée par l'affichage.

Le volume de visualisation normalisé est défini selon un repère écran normalisé.

### 2. DEFINITION DU VOLUME DE VISUALISATION

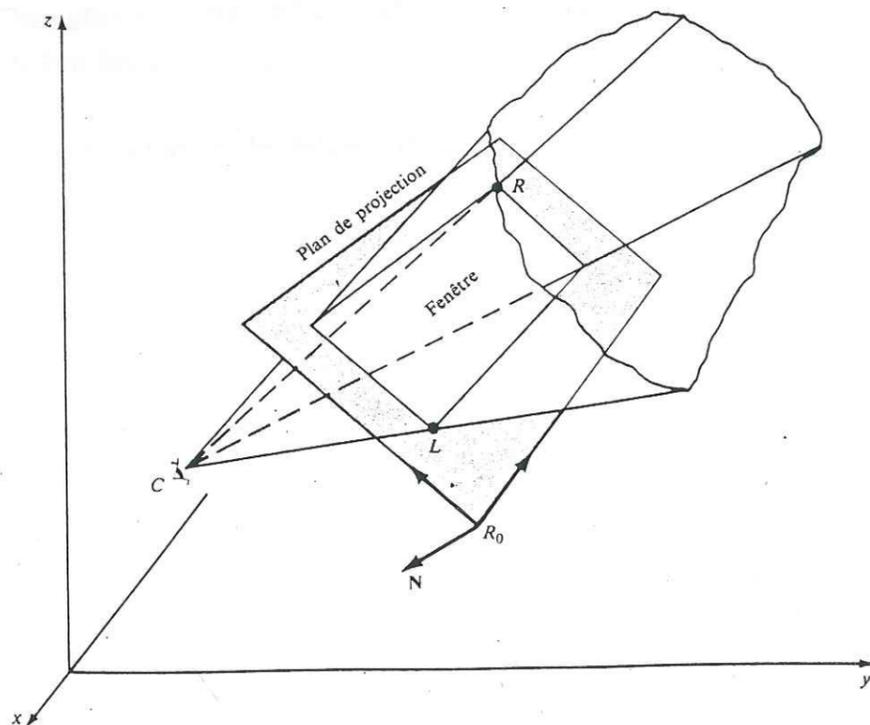


Figure 6.14(Plastock 83)

Le volume de visualisation délimite une région de l'espace des coordonnées absolues dont la projection selon l'axe de vision donne la fenêtre courante (voir figure 6.14).

Pour une vue en perspective, le volume de visualisation correspondant à la fenêtre donnée est un pyramide infini dont le sommet est le centre de projection.

Pour une vue issue de projections parallèles, le volume de visualisation est un parallélépipède dont les arêtes sont parallèles à la direction de projection.

### 3. FENETRAGE AVEC UN VOLUME DE PROJECTION FINI

Le volume de visualisation défini ci-dessus est infini; pour limiter le nombre de points à projeter, on limite le volume par les plans de fenêtrage avant et arrière, parallèles au plan de projection. On peut se limiter au plan de projection et à un plan arrière.

### 4. DECOUPAGE ET FENETRAGE

Le fenêtrage en trois dimensions demande un volume extraordinaire de calcul, pour cette raison deux stratégies de fenêtrage ont été fixées :

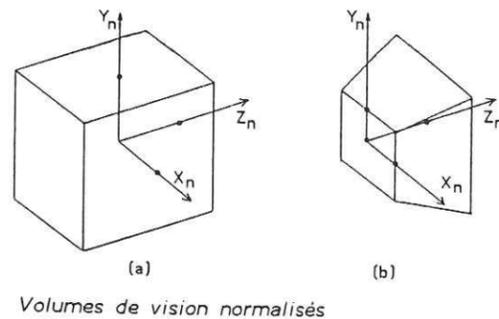
-Fenêtrage direct : ce fenêtrage est directement effectué dans le volume de visualisation, c'est le découpage en coordonnées observateur.

-Fenêtrage canonique : c'est un découpage en coordonnées écran normalisées. On effectue des transformations de normalisation qui transforment le volume de visualisation en un volume de visualisation normalisé (Figure 6.15).

Pour les projections parallèles le volume normalisé est le cube défini par :  $x=-1, x=1, y=-1, y=1, z=-1, z=1$

Pour les projections perspectives, le volume de vision normalisé

est limité par les plans :  $x = -z$ ,  $x = z$ ,  $y = -z$ ,  $y = z$ ,  $z = 0$ ,  $z = 1$ .



Volumes de vision normalisés

Figure 6.15(Newman 79)

### 5. DEFINITION DU SYSTEME DE COORDONNEES NORMALISEES 3D

La définition d'un système de coordonnées normalisées nous permet de ne pas tenir compte de la taille effective de l'écran ou de la partie de l'écran fenêtre dans laquelle on a choisi d'afficher l'image. Cependant, l'extension du repère normalisé 2D  $(x_n, y_n)$  à un repère normalisé 3D se fait par l'introduction d'une troisième dimension  $z_n$ .

On a déjà défini la transformation qui permet de passer des coordonnées d'un point dans le repère objet à ses coordonnées dans le repère observateur.

On étudiera par la suite la transformation de normalisation qui nous fournit les coordonnées d'un point dans le système d'écran normalisé, à partir de ses coordonnées dans le système observateur.

### 6. ELIMINATION D'UN POINT CACHE

L'étude de ce problème revient à déterminer le choix de l'orientation du repère normalisé. Pour afficher un point sur l'écran, on a besoin de connaître ses coordonnées  $x_e, y_e$  qui définissent sa position sur l'écran. Ces coordonnées sont facilement déduites des coordonnées  $x_n, y_n$  du point. Cependant, un problème se pose dans le cas où l'on dispose de deux ou plusieurs points ayant les mêmes coordonnées  $(x_n, y_n)$  mais de profondeurs ( $z_n$ ) et de couleurs différentes (voir figure 6.16). Quel point faut-il afficher dans ce cas?

Intuitivement, on déduit qu'un point A cache un point B si le point B est le plus éloigné de l'observateur. Ce problème est résolu en choisissant l'orientation du repère normalisé qui permettra de donner la valeur la plus importante à  $z_n$  pour le point le plus éloigné de l'observateur. On affiche donc le point le plus proche de l'observateur, soit le point ayant le plus petit  $z_n$ .

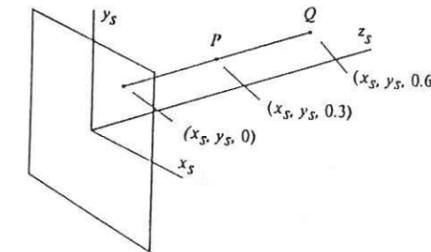


Figure 6.16(Newman78)

### 7. DEFINITION DE LA MATRICE DE NORMALISATION 3D

La décision de n'afficher l'image que dans une partie de l'écran (fenêtre) se traduit en 3D par une décision de projeter tout le volume de visualisation sur cette fenêtre. Cela se traduit aussi par le fait qu'un point n'est pas simplement défini par ses coordonnées normalisées  $x_n, y_n$  mais aussi par  $z_n$ . Cette troisième dimension est orientée de façon à ce que le repère normalisé soit indirect, ce qui se traduit par le fait que la valeur de  $z_n$  augmente en fonction de l'éloignement du point par rapport à l'observateur.

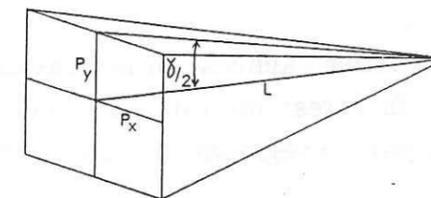


Figure 6.17

On remarque que le rapport de la distance observateur-écran et de la taille du plan de projection (fenêtre) définit l'angle d'ouverture (angle de vision)

C'est l'utilisateur qui définit l'angle d'ouverture verticale. Cela revient à définir la demi hauteur. On détermine  $p_x, p_y$  connaissant la distance  $L$  et l'angle d'ouverture (voir figure 6.17)

$$p_y = L * \text{tg}(\theta/2)$$

$$p_x = p_y * \sin \theta$$

On a par ailleurs déjà signalé qu'on envisage le découpage des objets afin d'éviter une image erronée. Ce découpage est fait selon le volume de vision, nécessairement avant la projection. Cependant, on peut l'effectuer après la transformation perspective: on peut alors définir une matrice qui traduit la composition de la normalisation et de la transformation perspective. Ce n'est qu'ensuite qu'on effectue la projection proprement dite qui, consiste à éliminer la troisième dimension.

On a déjà vu l'utilité du choix du système des coordonnées écran normalisées pour l'élimination des parties cachées. En effet, on peut concevoir celle-ci comme un processus qui va décider, lorsque deux points ont la même projection, que c'est celui qui a la plus petite  $z_n$  qui est visible, car il est le plus proche de l'observateur. En normalisant selon l'axe  $z$ , on ramène un intervalle de "distance à l'observateur" entre 0 et 1, c'est à dire que l'on introduit deux plans de découpage supplémentaires (un plan frontal et un plan arrière).

Pour l'élimination des parties cachées, on ne travaille pas point par point mais plutôt sur des facettes: on interpole l'éloignement en se déplaçant entre deux points sur un segment, il est donc nécessaire que la linéarité soit conservée.

Or, dans le cas de la perspective, la linéarité est détruite du fait de la division des coordonnées  $x_n, y_n$  par  $z_p$  ( $z_p$  change selon les points) lors du passage des coordonnées homogènes aux coordonnées cartésiennes, c'est ce passage qui pose problème.

On doit pouvoir travailler en coordonnées d'écran normalisées après la perspective tout en conservant l'information d'éloignement de l'observateur  $z_p$  requise pour l'algorithme de l'élimination des parties cachées. L'astuce vient du fait que le passage de la représentation homogène à la cartésienne implique une division par la coordonnée  $w$  et que la linéarité est conservée tant que le retour à la représentation cartésienne n'est pas effectué.

Le plan de projection est le plan frontal  $Z = L$  alors que le plan arrière est le plan  $Z = \text{depth}$ .

Les coordonnées d'écran normalisées (homogènes) sont définies comme suit :

$$x_n = x_p / p_x$$

$$y_n = y_p / p_y$$

$$z_n = (z_p/L - 1) / (1 - L/\text{depth})$$

$$w_n = z_p/L$$

Pour l'axe  $Z$ , on désire avoir des valeurs comprises entre 0 et 1 lors du retour en valeurs cartésiennes; en effet, en normalisant  $z$  comme défini ci-dessus, on obtient pour  $z_p=L$  la coordonnée cartésienne:

$$z_{\text{cart},n} = z_n / w_n$$

$$= (L/L - 1) * (1 - L/\text{depth}) / w_n$$

$$= 0 / w_n = 0$$

et pour  $z_p = \text{depth}$

$$z_{\text{cart},n} = z_n / w_n$$

$$= (\text{depth} / L - 1) * (1 - L/\text{depth}) / w_n$$

$$= (\text{depth}/L) / (\text{depth}/L) = 1$$

La coordonnée  $w$  sera utilisée pour réaliser la perspective, elle vaut  $z_p/L$ . La division par  $w$  produit bien l'effet de réduction en fonction de l'éloignement.

Ce changement de repère, traduit par les équations, et qui effectue la normalisation et la perspective, s'exprime par la matrice NORMPER =

$$\begin{vmatrix} 1/px & 0 & 0 & 0 \\ 0 & 1/py & 0 & 0 \\ 0 & 0 & 1/L^* & -1/ \\ 0 & 0 & 1/L & 1 \end{vmatrix} = 1 - L/\text{depth}$$

Il faut s'assurer que depth est effectivement plus grand que L, sinon on ne considère que le sommet de la pyramide de vision.

### 8. ALGORITHME DE DECOUPAGE EN 3D

#### a) DECOUPAGE POUR UNE PROJECTION PERSPECTIVE

##### (1) découpage d'un point

Définissons les équations des plans exprimant le volume de visualisation ; les plans latéraux étant verticaux, leurs équations exprimées dans le repère observateur orienté à gauche, sont les suivantes :

$$\text{Plan gauche: } xp + tg(x/2) * zp = xp + (px/l) * zp = xp + mx * zp = 0$$

$$\text{Plan droit: } xp - tg(x/2) * zp = xp - (px/l) * zp = xp - mx * zp = 0$$

Les plans inférieurs et supérieurs sont régis par des équations semblables où y remplace x:

$$\text{Plan inférieur: } yp + tg(y/2) * zp = yp + (py/l) * zp = yp + my * zp = 0$$

$$\text{Plan supérieur: } yp - tg(y/2) * zp = yp - (py/l) * zp = yp - my * zp = 0$$

Un point appartient au volume de vision si et seulement si:

$$xp + mx * zp > 0 \text{ (plan gauche)}$$

$$xp - mx * zp > 0 \text{ (plan droit)}$$

$$yp + my * zp > 0 \text{ (plan inférieur)}$$

$$yp - my * zp > 0 \text{ (plan supérieur)}$$

En transformant ces inégalités en coordonnées écran normalisées homogènes, on obtient les inégalités suivantes :

$$xn > -wn \quad \langle \rangle \quad wn + xn > 0 \text{ (plan gauche)}$$

$$xn < wn \quad \langle \rangle \quad wn - xn > 0 \text{ (plan droit)}$$

$$yn > -wn \quad \langle \rangle \quad wn + yn > 0 \text{ (plan inférieur)}$$

$$yn < +wn \quad \langle \rangle \quad wn - yn > 0 \text{ (plan supérieur)}$$

La normalisation implique que le plan avant aît pour équation:  $zn > 0$  et le plan arrière  $zn < +wn$ , ce qui correspond -après la conversion en coordonnées normalisées cartésiennes- à une coordonnée z comprise entre 0 et 1.

Cela conduit à définir, comme un outil de travail, un nouveau système de coordonnées. Il s'agit d'un système de coordonnées de volume de vision dans lequel un point défini dans le système par six coordonnées  $(xn, yn, zn, wn)$ , sera défini par six coordonnées  $(wn+xn, wn-xn, wn+yn, wn-yn, zn, wn-zn)$ . Ces coordonnées correspondent à la distance entre le point et chacun des plans délimitant le volume de vision.

D'après les inéquations définissant l'appartenance d'un point à un volume de vision normalisé, on peut déduire la conclusion suivante: Un point appartient au volume de vision normalisé si, et seulement si, ces 6 coordonnées (dans le système de coordonnées de volume normalisé) sont toutes positives.

##### (2) découpage de segment de droite

Le problème le plus important est de déterminer le point d'intersection d'un segment et d'un plan. Le segment étant donné par ses deux extrémités, on travaillera en représentation paramétrique. Cette représentation est une interpolation qui suppose nécessairement que la linéarité soit conservée. C'est la raison pour laquelle on effectue le découpage dans les coordonnées normalisées homogènes, et non pas cartésiennes. On rappelle que le passage des coordonnées normalisées homogènes aux coordonnées normalisées cartésiennes s'opère par une division par les coordonnées Z ce qui traduit la linéarité et empêche d'interpolation permise par les représentations paramétriques.

-la représentation paramétrique:

L'équation vectorielle du segment à découper est :  
 $p_1 + (p_2 - p_1)t$ . L'équation du plan F (passant par l'origine) est:  
 $ax + by + cz = 0$ .

Les coefficients du plan sont présentés par le vecteur  $F[a, b, c]$ , le point d'intersection P est défini par le produit scalaire  $P \cdot F = 0$ , ce qui correspond à  $(p_1 + (p_2 - p_1)t) \cdot F = 0$ . Au point d'intersection, on a  
 $t = P \cdot F / ((p_2 - p_1) \cdot F)$ .

Prenons par exemple le plan  $x = -w$  et un segment dont l'origine est à l'intérieur de la limite (car  $w_1 + x_1 \geq 0$ ) et dont l'extrémité est à l'extérieur ( $w_2 + x_2 < 0$ ). Ces deux quantités enregistrent la distance de chacun des deux points par rapport au plan  $x = -w$ . Le point d'intersection est caractérisé par la valeur

$$t = w_1 + x_1 / ((w_1 + x_1) \cdot (w_2 + x_2))$$

$t = 0$  à l'origine,  $t = 1$  à l'extrémité, pour définir le point d'intersection, on utilise la valeur de  $t$  pour calculer  $(x, y, z, w)$ . Par exemple  $x = t(x_2 - x_1) + x_1$ .

(3) l'algorithme de découpage

On calcule une valeur de  $t$ , à l'intersection de la droite avec ce plan. Pour chacun des plans définissant le volume, le nombre des points d'intersection avec le volume (convexe) ne peut être que 0, 1 ou 2. Les deux réels solutions de  $t$  sont le plus grand minimum et le plus petit maximum des six valeurs de  $t$  obtenues (ces deux valeurs solutions sont comprises entre 0 et 1) et le segment ne possèdera une intersection avec le volume (existence de découpage) que si le plus grand minimum est inférieur au plus petit maximum.

#### b) DECOUPAGE POUR UNE PROJECTION ORTHOGONALE

Comme on a déjà vu dans l'étude des étapes de visualisation des objets tridimensionnels, on a choisi d'effectuer le découpage en deux dimensions dans le cas de projection orthogonales.

Ce choix a été fait en raison de la position de l'observateur dans les projections orthogonales. Celui-ci étant placé à l'infini, on n'aura pas à éliminer les objets situés derrière lui. Le seul objectif du découpage dans ce cas est de confier l'image à l'intérieur de la fenêtre, ce qui peut être parfaitement réalisé à l'aide d'un simple découpage en deux dimensions.

Cependant, on a appliqué ce genre de découpage même dans le cas des projections perspectives mais uniquement quand on part de l'hypothèse qu'aucun point de l'objet n'est placé derrière l'observateur.

### E. L'ELIMINATION DES PARTIES CACHEES

#### 1. INTRODUCTION

On veut afficher des objets tridimensionnels avec un degré de réalisme important. On a vu dans les chapitres précédents que la projection en perspective permet de créer l'impression de profondeur, ce qui donne un réalisme à l'image produite. Mais la seule projection en perspective ne produit qu'un squelette filiforme appelé "fil de fer" ce qui n'est pas très clair et peut même être ambigu (voir figure 6.18). C'est le cas d'un cube dessiné avec toutes ses arêtes; seule l'élimination des parties cachées permet de déterminer sous quel angle il est vu.

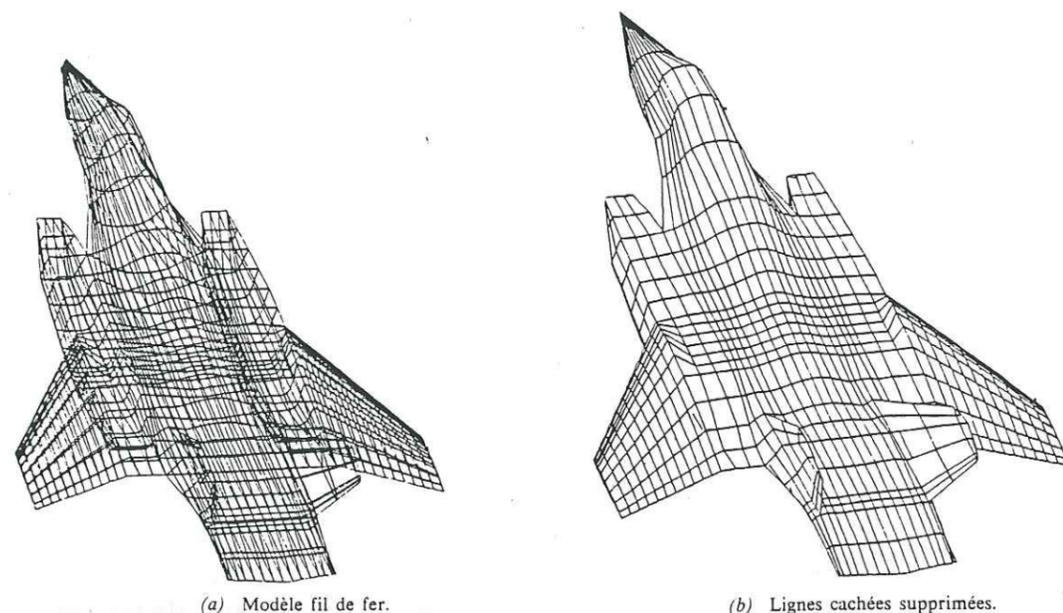


Figure 6.18(bouille 80)

L'élimination des parties cachées permet de conférer aux objets leur opacité naturelle, leur donnant l'aspect de volumes pleins opaques, masquant les objets placés derrière eux. Cela améliore le réalisme de l'image et la rend suffisamment représentative pour permettre l'identification des objets et de leurs positions respectives. Cependant il faut noter qu'une image dont les parties cachées ont été éliminées est plus réaliste, mais comporte moins d'information. Il est donc nécessaire de pouvoir l'examiner sous divers angles de vue pour observer tous les détails.

Il est évident que les parties cachées dépendent de la position de l'observateur, ce qui implique que l'élimination des parties cachées ne doit être faite qu'après application de la transformation de visualisation, c'est à dire la perspective dans le système de coordonnées écran normalisées

## 2. ETUDE ET CLASSIFICATION DES ALGORITHMES

Il existe un grand nombre d'algorithmes de traitement des parties cachées différents, dont aucun n'est vraiment le meilleur; ces différents

algorithmes répondent à différents besoins; parfois très spécifiques. Les algorithmes décrits ici présentent les principales solutions utilisées; tous peuvent être améliorés et combinés dans le but de produire de nouveaux algorithmes répondants à des applications différentes.

Les problèmes à résoudre pour l'élimination des parties cachées sont par ordre de complexité :

- Le retrait des faces arrières,
- le masquages des objets arrières,
- l'intersection des volumes.

En se basant sur la constatation que les objets proches masquent les objets lointains, l'élimination des parties cachées implique un tri des facettes selon la distance qui les sépare de l'observateur. En général, les algorithmes calculent d'abord la distance des facettes à l'observateur. Une seconde phase consiste à déterminer (par un tri latéral et vertical) les facettes placées derrière une autre et donc masquées par celle-ci. La rapidité de la méthode de tri employé détermine l'efficacité de l'algorithme.

Cependant on différencie deux grandes catégories d'algorithmes d'élimination en fonction du système de coordonnées dans lequel les calculs sont effectués. Cela est illustré par le tableau suivant:

algorithmes dans  
l'espace objet

principe : chaque facette est comparée à toutes les autres facettes

temps de calcul : croît avec le carré du nombre de facettes

données : une description géométrique du volume

résultat : une autre description géométrique, soit des lignes, soit des surfaces

ces algorithmes concernent l'élimination des lignes et des surfaces cachées

Avant de résumer le principe de chacun des algorithmes principaux, il est important de signaler que la manière de modéliser les volumes a une influence notable sur les algorithmes.

#### a) ) Z-BUFFER

C'est une méthode simple d'élimination des parties cachées qui opère dans l'espace image. Il s'agit de réserver une zone mémoire, appelée matrice Z (Z-buffer) dans laquelle on stocke pour chaque pixel sa coordonnée écran normalisée Z (entre 0 et 1). Pour chaque point (pixel : x,y) à tracer, l'algorithme calcule sa coordonnée Z et la

algorithmes dans  
l'espace image

principe : on ne calcule que les parties masquées qu'à la définition requise par le dispositif d'affichage.

temps de calcul : proportionnel au nombre de facettes F multiplié par le nombre de pixels de l'écran

données : une description géométrique du volume

résultat : ne fournit aucune description géométrique et le résultat est une image en surface qui ne permet plus aucune manipulation

ces algorithmes ne concernent que les surfaces cachées

compare avec la valeur de Z stockée dans le Z-buffer pour le même pixel.

Pour les mêmes coordonnées x,y le point le plus proche de l'observateur est celui qui a la valeur Z la plus petite.

Si  $Z(x,y) < Z\text{-buffer}(x,y)$  on affiche le nouveau point et on met à jour la matrice Z (en enregistrant la plus petite valeur de Z). Cette matrice contient donc à tout instant la coordonnée Z de chacun des points affichés.

L'algorithme du Z-buffer convient particulièrement pour des scènes complexes car le temps de traitement ne croît que linéairement avec le nombre des faces.

#### b) LIGNE DE BALAYAGE

Un algorithme de ligne de balayage (Schweizer 87) consiste essentiellement en deux boucles imbriquées : une boucle de balayage en x imbriquée dans une boucle de balayage en y (voir figure 6.19).

-Le balayage en y : Pour chaque valeur de  $y = a$ , on recherche l'intersection des polygones à afficher avec le plan de balayage  $y = a$ . Ce plan de balayage est parallèle au plan (x,y). Les intersections trouvées sont des segments dans ce plan

-balayage en x : pour chaque valeur de x, par exemple  $x = b$ , on recherche les intersections des segments trouvés ci-dessus avec les lignes de balayage  $x = b$  situées dans le plan de balayage y. Ces intersections constituent un ensemble de points situés sur la ligne de balayage x.

On trie ces points par rapport à leurs coordonnées z. Le point (x,y,z) qui a la plus petite valeur de z est visible et le pixel correspondant à (x, y) prend la couleur du polygone contenant le point.

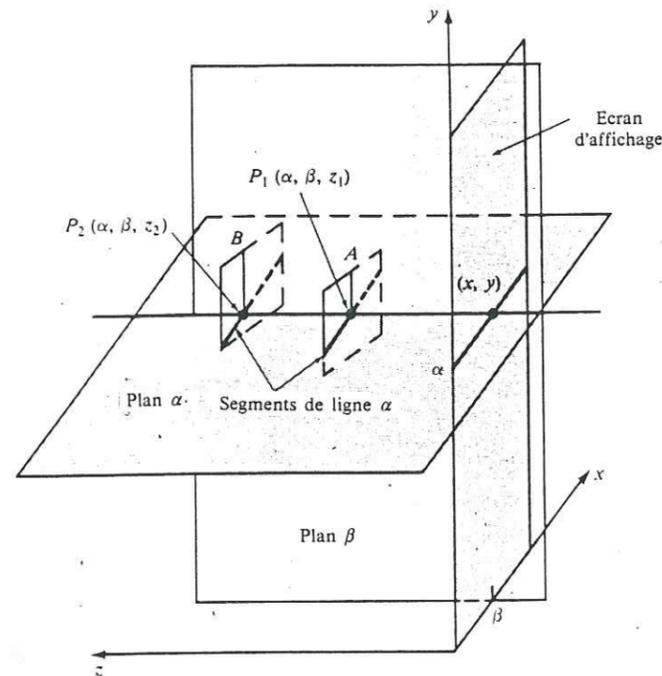


Figure 6.19 (Plastock 87)

Les algorithmes de lignes de balayages présentent deux avantages : d'une part ils peuvent être implantés dans le matériel et d'autre part ils peuvent être facilement généralisés à des éléments non-linéaires (non polygonaux).

#### c) SUBDIVISION

Les algorithmes de subdivision (Newman 79) mettent en oeuvre des procédures récursives, basées sur la subdivision de l'image, jusqu'à obtenir une sous-image simple à analyser. Le processus de subdivision est arrêté lorsque la sous-image ne comporte plus qu'une face ou que sa taille est devenue celle d'un point.

#### d) PEINTRE

Dans les algorithmes de peintre (Plastock 87), on trie les faces selon l'éloignement. Cet algorithme traite les polygones comme s'ils devaient être peints sur l'écran, ordonnés en fonction de leurs distances par rapport à l'observateur. La facette de plus basse priorité, c'est à

dire la plus éloignée de l'observateur, est tracée en premier. Les polygones plus proches sont peints par dessus les polygones les plus éloignés, les soustrayant partiellement ou totalement à la vue.

La difficulté des algorithmes de peintre réside dans le tri des facettes en fonction de leurs éloignement.

C'est l'algorithme de Newell, Newell et Sancha (N.N.S) qui a été choisi dans notre application. La justification de ce choix sera présentée ultérieurement.

### 3. CHOIX DE L'ALGORITHME

Nous présentons ci-dessous un tableau comparatif (voir figure 6.20) dressé par Lucas (1977) et reproduit par Boolles (1980) qui synthétise les caractéristiques des quatre algorithmes suivants :

-Galimberti et Montanar : Cet algorithme, comparable à celui d'Appri, est basé sur la comparaison des arêtes aux faces.

-Algorithme de Warnock : Appartient à la catégorie des algorithmes de balayage.

-N.Sancha : C'est un algorithme de peintre, il est donc basé sur le tri selon l'éloignement.

	Scènes	Image	Dessin	Affichage arêtes d'intersection	Transparence	Remplissage	Mémoire	Programmation	Comportement
WARNOCK	faces polygonales sans trou Pénétration autorisée	Moyen	Mauvais	Oui	Non	Moyen - le nombre de contours à remplir peut être important	$3 * nbscm + 37 * nbfaces$	Simple	Mauvais
WATKINS	faces polygonales Trous Pénétration pour la méthode boîte	Bon	Moyen	Non	Possible	bon - Adapté au balayage télévision	$3 * nbscm + 33 * nbfaces$	Complexe	bon - La méthode Pile est dans l'ensemble meilleure que la méthode boîte
ATHERTON	faces polygonales Trous Pénétration	Bon	Bon	Non	Possible	bon - Nombre minimal de contours à remplir	$5 * nbscm + 40 * nbfaces$	Moyen	Moyen
NEWELL	faces polygonales Trous Pénétration	Bon	Mauvais	Non	Oui	Mauvais - Grand nombre de contours à remplir	$3 * nbscm + 40 * nbfaces$	Simple	Moyen

Figure 6.20

En ce qui concerne l'efficacité des algorithmes étudiés (voir figure 6.21), Newell a dressé la comparaison suivante

Algorithm	Number of faces			Approximate ratio
	100	2,500	60,000	
Depth-buffer	7.5	7.5	7.5	1: 1: 1
Span-coherence	0.5	3	64	1: 6: 20
Warnock	1.5	9	43	1: 6: 5
Newell, Newell, and Sancha	0.14	1.4	71	1:10:50

Figure 6.21

Cela montre que pour le Z-buffer, le temps d'exécution n'est pas sensible au nombre de faces. ce qui le rend inintéressant pour des programmes traitant un petit nombre de faces ( de 10 jusqu' à 2500 ).

N.N.S a le temps d'exécution le plus faible mais celui-ci augmente presque exponentiellement au delà des 2500 faces .

Au vue de l'ensemble de ces caractéristiques ,et en fonction de l'application souhaitée, on a choisi l'algorithme de N.N.S pour les raisons suivantes .Nous les présentons par ordre d'importance :

- Le résultat de cet algorithme n'est pas une image mais un modèle de visibilité . Celui-ci est un ensemble ordonné de faces à afficher ; il n'est ,en fait , qu'une réorganisation du modèle de description ( la description initiale du volume ) . Si l'application désire un ensemble de vues successives d'une scène , ou si l'observateur change de position, il sera intéressant de réutiliser le modèle de visibilité comme modèle de description . Ainsi le nombre d'opérations à effectuer pour passer d'une vue à une autre sera réduit.

-Pour la synthèse d'image , cet algorithme permet de rendre les effets de transparence et de modifier l'aspect des éléments du modèle de description .

-Cet algorithme produit une bonne qualité d'image .

-Présente une facilité d'adaptation pour l'introduction de l'effet de luminosité.

-Présente une facilité d'adaptation pour les surfaces non planes

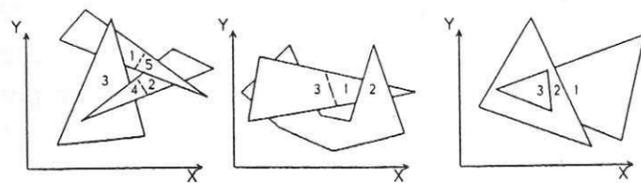
-Le temps d'exécution n'est pas important pour un nombre moyen de faces

#### 4. L'ADAPTATION DE L'ALGORITHME DE N.N.S

##### a) LE PRINCIPE DE L'ALGORITHME

Comme on l'a déjà signalé, l'algorithme de Newell, Newell et Sancha est dérivé de la méthode de peintre (Boule 80). Sa fonctionnalité principale consiste à attribuer une priorité à chacune des faces des objets modélisés en polygones.

Un tri préliminaire est effectué sur la coordonnée minimum de chaque face. Cependant, le simple tri selon la coordonnée  $z$  génère des ambiguïtés. Ces ambiguïtés s'accroissent en cas de chevauchement ou de superposition cyclique (voir figure 6.22).



Cas de superposition cyclique

Figure 6.22 (Schweizer 87)

L'algorithme présente des solutions élégantes pour résoudre ce genre de problèmes.

##### b) LA PARTICULARITE DES SURFACES GEOLOGIQUES

Le problème le plus délicat à résoudre dans l'algorithme de N.N.S c'est le cas de chevauchement cycliques des faces. Quand le cas se présente un découpage de faces devient nécessaire et un nouveau tri doit être effectué pour tenir compte des nouvelles faces (produites à partir du découpage). C'est le cas qui demande le plus d'opération à exécuter.

Pour la visualisation des surfaces géologiques, on adopte l'hypothèse où le cas du chevauchement cyclique ne se présente pas. La vérification de cette hypothèse est évidente : en effet si on veut visualiser le toit d'une couche, on découpe la surface du toit en plusieurs

facettes, l'intersection de deux facettes peut être un point, un arête ou l'ensemble vide. L'union de ces facettes est la surface de la couche, donc ces facettes ne peuvent pas se chevaucher.

Le choix de l'algorithme de N.N.S est convenable dans notre cas parce qu'on peut le simplifier en éliminant certaines opérations. Cette simplification est justifiée par l'hypothèse de la continuité des surfaces à visualiser.

##### c) AFFECTATION DES PRIORITES

On affecte aux polygones des priorités basées sur la profondeur, en déterminant quels polygones occultent un polygone  $P$ . On effectue d'abord un prétraitement qui détermine quels sont les polygones  $Q$  qui n'occultent pas le polygone  $P$ .

Le domaine  $Z$  d'un polygone est la région située entre le plan  $Z = Z_{min}$  et le plan  $Z = Z_{max}$  (figure 6.23).  $Z_{min}$  est la plus petite valeur de toutes les coordonnées  $z$  de tous les sommets du polygone; et  $Z_{max}$  est la plus grande valeur. De la même façon, on définit  $X_{min}$ ,  $X_{max}$ ,  $Y_{min}$ ,  $Y_{max}$ .

##### d) LE TEST DES FACES CACHEES

Le polygone  $P$  ne cache pas le polygone  $Q$ , si l'un des tests suivants, effectués dans l'ordre, est vrai :

-Test-0 : Les domaines  $Z$  de  $P$  et de  $Q$  ne se chevauchent pas et  $Z_{max}$  de  $Q$  est plus petit que  $Z_{min}$  de  $P$  (voir figure 6.23).

-Test-1 : Les domaines  $y$  de  $P$  et de  $Q$  ne se chevauchent pas (voir figure 6.24).

-Test-2 : Les domaines  $x$  de  $P$  et de  $Q$  ne se chevauchent pas.

-Test 3 : Tous les sommets de  $P$  sont situés du côté du plan  $Q$  le plus loin de l'observateur (voir figure 6.25).

-Test-4 : Tous les sommets de  $P$  sont situés du côté du plan de  $P$  le plus proche de l'observateur (voir figure 6.26).

-Test-5 : Les projections des polygones de P et de Q ne se chevauchent pas . Ceci est testé en recherchant les intersections de chaque côté d'un polygone avec chaque côté de l'autre .

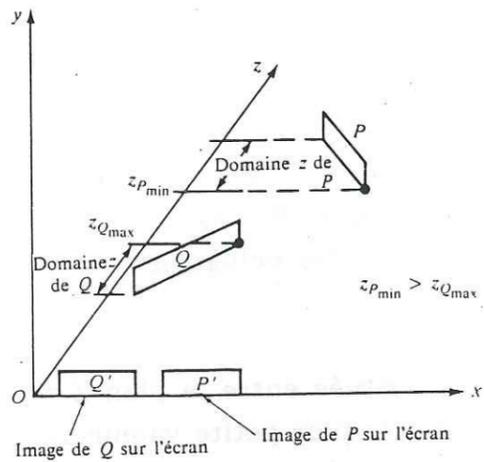


Figure 6.23

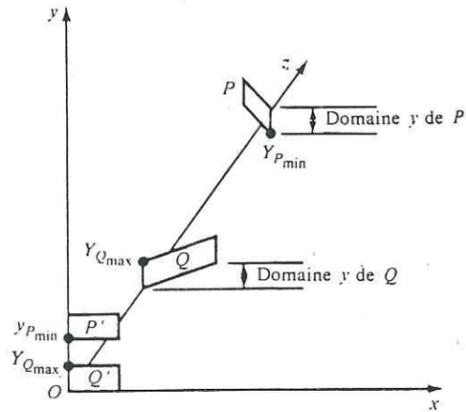


Figure 6.24

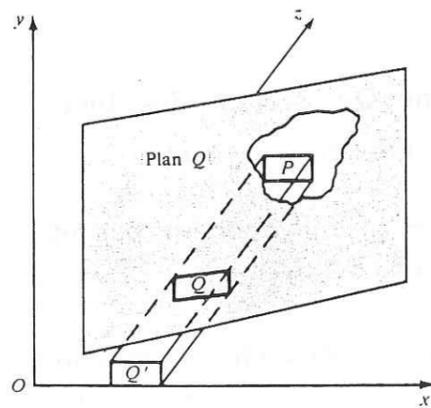


Figure 6.25

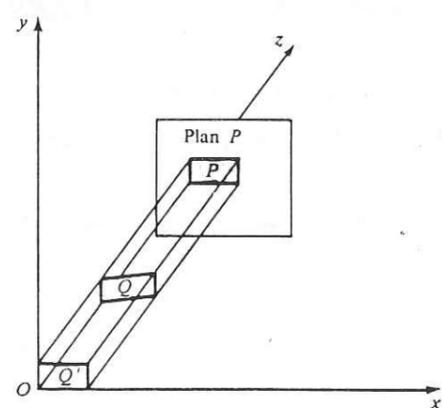


Figure 6.26

e) L'ALGORITHME GENERAL

1-Définir pour chaque polygone le Zmin , Zmax, Xmin , Xmax , Ymin, Ymax .

2-Trier la liste de polygones(liste de description) selon les coordonnées Zmin ,par ordre décroissant.

3-Soit P la face la plus éloignée (tête de liste ), et S la suivante , comparer la face P avec la face S pour vérifier si réellement P doit être dessinée avant S.

3-1: Si le sommet le plus proche PZmin de la face P est plus éloigné que le sommet le plus éloigné SZmax de S , P ne peut obscurcir S (test-0). L'ordre est correct et P peut être tracé .

3-2: Dans le cas contraire  $PZ_{max} < SZ_{min}$  , l'ordre de ces deux faces est incorrect , mais peut-être aussi l'ordre des faces qui suivent . Il faut donc définir l'ensemble Sp tel que:  $SP = \{S / Z_{max}(S) \geq Z_{min}(P) \}$  . C'est l'ensemble des faces pouvant être potentiellement masquées par P . Cet ensemble doit être réordonné . La face P ne pourra être affichée qu'après s'être assuré qu'elle ne cache aucune face de l'ensemble S , ce qui est le cas dès que l'un des cinq tests suivants est positif :

-Les domaines y de P et de S ne se chevauchent pas (test 1)

-les domaines x de P et de S ne se chevauchent pas (test 2)

-la face P est entièrement du côté opposé à l'observateur par rapport au plan déterminé par S .

-la face S est entièrement du même côté que l'observateur par rapport au plan déterminé par P .

-les projections de P et de S ne se chevauchent pas

3-2-1: Ces tests sont faits dans l'ordre croissant de leur complexité, et doivent être appliqués à chaque face de l'ensemble {Sp}. Si l'un des tests est positif, la face P ne peut masquer S et sera affichée.

3-2-1: Lorsque tous ces tests sont négatifs pour un couple de face P et S, cela signifie que P masque S. On échangera alors P et S dans la liste. Si maintenant les tests sont positifs, on sait que S ne masque pas P et l'ordre est trouvé. Dans le cas contraire, les faces P et S se recouvrent mutuellement. La face P est alors décomposée en deux faces selon sa ligne d'intersection avec le plan de S. Ces deux faces remplacent P dans la liste et les mêmes tests leurs sont appliquées

#### f) L'ANALYSE ET LA REALISATION DE L'ALGORITHME

##### (1) le modèle de description

###### (a) description des éléments

On étudie ici la visibilité d'un ensemble de faces polygones planes. Chaque face polygonale est décrite par un contour extérieur convexe ou non, ainsi que par des contours intérieurs qui définissent des trous. De plus les éléments qui constituent la scène peuvent se pénétrer (ce n'est pas le cas dans notre application). On n'utilise dans cet algorithme aucune information d'aspect pour déterminer les éléments visibles

###### (b) représentation des éléments traités

Il a déjà été signalé que la représentation des éléments dépend des algorithmes choisis. On détaillera la représentation des volumes dans un chapitre suivant. Cependant, on peut dès maintenant signaler que chaque contour est représenté par la suite des sommets qui le définissent. D'autres informations, pour chaque face, sont calculés avant l'étude de visibilité. Il s'agit :

- De  $Z_{max}, Z_{min}, Y_{max}, Y_{min}, X_{max}, X_{min}$
- Des coefficients de l'équation du plan, support de chaque face

##### (2) tri initial de données

Comme cela a été établi dans la description de l'algorithme général, l'ensemble des faces constituant le modèle de description sera trié suivant la valeur  $Z_{min}$  par ordre décroissant. La première face de la liste aura une forte probabilité de ne cacher aucune autre face.

##### (3) les opérations géométriques utilisées

selon l'hypothèse de la non-superposition des faces Les opérations de découpages des faces ne seront pas exécutées. Les opérations géométriques nécessaires sont celles de la détermination des positions relatives des projections des deux faces

Il s'agit de trouver une des réponses suivantes

- une face contient une autre
- les deux faces sont disjointes
- les deux faces se coupent.

Pour définir la relation entre deux faces, on utilise les tests suivants par ordre croissant :

-Tests minimaux : Les deux faces 1 et 2 sont disjointes, si et seulement si:

$$(X_{max1} < X_{min2}) \text{ ou } (X_{max2} < X_{min1})$$

-Comparaison des deux contours :

- On cherche si les deux contours se coupent,
- s'ils ne se coupent pas, on détermine lequel contient l'autre,
- si aucune conclusion n'est trouvée les deux contours sont disjointes.

##### (4) la fonction stratégie

La fonction stratégie repose sur le fait que l'on dispose de tests simples (test1, test2, test3, test4, test5) qui permettent de montrer

qu'une face n'en cache pas une autre mais qui ,par contre ,ne permettent pas d'affirmer qu'une face en cache une autre .

(5) l'ensemble des représentations intermédiaires

Dans une représentation intermédiaire se trouvent deux types de faces :

- des faces qui appartiennent au modèle de description ,
- des faces qui proviennent d'un découpage

(6) le modèle de visibilité

Le modèle de visibilité est constitué par un ensemble de faces ordonnées . L'ordre est en fait celui suivant lequel les faces doivent être affichées ou traitées par le module de synthèse d'image . Le modèle de visibilité n'est qu'une réorganisation du modèle de description .Les seules modifications sont celles dues au découpage

Les figures suivantes illustrent l'importance du tri des faces ,selon la valeur de  $z$  , ainsi que l'effet de l'élimination des surfaces cachées .

La figure 6.5 représente une projection perspective d'un modèle de terrain

La figure 6.27 représente la projection perspective selon le même point de vue, du même terrain mais sans l'élimination des surfaces cachées.

La figure 6.28 représente la projection perspective, selon le même point de vue , du même terrain , sans élimination des surfaces cachées mais avec un tri des faces selon la valeur de  $z$  : le tri des faces selon  $z$  n'est pas suffisant pour rendre une image réaliste .

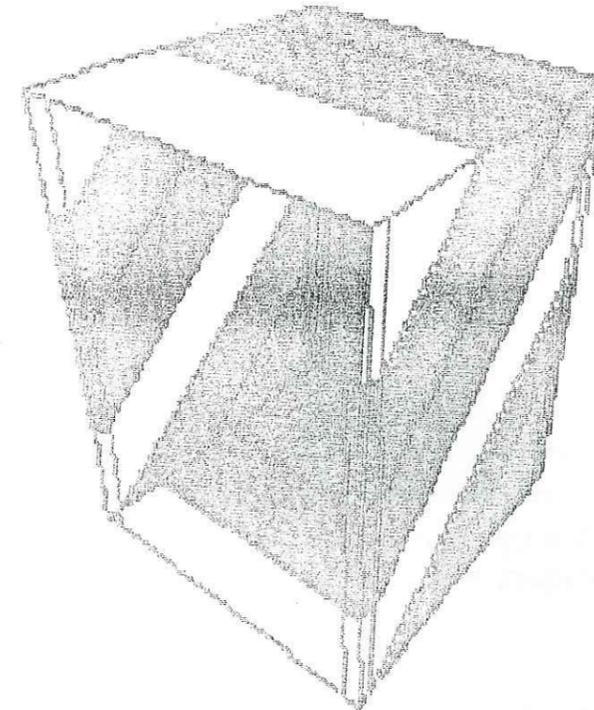


Figure 6.27:sans tri

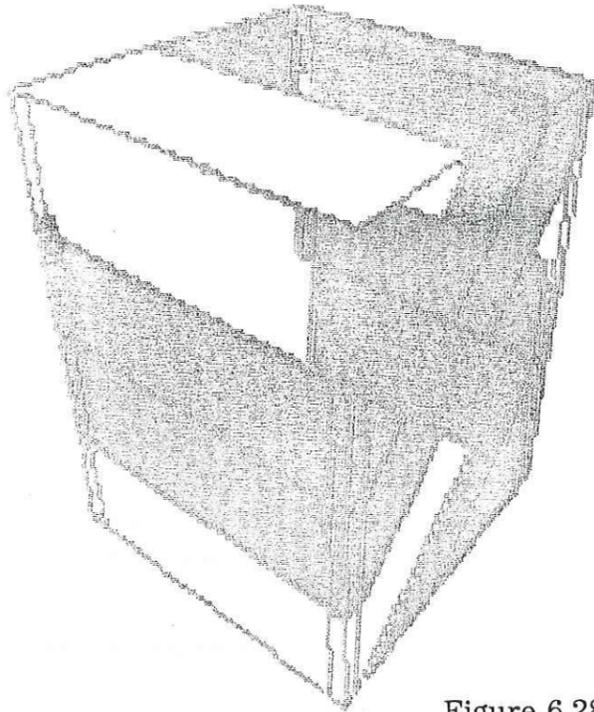


Figure 6.28: le tri selon Z est insuffisant

## VII. LE DOSSIER DE REALISATION DE LA BIBLIOTHEQUE GRAPHIQUE

### A. INTERET DE LA BIBLIOTHEQUE GRAPHIQUE

Dans la pratique, la majorité des applications graphiques sont construites en s'appuyant sur des bibliothèques graphiques existantes.

Le choix de réaliser une bibliothèque graphique a été effectué dans le but d'assurer une certaine portabilité du logiciel. En effet, en développant les applications à partir d'une bibliothèque graphique, on les rend portables, c'est à dire indépendantes de la configuration matérielle utilisée. Cela permet de travailler à un niveau d'abstraction plus élevé et de mieux séparer les problèmes graphiques de bas niveau du reste de l'application.

L'avantage de la bibliothèque graphique est de permettre à l'utilisateur de disposer de divers modules de routines orientés vers différentes applications. En général, les langages de programmation n'offrent pas de primitives graphiques avancées. Le turbo pascal 4 ne nous offre que les primitives de traçage de base dans une unité graphique. Nous avons utilisé ces primitives de base pour la construction de notre bibliothèque.

### B. PRESENTATION DES MODULES

Cette bibliothèque est formée par un ensemble de modules de modélisation et de visualisation des objets en deux et en trois dimensions, avec l'élimination des surfaces cachées.

nous avons construit cette bibliothèque sur la base des objets définis dans le cadre de la conception orientée objet. Pour cela nous avons exploité le caractère typé du Pascal-4 ainsi que ses possibilités de création de modules et de leurs compilation séparée. On a donc associé à chaque objet un type ou un module. La compilation séparée des modules nous a permis d'organiser la bibliothèque selon une architecture modulaire et hiérarchique.

Les procédures de visualisation , de perspective et d'élimination des parties cachées qui s'exécutent à l'intérieur de ces modules ont été détaillées dans le chapitre précédant: "Réalisation des procédures de visualisation" .

Le choix des structures de données ainsi que les procédures de modélisation des volumes seront détaillés dans un chapitre suivant : "La modélisation géométrique des objets".

On présente dans ce chapitre les différents modules composant la bibliothèque graphique , leurs rôles et leurs interfaces

L'interface d'un module est la partie accessible par les autres modules et comporte les déclarations de variables et de types globaux ainsi que les fonctions d'accès. Pour chacune des fonctions d'accès, on spécifie les paramètres et l'effet .

Pour les modules que nous détaillerons par la suite ,nous avons choisi la représentation la plus proche de celle du turbo pascal-4.

### 1. LE MODULE POINT

#### LES FONCTIONS D'ACCES:

Certaines de ces fonctions d'accès utilisent des variables globales concernant surtout la définition d'un point, des systèmes de coordonnées et des transformations de visualisation 2D et 3D . Ces variables sont contenues dans un module de déclarations globales

Les coordonnées écran sont exprimées en coordonnées écran relatives c'est à dire en coordonnées fenêtre :c'est le principe d'affichage relatif qui nous impose cette contrainte; les transformations 2d et 3d nous donnent directement les coordonnées fenêtre d'un point .

La procédure ts2de fait correspondre à un point du plan utilisateur ,un point de l'écran exprimé en coordonnées fenêtre, en utilisant la transformation courante 2D:

```
PROCEDURE ts2de (ptin: point2d; var ptout :point);
```

Tse2d est la procédure inverse de la précédente ,elle fait appel à l'inverse de la transformation courante :

```
PROCEDURE tse2d (ptin: point ; var ptout:point2d);
```

La transformation des coordonnées homogènes d'un point en coordonnées cartésiennes est effectué par tsh2d2d:

```
PROCEDURE tsh2d2d (ptin:pointh2d; var ptout:point2d);
```

La procédure ts2dh2d effectue la transformation des coordonnées cartésiennes (dans le plan) d'un point, en coordonnées homogènes :

```
PROCEDURE ts2dh2d (ptin:point2d ; var ptout:pointh2d);
```

La procédure suivante fait correspondre à un point exprimé en coordonnées homogène du plan un point de l'écran ,cette procédure utilise la transformation courante 2D:

```
PROCEDURE tsh2de (ptin:pointh2d; var ptout:point);
```

La procédure tseh2d est la procédure inverse de la précédente , elle utilise l'inverse de la transformation courante dans le plan pour faire correspondre à tout point de l'écran un point du plan utilisateur:

```
PROCEDURE tseh2d (ptin:point ; var ptout:pointh2d);
```

la procédure ts3de fait correspondre à un point exprimé en coordonnées cartésiennes dans le repère objet ,un point écran . Cette procédure utilise la transformation courante 3d issue d'une transformation perspective ou d'une projection orthogonale

```
PROCEDURE ts3de (ptin:point3d ; var ptout: point);
```

La procédure suivante transforme un point dans le repère objet 3D en un point dans le plan exprimé en coordonnées cartésiennes :

```
PROCEDURE ts3d2d (ptin:point3d ; var ptout:point2d);
```

la procédure tsh3de fait correspondre à un point exprimé en coordonnées homogène dans le repère 3D, un point de l'écran en utilisant la valeur de la transformation de visualisation 3d courante. C'est dans cette procédure comme dans ts3de et tsnh3de que la vraie projection s'effectue, rappelant ainsi que la transformation perspective n'est pas la projection perspective:

```
PROCEDURE tsh3de (ptin:pointh3d; var ptout:point);
```

La procédure suivante fait correspondre à un point exprimé en coordonnées écran normalisées homogènes 3D, un point de l'écran:

```
PROCEDURE tsnh3de (ptin:pointh3d; var ptout:point);
```

La procédure tsh3dnh3d permet de transformer les coordonnées homogènes d'un point en coordonnées écran normalisée homogène 3D

```
PROCEDURE tsh3dnh3d (ptin:pointh3d; var ptout:pointh3d);
```

La procédure ts3dh3d transforme des coordonnées cartésiennes 3D d'un point en coordonnées homogènes :

```
PROCEDURE ts3dh3d (ptin:point3d ; var ptout:pointh3d);
```

#### LES FONCTIONS D'ACCES DE DESSIN

Dpt permet l'affichage d'un point exprimé en coordonnées écran

```
PROCEDURE DPT (dot :point);
```

Dpt2d permet l'affichage d'un point exprimé en coordonnées utilisateur 2D:

```
PROCEDURE DPT2D (dot :point2d);
```

L'affichage d'un point exprimé en coordonnées homogènes 2d:

```
PROCEDURE DPTH2D (dot :pointh2d);
```

Dpth3d permet l'affichage d'un point exprimé en coordonnées homogènes 3D:

```
PROCEDURE DPTH3D (dot :pointh3d);
```

La procédure dpt3d dessine un point exprimé en coordonnées cartésiennes 3d

```
PROCEDURE DPT3D (dot :point3d );
```

#### 2. LE MODULE DROITE

Le module droite fait appel aux modules des déclarations globales, module vecteur et module point

#### DEFINITION DES TYPES VISIBLES

La définition d'une droite écran:

```
droite =record orig,extr :point ; end;
```

La définition d'une droite cartésienne dans le plan :

```
droite2d =record orig,extr :point2d ; end;
```

La définition d'une droite homogène

```
droiteh2d =record orig,extr :pointh2d ; end;
```

La définition d'une droite cartésienne dans l'espace:

```
droite3d =record orig,extr :point3d ; end;
```

La définition d'une droite homogène dans l'espace :

```
droiteh3d =record orig,extr :pointh3d ; end;
```

#### LES FONCTIONS D'ACCES D' AFFICHAGE

L'affichage d'une droite écran :

```
PROCEDURE AFDROITE (seg :droite );
```

L'affichage d'une droite du plan dont les extrémités sont exprimées en coordonnées cartésiennes:

```
PROCEDURE AFDROITE2D (seg :droite2d );
```

L'affichage d'une droite du plan dont les extrémités sont exprimées en coordonnées homogènes :

```
PROCEDURE AFDROITEH2D (seg :droiteh2d );
```

L'affichage d'une droite de l'espace 3D dont les extrémités sont exprimées en coordonnées cartésiennes :

```
PROCEDURE AFDROITE3D (seg :droite3d );
```

L'affichage d'une droite de l'espace 3d dont les extrémités sont exprimées en coordonnées homogènes:  
 PROCEDURE AFDROITEH3D (seg :droiteh3d );

### 3. LE MODULE VECTEUR

Le module vecteur fait appel aux déclaration globale dans le module glob.

#### DEFINITION DES TYPES VISIBLES

On définit un vecteur écran par ses coordonnées ,on l'assimile à un point écran.

vecteur = point;

On assimile un vecteur dans le plan cartésien par à point dont les coordonnés sont exprimées dans le repère cartésien 2D

vecteur2d =point2d;

Le vecteur homogène dont le plan est exprimé en coordonnées homogènes

vecteurh2d = pointh2d;

Un vecteur3D est un vecteur dont les coordonnées sont exprimées dans le repère cartésien 3d, on l'assimile alors à un point3d.

vecteur3d =point3d;

Un vecteurh3d est un vecteur de l'espace 3d dont les coordonnées sont homogènes

vecteurh3d = pointh3d;

#### FONCTIONS D'ACCES

#### DEFINITION

On définit un vecteur écran par ses deux extrémités, exprimées en coordonnées écran

PROCEDURE DEFVECT (pt1,pt2 :point ;var vec :vecteur);

La procédure suivante définit un vecteur du plan cartésien à partir de ses deux extrémités, exprimées en coordonnées cartésiennes:

PROCEDURE DEFVECT2D (pt1,pt2 :point2d ;var vec:vecteur2d );

La procédure defvecth2d est semblable à defvect2d mais en cas de coordonnées homogènes:

PROCEDURE DEFVECTH2D (pt1,pt2:pointh2d ;var vec:vecteurh2d)

Définition d'un vecteur 3d à partir de ses extrémités exprimées en coordonnées cartésiennes :

PROCEDURE DEFVECT3D (pt1,pt2 :point3d ;var vec:vecteur3d );

Cette procédure est similaire à la précédente dans le cas des coordonnées homogènes :

PROCEDURE DEFVECTH3D (pt1,pt2 :pointh3d;var vec:vecteurh3d)

#### DEFINITION D'UN VECTEUR NORMAL

VNORMAL3D définit le vecteur normal à deux vecteur données:

PROCEDURE VNORMAL3D(vec1,vec2:vecteur3d;var vec:vecteur3d);

#### CALCUL DE NORME

Chacune de ces procédures calcule la norme d'un vecteur selon le repère choisi:

PROCEDURE NORME (vec:vecteur ;var dis : real);  
 PROCEDURE NORME2D (vec:vecteur2d ;var dis : real);  
 PROCEDURE NORMEH2D (vec:vecteurh2d ;var dis : real);  
 PROCEDURE NORME3D (vec:vecteur3d ;var dis : real);

#### TRANSFORMATION D'UN VECTEUR

Cette procédure transforme les coordonnées homogènes d'un point en appliquant une transformation 2d:

PROCEDURE VTRANS2D (var pt2:pointh2d; t:transfh2d  
 ;pt1 :pointh2d);

Vtransh3d effectue la même opération pour un point de l'espace exprimé en coordonnés homogènes :

PROCEDURE VTRANS3D (var pt2:pointh3d; t:transfh3d  
 ;pt1 :pointh3d);

#### 4. LE MODULE MATRICE

Ce module fait appel au module graphique de turbo4(graph) et au module des déclarations globales :

La procédure suivante effectue une multiplication matricielle de deux matrices carrées d'ordre 3. Ces deux matrices sont la représentation homogène de deux transformations dans le plan ; la multiplication traduit la composition des deux transformations et produit une matrice carrée d'ordre 3, qui représente la transformation résultante de la composition.

```
PROCEDURE MULTMATH2D (var res:transfh2d; m1,m2:transfh2d);
```

La procédure multmath3d est similaire à multmath2d dans le cas de matrices carrées d'ordre 4, représentant des transformations dans l'espace :

```
PROCEDURE MULTMATH3D (var res:transfh3d; m1,m2:transfh3d);
```

Les procédures invmath2d et invmath3d produisent les matrices inverses dans le cas de matrice d'ordre 3 et 4 :

```
PROCEDURE INVMATH2D (var res:transfh2d; m :transfh2d);
PROCEDURE INVMATH3D (var res:transfh3d; m :transfh3d);
```

#### 5. LE MODULE RECTANGLE

Le module rectangle fait appel aux déclarations globales du module glob et à certaines fonctions d'accès des modules vecteurs et points :

DEFINITIONS DES TYPES VISIBLES

On définit le rectangle écran par les extrémités d'une de ses diagonales exprimées en coordonnées écran. Le type rect sert à définir les variables fenêtres:

```
rect = record min,max :point ; end;
```

Le rect2d est un rectangle dans le plan utilisateur. Les extrémités des diagonales qui définissent le rectangle sont exprimées en coordonnées cartésiennes:

```
rect2d = record min,max :point2d ; end;
```

La définition du recth2d diffère par le fait que les extrémités sont exprimées, non pas en coordonnées cartésiennes, mais en coordonnées homogènes:

```
recth2d = record min,max :pointh2d; end;
```

FONCTION D'ACCES

Les procédures ordrect et ordrect2d vérifient si le rectangle en question a été défini dans l'ordre, c'est à dire si le "min" représente le point aux plus faibles coordonnées et si le "max" représente le point aux coordonnées les plus importantes. Les deux procédures rectifient la définition du rectangle en cas d'erreur, respectivement dans le cas de rectangle écran et dans le cas de rectangle dans le plan utilisateur:

```
PROCEDURE ORDRECT (var r: rect);
PROCEDURE ORDRECT2D (var r: rect2d);
```

Fcentre est une procédure qui définit les caractéristiques centrales d'une fenêtre, elle permet de caractériser la fenêtre par son centre et par ses demi-tailles:

```
PROCEDURE FENCENTRE (f: rect; var fc : centralrectangle);
```

La fonction clipfenet vérifie si la fenêtre est à l'intérieur de l'écran. Si la fenêtre dépasse l'écran, clipfenet la découpe et rectifie ses coordonnées. Boolean =faux quand la fenêtre est totalement en dehors de l'écran :

```
FUNCTION CLIPFENET (var r: rect):boolean;
```

LES FONCTIONS D'ACCES D'AFFICHAGE

Les procédures arect, arect2d et arecth2d dessinent des rectangles dont les extrémités sont exprimées respectivement dans le repère écran, dans le repère cartésien du plan utilisateur et dans le repère homogène :

```
PROCEDURE ARECT (r: rect);
PROCEDURE ARECT2D (r: rect2d);
PROCEDURE ARECTH2D (r: recth2d);
```

## FONCTIONS D'ACCES : FENETRAGE ET DECOUPAGE

La procédure setviewport est fournie par turbo4 ,elle permet de définir une fenêtre et indique si l'on active le découpage :

```
PROCEDURE SETVIWPORT(x1, y1, x2, y2 :word; limite :boolean);
```

Setcloture affecte la nouvelle clôtüre "nclôtüre" à la variable globale clôtüre , nclôtüre devient la clôtüre courante:

```
PROCEDURE SETCLOTURE (ncloture : rect2d);
```

6. LE MODULE TRANSFORMATION

Le module transformation fait appel aux modules graph, glob, outils, vecteur, matrice, point et rectangle:

## DEFINITION DES TYPES VISIBLES

Le type virtualcamera a été défini, suite à l'adoption du modèle de la boîte noire pour étudier les projections perspectives . Il est défini par la position de l'observateur ,la position du point de visé ,le pivotement et l'angle d'ouverture :

```
VIRTUALCAMERA = RECORD
    observer :point3d ;
    lookat   :point3d ;
    spin     :real    ; (*pivotement
                        :angle en
                        degrés*)
    aperture :real    ; (*angle
                        d'ouverture*)
END;
```

## FONCTION D'ACCES

La procédure prisevue2d définit la transformation de visualisation courante en deux dimensions, à partir de la fenêtre courante,la clôtüre courante et la composition avec une autre transformation-si nécessaire:

```
PROCEDURE PRISEVUE2D (NCLOTURE: rect2d ;
    NFENETRE : rect ;
    TF : transfh2d
);
```

Pour la définition de transformation perspective ,la procédure prisevue3d utilise comme donnée :une caméra,une profondeur de champ

et une fenêtre . Cette procédure met à jour la variable globale currenttrans3d correspondant à la transformation courante 3d :

```
PROCEDURE PRISEVUE3D (CAMERA : virtualcamera ;
    DEPTH : real ;
    NFENETRE : rect
);
```

Dans le cas des projections orthogonales ,l'observateur est toujours placé à l'infini sur l'axe principale correspondant à celui de la projection .Les procédures correspondantes ne sont donc pas paramétrées . Comme la procédure prisevue3d, elles mettent à jour la transformation de visualisation courante en lui affectant la valeur de la matrice de projection orthogonale selon ,l'axe choisi.

Les procédures de projections orthogonales sont les suivantes :

-PROCEDURE VUEDEFACE :Projection orthogonale selon l'axe des x,l'observateur est placé à plus l'infini.La position de l'observateur est très importante pour l'élimination des faces cachées.

-PROCEDURE VUEDEDERRIERE : Projection orthogonale selon l'axe des x,l'observateur étant placé à moins l'infini.

-PROCEDURE VUEDEDROITE : Projection orthogonale selon l'axe des y , l'observateur est placé à plus l'infini ;

-PROCEDURE VUEDEGAUCHE :Projection orthogonale selon l'axe des y,l'observateur est placé à moins l'infini

-PROCEDURE VUEDEDESSUS :Projection orthogonale selon l'axe des z, l'observateur est placé à plus l'infini

## 7. LE MODULE POLYGONE

UNIT POLYGONE;

INTERFACE

Le module polygone utilise l'interface des modules suivants  
:graph, glob, outils, vect, pt, droit, rectangl.

DEFINITION DES CONSTANTES VISIBLES

```
NB10 = 10;
nb50 = 50;
nb100 = 100;
nb150 = 150;
```

DEFINITION DES TYPES VISIBLES

```
nbt = (n10, n50, n100, n150);
```

Les constantes nb10, nb50, nb100 et nb150 ainsi que le type nbt vont servir à définir la structure de données qui correspond à un polygone, sous forme d'un tableau.

Le type tabpoly répond au fait que les procédures de dessin et de remplissage de polygones fournies par turbo4, ne reconnaissent que des tableaux de points.

L'idée d'utiliser le type enregistrement avec partie variable est intéressante pour satisfaire la contrainte de mémoire. En effet il s'agit d'avoir la possibilité de définir à chaque fois la taille du tableau, selon le nombre de points.

```
tabpoly = record
  nb : word;
  case tab : nbt of
    n10 : (tab10 : array[1..nb10] of pointtype);
    n50 : (tab50 : array[1..nb50] of pointtype);
    n100 : (tab100 : array[1..nb100] of pointtype);
    n150 : (tab150 : array[1..nb150] of pointtype);
  end;
```

On peut définir une autre structure de données pour les polygones : il s'agit des listes chaînées. Ces listes sont construites par allocation dynamique et ont l'avantage de permettre l'accès au contenu d'un élément par l'intermédiaire de son adresse. Les listes chaînées sont

souvent moins coûteuses en place mémoire que les tableaux. On peut définir ainsi les listes chaînées suivantes :

Liste chaînée de pointtypes (points enregistrements):

```
recpoly = ^recsom;
recsom = record
  pt : pointtype;
  sv : recpoly;
end;
```

Liste chaînée de points exprimés en coordonnées écran:

```
poly = ^som ;
som = record
  pt : point;
  sv : poly ;
end;
```

liste chaînée de points du plan utilisateur exprimés en coordonnées cartésiennes:

```
poly2d = ^som2d ;
som2d = record
  pt : point2d;
  sv : poly2d ;
end;
```

liste chaînée de points du plan utilisateur exprimés en coordonnées homogènes:

```
polyh2d = ^somh2d ;
somh2d = record
  pt : pointh2d;
  sv : polyh2d ;
end;
```

Liste chaînée de points de l'espace 3d exprimés en coordonnées homogènes:

```
polyh3d = ^somh3d ;
somh3d = record
  pt : pointh3d;
  sv : polyh3d ;
end;
```

Liste chaînée de points de l'espace exprimés en coordonnées cartésiennes:

```

poly3d    = ^som3d ;
som3D     = record
            pt : point3d;
            sv : poly3d ;
        end;

```

On peut définir un polygone par le nombre de points de contour et par la liste chaînée contenant ces points:

Recpolyg définit un polygone de n sommets dont les coordonnées sont exprimées dans le repère écran sous forme d'enregistrement:

```

recpolyg = record
            p: recpoly;
            n: integer;
        end;

```

Polyg définit un polygone de n sommets dont les coordonnées sont exprimées dans le repère écran sous forme de points:

```

polyg     = record
            p: poly;
            n: integer;
        end;

```

Polyg2d définit un polygone de n sommets dans le plan utilisateur et dont les coordonnées sont cartésiennes:

```

polyg2d   = record
            p: poly2d;
            n: integer;
        end;

```

polyg3d définit un polygone de n sommets de l'espace 3d et dont les coordonnées sont cartésiennes:

```

polyg3d   = record
            p: poly3d;
            n: integer;
        end;

```

polygh2d et polygh3d définissent des polygones dont les coordonnées des points sont homogènes:

```

polygh2d  = record
            p: polyh2d;
            n: integer;
        end;

polygh3d  = record
            p: polyh3d;
            n: integer;
        end;

```

La fonction valeurnbt renvoie comme résultat, la taille du tableau à définir pour enregistrer le polygone de i points

```

function valeurnbt(i:integer): nbt;

```

Les procédures tsrecpota et tspolytab sont des procédures de conversion, elles transforment une structure de données de type liste chaînée en un tableau contenant les mêmes points. c'est un changement du contenant et non du contenu

La procédure tsrecpota transforme une liste chaînée contenant un polygone de pointtype, en un tableau de pointtype :

```

PROCEDURE TSRECPOTA (rpg : recpolyg; var ptab: tabpoly);

```

La procédure tspolytab transforme un polygone de points, en un tableau de pointtype:

```

PROCEDURE TSPOLYTAB (pg : polyg ; var ptab: tabpoly);

```

La procédure tspoly2de transforme une liste chaînée en une liste chaînée. En effet elle transforme les points de la première liste exprimés en coordonnées cartésiennes du plan en des points en coordonnées écran. Cette opération s'effectue en appliquant la transformation de visualisation courante dans le plan :

```

PROCEDURE TSPOLY2DE (pg2 : polyg2d ; var pg : polyg);

```

La procédure tspoly3de a le même effet que tspoly2de dans le cas de point3d, on appelle la transformation de visualisation courante 3d :

```

PROCEDURE TSPOLY3DE (pg3 : polyg3d ; var pg : polyg);

```

Les procédures suivantes permettent à l'utilisateur de rentrer par l'intermédiaire du clavier avec écho sur écran les coordonnées des points sommets du polygone. Le système des coordonnées des points et la structure de données (dans laquelle les points sont enregistré)s varient en fonction des procédures:

```

PROCEDURE SRECPOLY (var pg: recpolyg);
PROCEDURE STABPOLY (var ptab : tabpoly);
PROCEDURE SAISIPOLYG (var pg: polyg );
PROCEDURE SPOLYG2D (var pg: polyg2d);

```

Les procédures suivantes permettent de dessiner le contour du polygone:

Aftpoly dessine le contour de polygone dont les points sont de type pointtype exprimés en coordonnées écran et stockés dans un tableau:

```
PROCEDURE AFTPOLY (ptab : tabpoly);
```

Dans le cas de afrpoly, les points sont exprimés en coordonnées écran et sont stockés dans une liste chaînée:

```
PROCEDURE AFRPOLY (rpg : recpolyg);
```

Les polygones dans afpoly,afpoly2d et afpoly3d sont stockés dans des listes chaînées.

```
PROCEDURE AFPOLY (pg : polyg );
PROCEDURE AFPOLY2D (pg : polyg2d );
PROCEDURE AFPOLY3D (pg : polyg3d );
```

Les procédures de remplissage de polygone sont semblables à celles du dessin de contour:

```
PROCEDURE REMTPOLY (ptab : tabpoly);
PROCEDURE REMRPOLY (rpg : recpolyg);
PROCEDURE REMPOLY (pg : polyg );
PROCEDURE REMPOLY2D (pg : polyg2d );
PROCEDURE REMPOLY3D (pg : polyg3d );
```

## 8. LE MODULE VOLUME

Le module volume utilise les interfaces des modules graph, glob, outils, vect, pt, droit, rectangl, polygone:

La structure de données concernant les volumes est très importante pour les algorithmes de saisie, d'affichage et d'élimination des parties cachées. Cette structure dépend de la façon choisie pour modéliser les volumes. Le choix de cette structure repose sur le fait qu'on modélise le volume non seulement par la liste chaînée des sommets, mais en plus par une liste des arêtes et une autre liste des faces. Dans la liste des arêtes, on indique pour chacune des extrémités d'une arête l'adresse du point dans la liste des sommets. Dans la liste des faces, on construit pour chaque face une liste comportant les adresses (dans la liste des sommets) des points formants le contour de la face.

## DEFINITION DES TYPES VISIBLES

```
name = string[20];
```

On caractérise un plan par les quatre coefficients correspondant à son équation  $ca X + cb Y + cc Z + cd = 0$  :

```
planh3d = record
    ca : real; (* ca,cb,cc : vecteur normal dirigé vers
                l'intérieur*)
    cb : real;
    cc : real;
    cd : real; (* distance entre le plan et l'origine*)
end ;
```

La liste chaînée des sommets du volume :

```
sompotr = ^somet;
somet = record
    somsuiv : somptr;
    som : pointh3d;
end;
```

Arptr est la liste chaînée des arêtes du volume. Les éléments de cette liste ne comportent pas les coordonnées des points origine et extrémité mais simplement leur adresse dans la liste des sommets du volume. C'est par l'intermédiaire de ces adresses qu'on peut accéder aux coordonnées des points.

```
arptr = ^arret;
arret = record
    arsuiv : arptr;
    orig,extr : somptr;
end;
```

On définit ici la liste des adresses des sommets d'une face. Comme pour les arêtes, cette liste ne comporte que les adresses des points dont les coordonnées sont stockées dans la liste des sommets seulement- aux adresses indiquées dans la liste somfptr-

```
somfptr = ^somface;
somface = record
    fsom : somptr;
    fsomsv : somfptr;
end;
```

Le type extrem permettra de définir un tableau dans lequel on stockera pour une face ses coordonnées extrêmes:

```
extrem = (xmin,xmax,ymin,ymax,zmin,zmax);
```

Finalement ,on définit une face par un enregistrement de plusieurs champs . Le faceptr est le pointeur sur la face, c'est à dire son adresse dans la liste des faces.

Pour chaque face on indique les éléments suivants:

-Facesv :Adresse de la face suivante dans la liste des faces du volume.

-Polyf :La liste (définie par l'adresse de son premier élément) des adresses des sommets de la face.

-Fcol :La couleur de la face.

-Fsty :Le style de remplissage du volume.

-Nvect :Le vecteur normal à la face.

-Fplan :L'équation du plan de la face.

-numface:Le numéro de la face dans la liste des faces.

-nb :Le nombre de points définissant le contour de la face.

-échange:Une valeur logique indiquant si la position de la face dans la liste des faces a été changée, pour des problèmes d'élimination des surfaces cachées:

```
faceptr = ^face;
face = record
    facesv : faceptr;
    polyf : somfptr;
    fcol : word; (*colorindex*)
    nvect : vecteurh3d;
    fplan : planh3d;
    limit : array[extrem] of real ;
    numface : word;
    nb : word;
    échange : boolean;
end;
```

On définit ici une chaîne listée de volume . Volptr est le pointeur correspondant . Chaque volume de la liste est un enregistrement comportant les champs suivants:

-Vsom :La liste des sommets du volume .

-Arvol :La liste des arêtes du volume.

-Vface :La liste des faces du volume.

-Arcol :La couleur des arêtes.

-Volsv :L'adresse du volume suivant, dans la liste des volumes.

```
volptr = ^volh3d;
volh3d = record
    arcol : word; (*colorindex*)
    volsv : volptr;
    vsom : somptr;
    arvol : arptr;
    vface : faceptr;
end;
```

Ainsi on peut décomposer l'objet en plusieurs volumes:

```
objetptr = ^ objet;
objet = record
    objetsv : objetptr;
    listvol : volptr;
end;
```

Le choix de la structure de données-volume est détaillé dans le paragraphe modélisation des volumes ;un schéma d'illustration a été donné dans le même paragraphe .

Les fonctions d'accès de saisie de volume sont svolume,sficvol et saifich.

La procédure svolume permet une saisie du volume, directement dans la structure de donnée volume . Cette façon de saisir est moins lourde mais ne permet pas d'avoir un enregistrement physique pour conserver les données du volume.Les points enregistrés sont exprimés en coordonnées objet:

```
PROCEDURE SVOLUME (var vol : volptr);
```

La procédure sficvol permet de saisir les données du volume dans un fichier de point.Chaque face commence par un point fictif et on enregistre pour chaque face les coordonnées de tous les points contours de la face. La saisie se passe comme pour la procédure précédente, en rentrant les coordonnées par le clavier. L'inconvénient de cette méthode est qu'elle n'enregistre que les valeurs des points, mais pas les autres caractéristiques des faces (la couleur ,le style de remplissage,le numéro de la face et le nombre de points):

PROCEDURE SFICVOL (var fc : name );

La procédures saifich permet de saisir tous les points de chaque face du volume, en plus des autres caractéristiques telles que la couleur ,le style de remplissage ,le numéro de la face et le nombre des points.:

PROCEDURE SAIFICH (var fc : name );

Les fonctions d'accès de transfert sont fcnh3d et tsfsdv

La fonction fcnh3d permet la saisie de volume en coordonnées nh3d dans une structure de donnée volume ,à partir d'un fichier de pointh3d saisie, par l'intermédiaire de sficvol: Il s'agit d'un parcours séquentiel de ce fichier dans lequel on traite des sous séquences qui sont les faces .

On effectue sur tous les points d'une face, une transformation en coordonnées écran normalisées homogène, selon la transformation perspective courante et on applique le découpage à toutes les arêtes ,selon le volume de visualisation défini ; on ne retient que les points résultants du découpage .

On tient compte dans ce parcours du point fictif au début de chaque face, et à la fin du fichier, ce qui correspond au début d'une face vide qui finit le volume.

PROCEDURE FCNH3D (fc :name ; var v : volptr);

La procédure tsfsdv réalise le transfert des points à partir du fichier saisie par saifich à la structure de donnée volume ,sans traitement de points . On obtient ainsi un volume dont les points sont définis en coordonnées objet. C'est un parcours séquentiel semblable à celui de fcnh3d ,mais sans traitement des points ,ni de découpage .

Dans ce parcours on doit prendre en compte que chaque face débute par un point fictif et que le deuxième point n'indique pas des vraies coordonnées d'un point, mais respectivement le numéro de la face ,le style,la couleur et le nombre de point

PROCEDURE TSFSDV(fc:name; var vh3d:volptr);

La procédure tsortho effectue un parcours séquentiel de la chaîne des sommets du volume au cours duquel, le traitement consiste en la transformation des coordonnées des points, en utilisant la matrice currenttrans3d (qui est la matrice définie par une des procédures de projection orthogonale ou par la procédure de projection perspective prisevue3d). La valeur de Z n'a qu'une valeur relative ,elle est plus importante quand on s'éloigne de l'observateur .

PROCEDURE TSORTHO (var v:volptr);

Les procédures d'affichage sont fildeferh3d, fildefernh3d, afvolumnh3d, affxortho , affyortho et affzortho .

Les procédures fildefer permettent d'afficher tout les arêtes du volume .Dans le cas de fildeferh3d, les points du volume sont en coordonnées objet , mais dans le cas de fildefernh3d les points ont déjà été transformés en coordonnées écran normalisées homogènes:

PROCEDURE FILDEFERH3D (vh3d : volptr);  
PROCEDURE FILDEFERNH3D (vnh3d : volptr );

La procédure afvolumenh3d permet d'afficher avec remplissage de taches un volume dont les coordonnées des points ont déjà été transformées en coordonnées écran normalisées

PROCEDURE AFVOLUMENH3D (vnh3d : volptr);

Les procédures affxortho,affyortho et affzortho permettent d'afficher un volume avec une projection orthogonale selon l'axe des x ,l'axe des y et l'axe des z . Ces procédures font également appel à la transformation de visualisation 2d définie par la donnée d'une clôture dans le plan de projection et par une fenêtre de l'écran.

PROCEDURE AFFXORTHO(var v:volptr;f:viewporttype;  
ymin,zmin,ymax,zmax :real);  
PROCEDURE AFFYORTHO(var v:volptr;f:viewporttype;  
xmin,zmin,xmax,zmax :real);  
PROCEDURE AFFZORTHO(var v:volptr;f:viewporttype;  
xmin,ymin,xmax,ymax :real);

### 9. LE MODULE D'ELIMINATION DE SURFACES CACHEES

Le module des surfaces cachées utilise l'interface des modules suivants :glob, outils, vect, pt, rectangl, matrice, transformation, droit, polygone et volumes

La liste de description est la liste non ordonnée des faces du volume . La procédure produirelv ordonne cette liste de façon à ce qu'une face p soit placée avant une face q, si p ne cache pas q . La liste ainsi ordonnée est appelée liste de visualisation . La procédure volpeintre produit la liste de visualisation correspondant à une liste de faces d'un volume.

```
PROCEDURE PRODUIRELV(var listf:faceptr);
PROCEDURE VOLPEINTRE(var vpeint:volptr);
```

### 10. LE MODULE DES DECLARATIONS GLOBALES

Dans ce module, on définit les types et les variables globales . Ils sont accessibles à partir de tout les autres modules . Le nom de ce module est glob et ne présenté pas de fonction d'accées, l'interface est donc formé uniquement par certains types et variables représentants certain objets tels que point, fenêtre, matrice.

#### LA DEFINITION DES TYPES VISIBLES

La définition des systèmes de coordonnées:

```
coord      = (w,x,y,z);    (* les 4 coordonnées*)
coord2d    = x..y;
coordh2d   = w..y;
coord3d    = x..z;
coordh3d   = w..z;
point      = array[coord2d] of integer;
```

Les coordonnées écran:

```
point2d    = array[coord2d] of real;
```

Les coordonnées cartésiennes dans le plan:

```
pointh2d   = array[coordh2d] of real;
```

Les coordonnées cartésiennes dans le repère objet 3D:

```
point3d    = array[coord3d] of real;
```

Les coordonnées homogènes dans l'espace 3D :

```
pointh3d   = array[coordh3d] of real;
```

La définition d'une fenêtre écran:

```
rect       = record min,max : ptword; end;
```

La définition d'une fenêtre dans le plan utilisateur(clôture)est la suivante:rect2d = record min,max : point2d ;end;

La définition d'une fenêtre en coordonnées homogènes dans le plan: recth2d = record min, max : pointh2d ;end;

La définition d'une matrice homogène 2D:

```
transfh2d  = array[coordh2d,coordh2d] of real;
```

La définition d'une matrice homogène 3d:

```
transfh3d  = array[coordh3d,coordh3d] of real;
```

La caractérisation d'une fenêtre par son centre et ses deux demi-tailles:

```
centralrectangle2d = record
    . center : point2d;
    . halfsize : point2d;
end;
```

caractérisation d'une clôture par son centre et ses deux demi-tailles:

```
centralrectangle = record
    center : point;
    halfsize : point;
end;
```

#### LA DEFINITION DES VARIABLES VISIBLES

La variable globale ayant pour valeur la clôture courante:

```
clôture    : rect2D;
```

La variable globale ayant pour valeur la fenêtre courante:

```
fenêtre    : rect;
```

La caractérisation de la clôture courante par le centre et les deux demi-tailles:

centralviewport : centralrectangle;

La variable globale ayant comme valeur la matrice de la transformation courante dans le plan :

currenttrans2d : transfh2d ;

L'inverse de la matrice de la transformation courante , permet de transformer des coordonnées écran en coordonnées plan:

invcurrenttrans2d : transfh2d ;

La matrice de la transformation courante en 3 dimension :

currenttrans3d : transfh3d ;

La valeur de la couleur courante :

couleur : word;

#### LA DEFINITION DES CONSTANTES VISIBLES

La matrice homogène nulle dans plan:

```
matnulh2d:transfh2d =( (0,0,0),
                       (0,0,0),
                       (0,0,0)
                       );
```

La matrice homogène identité dans le plan:

```
matideh2d:transfh2d =( (1,0,0),
                       (0,1,0),
                       (0,0,1)
                       );
```

La matrice homogène identité dans l'espace:

```
matideh3d:transfh3d =( (1,0,0,0),
                       (0,1,0,0),
                       (0,0,1,0),
                       (0,0,0,1)
                       );
```

La matrice homogène nulle dans l'espace:

```
matnulh3d:transfh3d =( (0,0,0,0),
                       (0,0,0,0),
                       (0,0,0,0),
                       (0,0,0,0)
                       );
```

);

#### L'IMPLEMENTATION

Dans ce module il n'y ni fonction d'accès ni procédure interne . La partie implantation est vide.

#### 11. LE MODULE DES OUTILS

unit outils;

#### INTERFACE

Le module outils n'est pas un module graphique ,son rôle est de fournir des procédures d'acquisition de données sur écran et de leur affichage .

#### DEFINITION DES VARIABLES VISIBLES

tampx, tampy : integer;

tampx,tampy servent à gérer la position du curseur

#### LES FONCTIONS D'ACCES

les procédures avpte, avpt2d, avpth2d, avpt3d, avpth3d permettent respectivement l'acquisition des points, point2d , pointh2d , point3d et pointh3d . Les procédures afpte, afpt2d ,afpth2d , afpt3d , afpth3d affichent respectivement des points, point2d ,pointh2d,point3d, pointh3d :

```
PROCEDURE avpte( var xx,yy: integer ;var f:boolean);
PROCEDURE afpte(pt:point);
PROCEDURE avpt2D( var xx,yy: real ;var f:boolean);
PROCEDURE afpt2d(pt:point2d);
PROCEDURE avpth2d( var xx,yy,ww: real ;var f:boolean);
PROCEDURE afpth2d(pt:pointh2d);
PROCEDURE avpt3d( var xx,yy,zz: real ;var f:boolean);
PROCEDURE afpt3d(pt:point3d);
PROCEDURE avpth3d( var xx,yy,zz,ww: real ;var f:boolean);
PROCEDURE afpth3d(pt: pointh3d);
```

Les procédures suivantes saisissent et affichent des matrices d'ordre 3 et d'ordre 4:

```
PROCEDURE STRANSFH2D( var mat:transfh2d; var f: boolean);
PROCEDURE STRANSFH3D( var mat:transfh3d; var f: boolean);
PROCEDURE AFMATH2D(mat:transfh2d);
PROCEDURE AFMATH3D(mat:transfh3d);
```

## VIII. LA MODELISATION GEOMETRIQUE DES DONNEES

Dans le chapitre précédent , nous avons vu que certains modules notamment le module volume et le module de l'élimination des surfaces cachées ,reposent sur une structure de donnée traduisant la modélisation géométrique choisie . Le choix de cette modélisation va être détailler dans ce chapitre.

Egalement , comme on l'a déjà signalé , nous avons vu que la bibliothèque réalisée ne comporte que des modules de modélisation et de visualisation . Ces modules ne traduisent donc que les objets graphiques et certains objets de modélisations . Des objets comme le maillage ,la triangulation, la carte n'ont pas été implantés . Ces objets sont le sujet de développement algorithmiques dans ce chapitre.

### A. LA MODELISATION DES VOLUMES

#### 1. SURFACE PLANE - SURFACE MATHEMATIQUE

On a choisi de modéliser les volumes à partir des facettes planes et non à partir des surfaces mathématiques ;ce choix est justifié par les raisons suivantes :

-La modélisation à partir des facettes planes est plus facile à mettre en oeuvre , occupe moins de mémoire et nécessite moins de calcul.

-l'élimination des parties cachées est beaucoup plus simple dans le cas des facettes planes , cependant l'algorithme d'élimination des faces cachées sélectionné (N.N.S) est facilement adaptable aux surfaces mathématiques .

## 2. LA MODELISATION

### a) LE PRINCIPE DE LA MODELISATION

Un volume peut être donné par

- ses sommets
- ses arêtes
- ses faces

La modélisation d'un volume par la simple liste de ses sommets est insuffisante car, rien n'indique comment ils sont reliés pour définir les surfaces du volume.

La représentation par les arêtes n'est pas non plus suffisante, parce qu'elle ne permet qu'un dessin du squelette filiforme de l'objet. Elle ne comporte donc aucune information sur l'élimination des parties cachées.

Nous avons choisi une représentation dans laquelle les coordonnées des sommets ne sont pas données plusieurs fois mais où l'on décrit chaque arête par des pointeurs sur les sommets (représentation implicite).

On définit chacune des faces par une liste des sommets, en utilisant la même notation indirecte. C'est à dire qu'on ne répétera pas les coordonnées des sommets mais qu'on utilisera des pointeurs.

Cette représentation présente les avantages suivants:

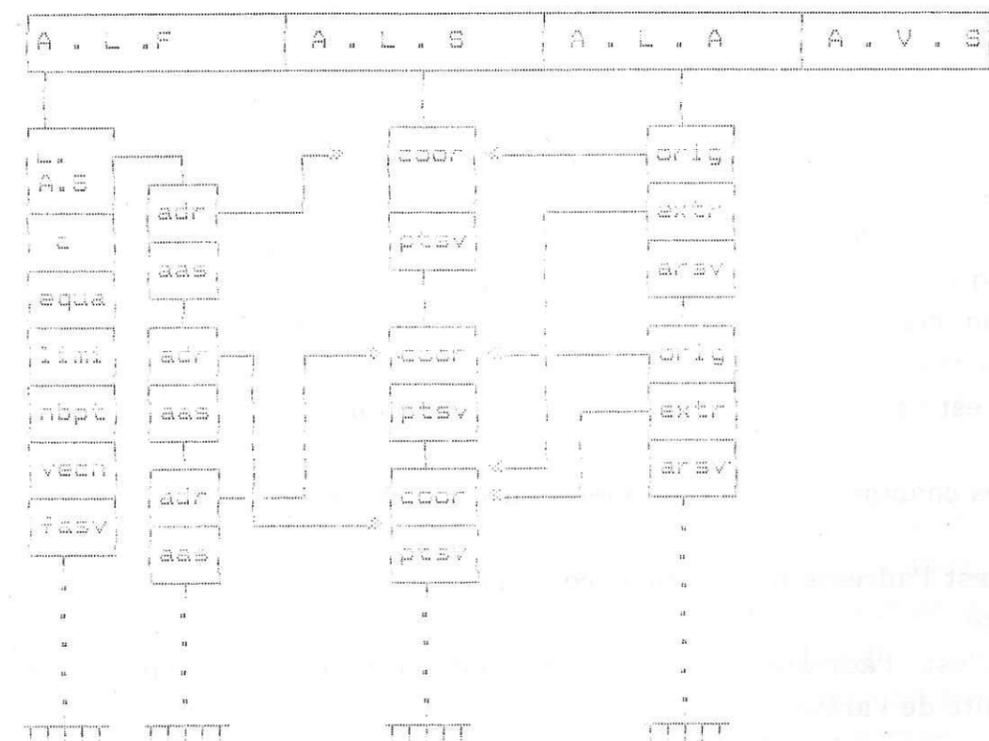
- Une transformation ne sera appliquée qu'une seule fois, à chaque sommet.
- Un sommet ne peut pas diverger en deux triplets différents de coordonnées suite à l'accumulation d'erreurs d'arrondies. Ces derniers empêcheraient l'identification de tous les arêtes aboutissant à un même sommet.

Cette représentation permet aussi de modifier facilement le volume en déplaçant les sommets.

### b) LE SCHEMA

La modélisation dont le principe a été décrit ci-dessus peut être représenté par le schéma suivant:

#### VOLUME



Les champs constituant l'enregistrement volume sont:

- A.L.F : Adresse de la liste des faces,
- A.L.S : Adresse de la liste des sommets,
- A.L.A : Adresse de la liste des arêtes

Les champs constituant une face sont:

- L.A.S : C'est l'adresse de la liste des adresses des sommets, chaque élément de cette liste est constitué par les champs suivants :

- adr : C'est l'adresse dans laquelle les coordonnées du point sont stockées.

- aas : C'est l'adresse de l'adresse suivante
- C : C'est la couleur de remplissage de la face,
- equa: Dans ce champ, on stocke les coefficients du plan de la face,
- limit : C'est un tableau dans lequel on conserve les coordonnées des sommets extrêmes de la face ,
- vecn : Se sont les coordonnées du vecteur normal à la face
- nbpt : C'est un entier indiquant le nombre de points formant le contour de la face
- fasv : c'est l'adresse de la face suivante dans la liste des faces.

Un élément de la liste des sommets est constitué par les champs suivants:

- coor: Ce sont les coordonnées du sommet . C'est uniquement dans ce champ qu'on enregistre les coordonnées des points ; dans les autres listes, on manipule les adresses seulement là où les coordonnées des points sont enregistrées,
- ptsv: C'est l'adresse de l'élément suivant dans la liste

Les champs constituant un élément de la chaîne des arêtes sont :

- orig: C'est l'adresse de l'élément contenant les coordonnées de l'origine de l'arête,
- extr: C'est l'adresse de l'élément contenant les coordonnées de l'extrémité de l'arête .

### 3. LA REALISATION DE LA MODELISATION

Nous avons réalisé trois versions différentes de saisie et de modélisation de volume:

- La première version consiste à saisir par allocation dynamique, les sommets (exprimés en coordonnées objet) de chacune des faces , directement dans la structure de donnée-volume qu'on vient de détailler. Cette version est réalisée par la procédure SVOLUME (voir le module volume dans le dossier de réalisation ) . La procédure SVOLUME permet une saisie qui n'est pas coûteuse en nombre d'opérations. Mais elle présente l'inconvénient de ne pas permettre l'enregistrement physique du volume (par exemple dans un fichier de points ) de façon à pouvoir le reconstituer ,le traiter et l'afficher sous différents points de vue, à tout instant.

-La deuxième version réalisée permet la saisie et la modélisation en deux étapes:

\* La première : Il s'agit de saisir dans un fichier de points ,les coordonnées homogènes des sommets des faces, exprimées dans le repère objet . Cette étape est réalisée par la procédure saifich et permet le stockage du fichier sur un support physique .

\* La deuxième : Elle consiste à stocker les sommets de chacune des faces récupérées à partir du fichier ,dans la structure de donnée-volume. Il s'agit de transférer les points du fichier dans une structure de donnée-volume . Cette étape est réalisée par la procédure TSFSDV (voir l'unit volume dans le chapitre 5).Les sommets sont exprimés en coordonnées objet homogènes et ne subissent aucune transformation .

Cette version produit donc pour chaque volume un fichier sur un support physique qu'on peut garder ,et une structure de donnée-volume qui servira pour les traitements et les visualisations ultérieures. Cette structure disparaîtra à la fin d'une séance de travail.

-La troisième version est également réalisée en deux étapes:

\*La première : saisir -comme dans la deuxième version - les sommets des faces dans un fichier de points et le stocker sur un support physique . Cette étape est réalisée par la procédure SFICVOL

\*La deuxième : Transférer les points du fichier dans une structure de donnée-volume après avoir effectué la transformation perspective 3D normalisée et le découpage selon le volume de visualisation . On ne saisit dans la structure de donnée que les points,les arêtes et les faces inclus dans le volume de visualisation . Les sommets des points sont exprimés dans le repère écran normalisé homogène .Cette étape est réalisée par la procédure FCNH3D

Dans cette version, on réalise la modélisation et le traitement dans une même phase.

## B. LA TRIANGULATION

### 1. CRITIQUE DE LA METHODE DES RESEAUX REGULIERS

La plupart des systèmes de cartographie numérique (exemple :mallet 74 ) partent d'un domaine d'étude D rectangulaire et d'une fonction continue  $Z = F(x,y)$  à cartographier. Cette fonction est donnée sous forme d'un réseau régulier, parallèle aux côtés du domaine D . On connaît la valeur  $Z_i = F(x_i, y_i)$  à chaque noeud  $(x_i, y_i)$  du réseau.

L'algorithme de cartographie dans ce cas de figure doit résoudre des problèmes d'interpolation de la fonction F, entre les noeuds du réseau, et le long d'une courbe de niveau donnée.

Ces interpolations seront ajoutées à l'interpolation initiale qui a permis de définir la fonction F dans les noeuds du réseau régulier .

En effet , dans la pratique ,on dispose des valeurs de la fonction à cartographier en un amas de points dispersés et non en un réseau régulier de données . C'est pour cette raison que nous avons plutôt choisi de bâtir notre algorithme de cartographie sur une triangulation découpant le domaine en triangles disjoints, dont les sommets sont les points auxquels la valeur de la fonction est définie . Cela nous évite une première interpolation et donc, des erreurs d'estimation et d'arrondi

L'utilisation de la triangulation présente d'autres avantages, notamment celui de permettre des algorithmes de cartographie qui simulent la démarche humaine . En effet la cartographie manuelle est basée sur le principe très simple d'interpolation linéaire entre les points donnés (non estimés) deux à deux .

### 2. TRIANGULATION ET MAILLAGE EN ELEMENTS FINIS

Une classe de plus en plus vaste d'équations aux dérivées partielles (provenant des problèmes physiques que rencontrent les ingénieurs) est maintenant traitée par les méthodes des éléments finis. Dans notre domaine, les modélisations-par éléments finis - de terrain ,des ouvrages de génie civil et des puits de pétrole sont de plus en plus courantes.

L'utilisation des triangulations permet l'amélioration du logiciel et son ouverture sur la modélisation, par les méthodes d'éléments finis . En effet, ces méthodes exigent la connaissance d'une triangulation du domaine D où a lieu le phénomène à étudier . Rappelons que la triangulation d'un domaine D, supposé borné et polyédrique , est par définition un ensemble T de simplexes ( triangle en dimension 2, tétraèdres en dimension 3 ) vérifiant les conditions suivantes :

- L'union des éléments de T est égale à D
- L'intersection de deux éléments de T est vide ou réduite à un point , une arête ou une face .

Pour être exploitable, une triangulation doit satisfaire de plus, aux deux conditions essentielles suivantes :

- Les éléments de T doivent être suffisamment réguliers . Quand il s'agit de triangles par exemple ,cela signifie qu'ils doivent être le plus proche possible de l'équilatéralité ,ou du moins, ne pas avoir d'angles intérieurs importants
- La densité des éléments doit être éventuellement plus grande dans certaines régions du domaine où le phénomène physique est accentué.

On peut distinguer deux grand problèmes soulevés par le maillage d'un domaine :

- La création d'un maillage
  - l'amélioration d'un maillage
- Dans cette étude on s'est d'avantage intéressé à la création du maillage qu'à son amélioration

Cependant, quand on s'intéresse au problème d'amélioration d'un maillage on peut se poser deux types de questions :

-comment améliorer le maillage d'une façon non topologique, où la seule possibilité est de modifier les coordonnées des noeuds internes .

-comment améliorer le maillage d'une façon topologique qui change les arêtes internes ,quand on s'interdit de modifier les coordonnées des noeuds externes ou internes

### 3. LES DIFFERENTES METHODES DE TRIANGULATIONS

On peut définir un maillage de type élément finis de la manière suivante:

-La zone étudiée est découpée uniquement en tétraèdres et chaque tétraèdre est défini par quatre sommets

-L'intersection de deux tétraèdres est limitée

- à un seul sommet ,
- ou à une seule arête (une arête est définie par deux sommets),
- ou à la face commune possible aux deux tétraèdres (une face est définie par trois sommets) ,
- ou bien enfin cette intersection est vide.

Chaque face de tétraèdre appartient à deux tétraèdres , à l'exception des faces sur la frontière de la zone étudiée qui appartient à un seul tétraèdre .

En dimension 2 la zone est découpé en triangles. Voici trois exemples en dimensions 2 d'une décomposition de l'espace (voir figure 8.1)qui n'est pas un maillage de type élément finis.

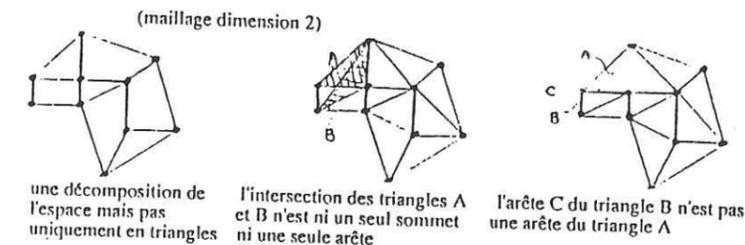


Figure 8.1(Hermeline 82)

Cependant ,diverses méthodes existent dans le domaine de génération de maillages éléments finis ,c'est à dire , la tétraèdrisation initiale de l'espace à mailler . Voici une description sommaire de certaines de ces méthodes

#### a) LA METHODE TRIVIALE DES SEGMENTS

Il s'agit de construire dans une première étape tout les segments reliant les points deux par deux . Ensuite on doit enlever les segments qui en coupent d'autres et, finalement organiser les segments restant en triangles ,c'est à dire , définir les segments participant à un même triangle .

Cet algorithme est très simple à mettre en oeuvre, mais il est très long à exécuter . Cela est dû au fait qu'il est basé sur un parcours séquentiel quadratique (N parcours de N points) de la séquence des points ,puis de la séquence des segments .

#### b) LA METHODE DE VORONOI

Il s'agit plutôt de la triangulation de Delaunay(Franco 85) qui, en fait, est le dual des polygones de Voronoi(voir figure 8.2) : Voici la définition du polygone de Voronoi:

Soient n points :  $x_1, \dots, x_n$  de l'espace  $R^2$  . Le polygone de Voronoi associé au point  $x_i$  est l'ensemble des points  $x \in R^2$  tel que  $d(x, x_i) \leq d(x, x_j)$  pour tout  $x_j, j=1, \dots, n; j \neq i$  ( $d(x, y)$  désigne la distance euclidienne de  $x$  à  $y$ ) . Les figures suivantes présentent un exemple de polygone de

voronoi associé à l'ensemble  $E=\{x_1, \dots, x_{13}\}$  et à la triangulation Delaunay correspondante.

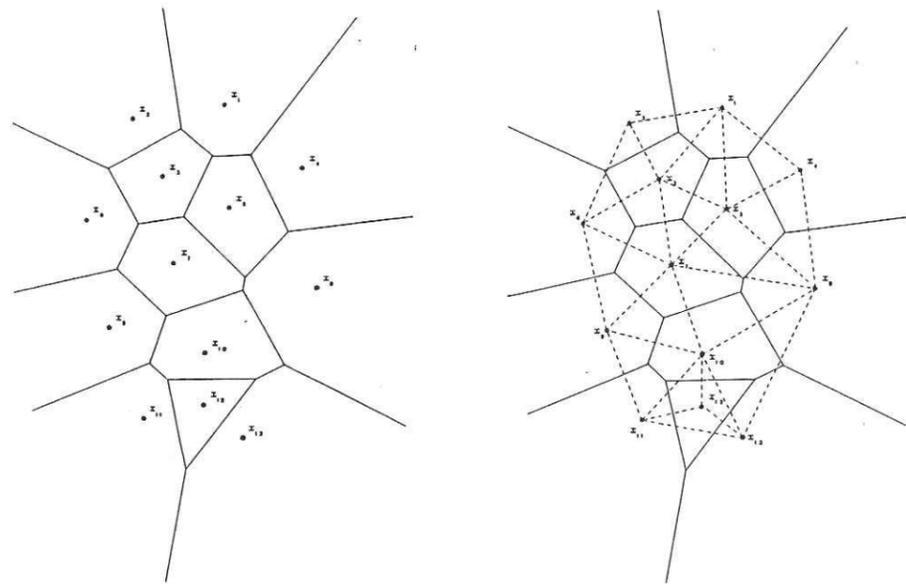


Figure 8.2(Hermeline 82)

La construction des polygones de voronoi en deux dimensions produit une partition du plan en polygones. Chaque polygone est l'intersection de  $N-1$  demi-plans ( $N$ :nombre de points). Les arêtes d'un polygone renfermant un point  $P$  sont des segments de droite, perpendiculaires au segment défini par  $P$  et chacun de ses points voisins (les plus proches de lui).

La triangulation de Delaunay est basée sur le principe de la liaison, par des segments perpendiculaires aux arêtes du polygones de Voronoi, du point  $P$  avec les points voisins.

#### c) AUTRES METHODES

D'autres méthodes de triangulation sont possibles, En effet on trouve des méthodes qui se basent sur la construction une par une des tétraèdres, d'autres méthodes sont basées sur le découpage de l'espace en polyèdres convexes.

Par ailleurs, certaines méthodes (Hermeline 82) donnent un algorithme en dimension  $n$ , basé sur les polyèdres de voronoi et la triangulation de Delaunay.

J.Y.Talon (Talon 87) a présenté des algorithmes de génération et d'amélioration de maillage en deux dimensions. On étudiera par la suite, l'algorithme de génération proposé et on suggèrera les simplifications et les améliorations adéquates à notre application.

#### 4. c) PRESENTATION ET AMELIORATION DE L'ALGORITHME CHOISI

##### a) L'ALGORITHME DE TALON

J.Y.Talon a présenté un algorithme en deux étapes; la première est de former les triangles extérieurs, la deuxième est la triangulation interne. Rappelons les données suivantes :

Soit un domaine polygonale  $D$  ( $D$  peut contenir des trous polygonaux), défini par un ensemble d'arêtes externes  $E_{ext}$  composant le contour de  $D$ . On se donne également  $V_{int}$ , un ensemble de point intérieurs à  $D$ , qui représente la discrétisation de  $D$ , et  $X=(X_{ext}, X_{int})$  la matrice des coordonnées des différents noeuds

(1) Algorithme de construction de  $D$  sans points intérieurs:

(a) présentation

Supposons  $V_{ext}$  ordonnée en  $(v_1, v_2, \dots, v_n)$

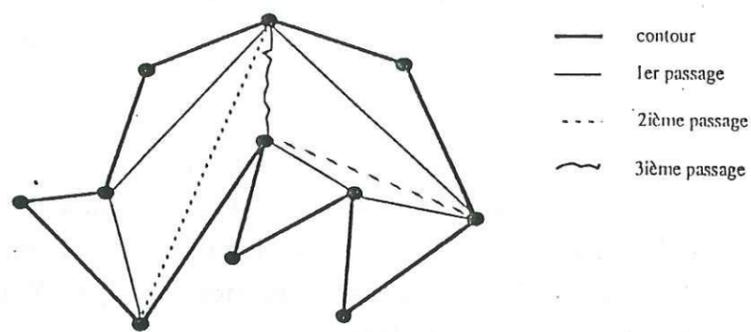
$n_{contour}=n$ ;  $contour = \{v_1, \dots, v_n\}$ ; notons  $v_{n+1} = v_1$  et  $v_{n+2} = v_2$ .

Voici l'algorithme de triangulation d'un polygone par ses points extérieurs (voir figure 8.3)

```

tant que ncontour > 3
pour i = 1, ncontour
  tester si l'arête  $v_i v_{i+2}$  est contenue dans le domaine
  contour pour former le triangle  $(v_i v_{i+1} v_{i+2})$ 
  si oui
    ncontour = ncontour - 1
    contour = contour -  $\{v_{i+1}\}$ 
    renuméroter les éléments de contours de 1 à
    ncontour
  fin
fin
fin

```



Triangulation d'un polygone par ses points extérieurs en 3 passages

Figure 8.3(Talon 87)

## (b) performances

On peut affirmer que l'algorithme est fini puisqu'à chaque passage dans la boucle "tant que", on décrémente au moins d'une unité la taille du contour.

L'algorithme exige, au pire des cas,  $(ne)$  passages dans la boucle "tant que". Le test concernant la validité de l'arête  $v_i v_{i+2}$  se réalise en  $(ne)^2$  opérations; l'algorithme forme donc une triangulation d'un domaine  $D$  sans point intérieur en  $(ne)^3$  opérations.

Cela paraît théoriquement peu performant, mais ce n'est pas le cas d'un point de vue pratique, puisqu'en général, en un passage dans la

boucle "tant que", le contour diminue très largement. Le comportement effectif de l'algorithme est plutôt réduit à l'ordre de  $O(ne)$

## (c) simplification

On propose de simplifier cet algorithme en éliminant le test, vérifiant si l'arête  $v_i v_{i+2}$  est contenue dans le domaine contour. Cela augmente les performances de l'algorithme.

(2) Algorithme de triangulation d'un domaine  $D$  muni de la discrétisation  $V_{int}$ :

## (a) principe

D'après les résultats précédents, le polygone  $(V_{ext}, E_{ext})$  peut être triangulé (sans perte de généralité on suppose que ce polygone ne comporte pas de traits intérieurs).

Soient  $U_1 \dots U_k$ , les triangles obtenus par l'algorithme exposé ci-dessus, dont tous les sommets sont des points de  $V_{ext}$ . Soit  $i \in V_{int}$ , le point intérieur à  $D$ . deux cas peuvent se présenter:

1) il existe un triangle  $U_a$ , tel que  $i$  soit contenu dans l'intérieur du triangle  $U_a$  (voir figure 8.4). Soient  $(x, y, z)$  les trois sommets du triangle  $U_a$ . Une triangulation est obtenue en remplaçant  $U_a$  par trois nouveaux triangles  $(x, y, i)$ ,  $(y, z, i)$  et  $(z, x, i)$ .

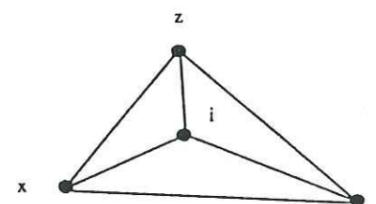


Figure 8.4

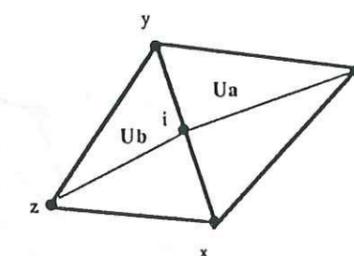


Figure 8.5

2) il existe deux triangles  $U_a$  et  $U_b$ , tels que  $i$  appartient aux deux triangles  $U_a$  et  $U_b$  (voir figure 8.5). Soient  $(x, y, z)$  les trois points du triangle  $U_a$  et  $(x, y, t)$  les trois points de  $U_b$ .

Une triangulation de  $D$  est obtenue en remplaçant  $U_a$  et  $U_b$  par quatre nouveaux triangles  $(x,i,z)$ ,  $(z,i,y)$ ,  $(i,t,y)$  et  $(x,t,i)$ .

(b) présentation

Cet algorithme nous fournit la construction d'une triangulation finale (voir figure 8.6) d'un domaine  $D$  -de contour  $(V_{ext}, E_{ext})$  et menu de la discrétisation  $V_{int}$  - lorsque le polygone  $(V_{ext}, E_{ext})$  a déjà été triangulé en  $(U_1, \dots, U_k)$  par l'algorithme précédant

L'ALGORITHME :

pour  $i = 1, n_i$

trouver le triangle  $U_a$ , ou les triangles  $U_a$  et  $U_b$  qui contiennent le point  $i$

fin

pour  $i = 1, n_i$

si  $i$  appartient à un seul triangle  $(x,y,z)$

former les trois triangles  $(x,y,i)$ ,  $(y,z,i)$  et  $(z,x,i)$

répartir les noeuds  $j > i$  figurant dans le triangle  $(x,y,z)$  dans les trois nouveaux.

sinon

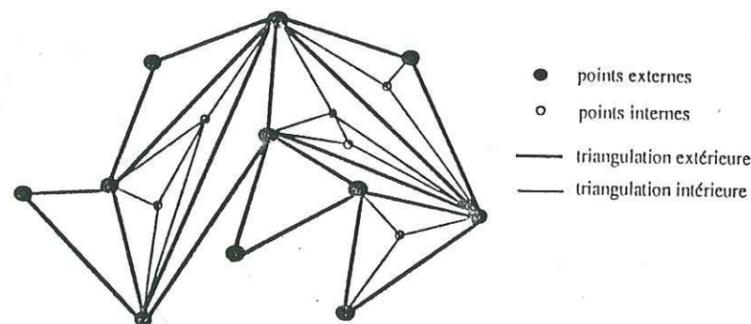
$i$  appartient à deux triangles  $(x,y,z)$  et  $(x,y,t)$

former les quatre nouveaux triangles  $(x,i,y)$ ,  $(z,i,y)$ ,  $(i,t,y)$  et  $(x,t,i)$

répartir les noeuds  $j > i$  figurant dans les triangles  $(x,y,z)$  et  $(x,y,t)$  dans les quatre nouveaux triangles

fin

fin



Triangulation finale du domaine par décomposition des triangles

Figure 8.6

(c) performance

La première boucle nécessite un nombre constant d'opérations de l'ordre de  $O(n_e * n_i)$ . La deuxième boucle admet une complexité théorique assez mauvaise : en effet, dans le cas le plus défavorable où tous les noeuds internes sont toujours situés dans le dernier triangle formé, le nombre d'opérations est de l'ordre de  $O(n_i)^2$ . On proposera par la suite une amélioration de l'algorithme pour ne pas avoir à se soucier de la bonne répartition des points.

(d) amélioration

On propose ici une nouvelle version de l'algorithme basé dans la deuxième boucle non pas sur un parcours séquentiel des points, mais des triangles

Dans la première boucle, on répartit les points sur l'ensemble des triangles. Cela nous permet d'avoir une liste de triangles dont chacun est caractérisé par l'ensemble des points qu'il contient. C'est ce qui permet de traiter dans la deuxième boucle chaque triangle à part. Dans cet algorithme on a choisi d'effectuer un parcours séquentiel des triangles. Le traitement de chaque triangle consiste à une triangulation du domaine formé par le triangle lui-même. On aperçoit déjà l'idée de la récursivité.

i) La première boucle

Il s'agit d'un parcours séquentiel, dans lequel on partitionne les points intérieurs selon les triangles extérieurs. Pour chaque point, on effectue une recherche du premier triangle à qui il appartient. Si un point appartient à un triangle il faut signaler s'il est strictement intérieur. Dans le cas contraire, c'est à dire, dans le cas où il appartient à une arête  $xy$ , on peut conclure qu'il appartient à un deuxième triangle et on continue la recherche du triangle à qui appartient le point. A la fin de la boucle on dispose de la liste des triangles extérieurs qui n'a pas changé, mais dans laquelle on a associé pour chaque triangle la liste de points qu'il englobe. Pour chaque point, on mentionne s'il est strictement intérieur.

{état initial, donnée :liste des points intérieur, liste de triangles  
extérieurs }

tant que  $i \neq \text{nil}$  (nil : fin de la liste  
trouver un premier triangle U qui englobe i  
si i appartient à une arête (xy) de U alors  
rechercher le deuxième triangle englobant i  
insérer le point (son adresse) dans la liste des  
points correspondant à chacun des 2 triangles  
sinon  
insérer i dans la liste du triangle U  
  
i := le suivant du point i

fin

*Le traitement d'un triangle externe* : c'est un parcours de la liste  
des points appartenant au triangle

procédure traiter-triangle (x,y,z)

i:= tête de liste des points

si  $i \neq \text{nil}$  alors

si i est strictement interne au triangle (x,y,z) alors

former les trois triangles (x,y,i) (y,z,i) et (z,x,i);

répartir les points qui suivent i dans la liste dans les  
trois triangles ;(on précise si le point est strictement  
intérieur ou appartenant à deux triangles)

remplacer le triangle(x,y,z)par les trois nouveaux  
triangles dans liste ;

traiter-triangle(x,y,i);

traiter-triangle(y,z,i);

traiter-triangle(z,x,i);

si i appartient à un arête xy de (xyz) alors

former les deux triangles (z,x,i) et (z,y,i);

répartir les points qui suivent i dans la liste, dans les  
deux nouveaux triangles ; (on précise si le point est  
strictement interne à un nouveau triangle ou s'il  
appartient à une arête)

remplacer le triangles par les deux nouveaux;

traiter-triangle(z,x,i);

traiter-triangle(z,y,i);

sinon le triangle est vide (on arrête la triangulation) .

ii) La deuxième boucle

*La deuxième boucle* -donnée : liste des triangles externes

-résultat : liste des triangles internes

c'est un parcours séquentiel dans lequel on applique la procédure  
traiter-triangle(qui est récursive) à tous les triangles externes .  
L'algorithme suivant peut produire à partir d'un triangle (père), un arbre  
de triangles dont les feuilles sont les triangles qui ne contiennent aucun  
point à leur intérieur .

```

tant que triangle-ext # nil
  traiter-triangle (triangle-ext)
  avancer (triangle-ext)
fin

```

IL est possible de généraliser l'algorithme récursif de la procédure traiter-triangle, pour produire une triangulation complète du domaine, sans passer par la triangulation externe. Ceci est réalisable en définissant au départ un triangle, englobant tout le domaine. L'application de la procédure traiter-triangle au triangle englobant, produira une triangulation complète du domaine.

### C. L'ALGORITHME DE LA CARTOGRAPHIE

Mallet a déjà proposé un algorithme de traçage de courbe d'isovaleur, en partant d'une maille régulière de données. Cet algorithme n'est pas applicable dans notre cas, puisqu'on a procédé par triangulation. Cependant, on s'est inspiré du travail de Mallet pour produire un algorithme de traçage d'une courbe d'isovaleur, à partir d'une triangulation du domaine.

Chaque point sommet d'un triangle est défini par ses coordonnées et par une valeur correspondant à la topographie, l'épaisseur, la profondeur... (selon l'application recherchée).

La fonction  $F$  est définie en tout point sommet de triangle;  $Z = F(x,y)$  est la valeur correspondant à la quantité étudiée.

Une courbe de niveau  $Z_0$  est l'ensemble des points  $(x,y)$  tels que  $F(x,y) - Z_0 = 0$ . Une courbe peut avoir plusieurs composantes "fermée" ou "ouvertes". Supposons qu'on numérote les points de chaque triangle  $T(i,j,k)$ . L'algorithme de détermination de la courbe de niveau  $Z_0$  est formé par l'ensemble des étapes suivante :

\*Etape 1: déterminer l'ensemble des triangles traversés par la courbe  $Z_0$ . Il s'agit de parcourir systématiquement tous les triangles  $T(i,j,k)$  et de retenir ceux qui sont traversés par une composante de la courbe. Chaque fois que la courbe passe par un côté du triangle, on note le numéro du point haut ( $Z > Z_0$ ) et celui du point bas ( $Z < Z_0$ ). Une

composante ne peut traverser un triangle que par deux côtés. On retient pour chaque triangle traversé un quadruplé formé par les extrémités haut et bas des deux côtés traversés  $(a,b,c,d)$ .

S'il y a  $N$  triangles traversés par  $Z_0$ , l'ensemble des diverses composantes de la courbe à dessiner sera complètement défini à partir de l'ensemble des  $N$  quadruplés  $(a,b,c,d)$  retenus et stockés dans un tableau non ordonné.

\*Etape 2 : Ordonner le tableau des éléments traversés par la courbe.

\*Etape 3 : Définir les coordonnées des points d'intersection entre la courbe et les quadruplés du tableau. Ce sont ces points d'intersection qui définissent la courbe  $Z_0$ .

### D. LA REALISATION DES PROFILS GEOLOGIQUES

#### 1. LE PROBLEME DE CORRELATION

La réalisation des profils géologiques est basée surtout sur la corrélation entre sondages. L'automatisation de cette corrélation est une opération très délicate. Il est très difficile d'établir des critères permettant la corrélation automatique. La (figure 8.7) montre qu'il existe plusieurs corrélations, théoriquement possibles, pour les mêmes sondages.

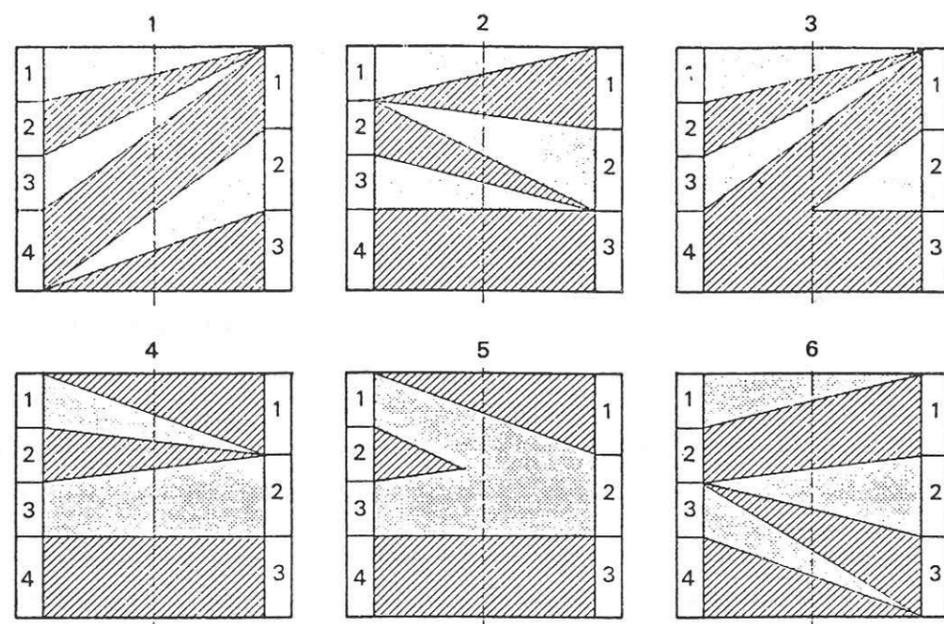


Figure 8.7

Il serait intéressant de construire un système expert contenant une base de connaissance spécifique pour une région donnée, (par exemple: l'âge des formations, les altérations possible, les failles, etc) afin de pouvoir établir par ordinateur l'évolution des couches dans le temps et dans l'espace.

Cependant, le choix effectué est beaucoup plus simple. Il est basé essentiellement sur l'interactivité. C'est le géologue utilisateur qui doit décider et dessiner sur l'écran les liaisons entre les couches, par l'intermédiaire d'un organe de commande qui est la souris.

## 2. L'ALGORITHME

Le but est de modéliser, dans une structure de donnée (qu'on peut conserver) les limites des couches dans l'espace, sous forme de facettes dont les coordonnées des sommets sont exprimées dans le même repère objet que les sondages. Cette modélisation doit être faite à partir des limites tracées par l'utilisateur sur l'écran.

On doit suivre les étapes suivantes :

-Etape 1 : Le calcul de l'équation du plan P, défini par les 2 sondage à corrélés.

-Etape 2 : La définition de la transformation de visualisation 2D, permettant de faire correspondre à tout point du plan P (contenu dans une clôture) un point de l'écran (contenu dans une fenêtre). Cette transformation est définie par la donnée d'une clôture et d'une fenêtre.

-Etape 3 : Définir la transformation inverse permettant de faire correspondre à tout point de la fenêtre écran, un point de la clôture contenue dans le plans P.

-Etape 4 : Le traçage sur l'écran des sondages, suite à une projection orthogonale perpendiculaire au plan P.

-Etape 5 : Afficher le curseur de la souris et définir les faces dont les sommets sont exprimés en coordonnées écrans à partir des points dessinés par l'utilisateur et qui délimitent les limites des couches.

-Etape 6: Transformer les coordonnées x,y des sommets de ces faces en coordonnées objets en appliquant la transformation de visualisation inverse. Puis déduire la coordonnée z, à partir des coordonnées x et y dans le repère objet et de l'équation du plan P.

-Etape 7: Stocker ces faces dans la structure de données modélisant le volume

E. LES BLOCS-DIAGRAMMES

1. LES BLOCS-DIAGRAMMES\_RELIEF

On modélise ici la topographie d'un terrain ou d'une couche. On peut considérer qu'il s'agit d'une fonction à deux variables  $F(x,y)=z$ .

Dans ce cas particulier de modélisation, on ne tient pas compte des caractéristiques de remplissage des faces (couleur et style) correspondant à la nature lithologique du terrain.

La définition des facettes planes du volume diffère selon qu'on dispose d'une réseau régulier de données ou d'un amas irrégulier

a) CAS DE MAILLE REGULIERE

Le principe est de considérer les faces planes qui sont perpendiculaires à l'axe des x ou à l'axe des y. Ces faces sont caractérisées par le fait que l'ensemble des sommets de contour sont définis par  $F(x,y)=z$ , avec La même valeur pour la coordonnée x (x =une constante) ou avec y = une constante.

Les figures (8.8,8.9)représentent le relief d'un terrain (butte Sainte-Geneviève), à partir d'une maille régulière de données d'altitudes, relatives à la butte.

La figure 8.8 représente le plan d'échantillonnage des altitudes relatives à la butte.

La figure 8.9 représente une projection perspective du volume modélisé à partir du réseau de données.

372	372	372	370	366	355	320	305	293	284	274	266	261	262	262	262	257	254	255	248
368	368	368	367	364	355	324	309	295	282	279	276	278	278	275	264	262	263	260	254
366	367	366	364	363	357	330	312	295	286	295	296	297	290	278	272	272	271	267	260
364	364	363	362	361	359	345	325	312	309	307	309	307	297	288	284	282	278	273	265
361	362	362	361	359	357	353	345	329	322	318	319	318	310	302	295	287	283	275	267
359	359	358	357	355	340	340	345	352	345	341	344	337	319	311	304	299	279	260	260
357	357	355	359	342	322	325	335	351	356	356	357	356	345	317	305	287	272	256	240
356	348	327	315	310	308	311	323	342	351	357	359	359	355	321	305	298	276	260	245
345	320	310	300	295	295	298	309	330	349	357	361	362	360	328	309	299	282	265	250
324	310	300	290	285	284	290	305	324	347	357	361	361	358	330	310	297	283	269	260
300	300	293	283	278	270	285	302	319	335	343	341	339	332	320	303	285	272	257	250
300	293	285	277	266	264	279	293	306	316	323	320	320	315	304	290	272	255	242	235
293	287	275	268	257	258	270	281	290	300	310	305	300	298	287	275	257	245	232	226
290	279	270	262	252	247	259	267	276	286	292	291	285	280	265	254	245	237	227	220
277	268	260	257	247	242	248	255	265	272	275	272	268	262	248	242	235	230	223	216
265	259	250	246	240	237	241	246	253	257	254	248	242	237	232	225	220	215	212	209
252	245	241	236	231	228	233	240	242	243	238	237	235	232	227	221	216	214	210	209
239	235	232	229	225	219	225	230	229	228	226	234	221	217	214	212	209	208	208	208
239	230	228	227	222	218	218	218	217	218	214	213	212	210	208	208	207	207	207	207
225	224	222	220	216	215	210	210	209	209	209	209	209	208	207	206	207	207	207	207

Figure 8.8 :plan d'échantillonnage



Figure 8.9

b) CAS D'AMAS IRREGULIER:

On ne dispose ici que de données en des point géographiquement dispersées . Il s'agit de subdiviser la surface topographique, en un ensemble de facettes planes dont les points de contours sont les points donnés .La triangulation est la façon la plus simple pour produire une telle répartition .

Dans la figure 8.10 , on présente une vue en perspective du même terrain (la butte de Sainte-Geneviève) déjà présenté dans la figure 8.9 .Mais cette fois ,le volume a été modélisé à partir d'un amas irrégulier de données .

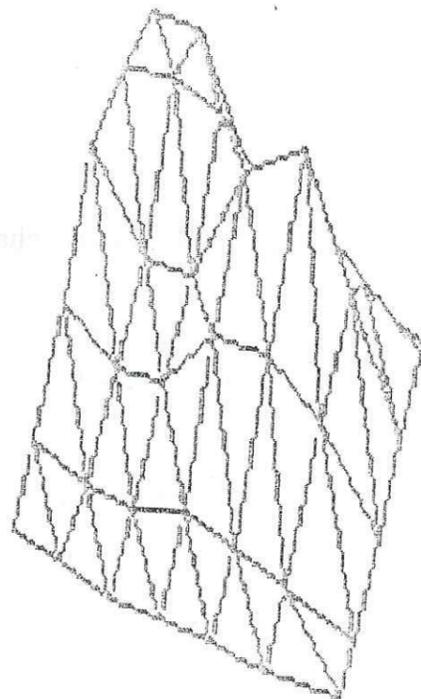


Figure 8.10

2. LES BLOCS-DIAGRAMMES TERRAIN

Un "bloc-diagramme terrain" est un volume délimité sur les côtés par les profils géologiques -définis par corrélation entre sondages périphériques - et du dessus par la surface du sol affleurant ou par le toit d'une couche déterminée .

La modélisation du bloc-diagramme est la définition de l'ensemble des facettes de son volume .Cet ensemble est défini par l'union des ensembles dont les éléments sont les facettes des profils d'une part ,et par les facettes de la surface de dessus d'autre part. Contrairement au bloc-diagramme relief ,on doit indiquer pour chacune des facettes les caractéristiques de remplissage ,correspondant à la nature lithologique de la couche .

Nous avons déjà détaillé notre choix concernant la modélisation d'un volume .

Il s'agit en fait ,de modéliser chaque profil géologique, ainsi que la surface de dessus, dans un volume représenté par un fichier . C'est à partir de ces fichiers qu'on obtient le fichier représentant le bloc-diagramme, en exécutant la procédure Concaten . Ce n'est pas une simple concaténation, car certaines autres opérations sont imposées par la structure choisie des fichiers.

Une fois le fichier obtenu ,on peut réaliser le transfert dans une structure de données-volumes en faisant appel à la procédure tsfsdv.

Les figures suivantes représentent la modélisation d'un bloc diagramme, à partir des profils géologiques et de la surface de dessus d'un terrain modèle d'un synclinal perché .

La figure 8.11 illustre les forages périphériques et leurs positions en perspective .

La figure 8.12 illustre les profils géologiques et la surface de dessus, à partir desquels on a modélisé le volume.

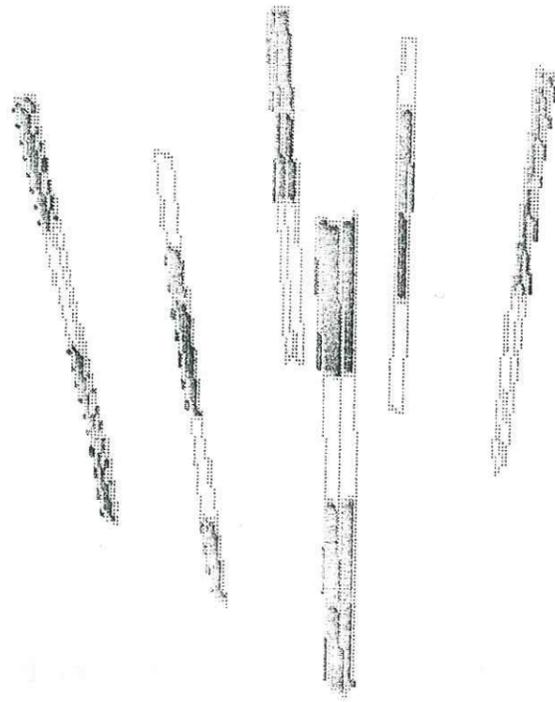


Figure 8.11

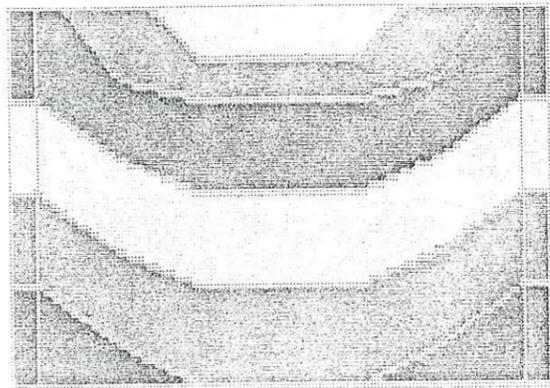


Figure 8.12 - a :profil géologique N°1 (coté de face)

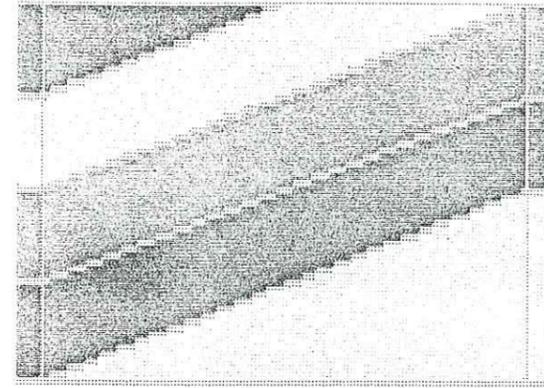


Figure 8.12-b : profil géologique N°2 (coté de droite)

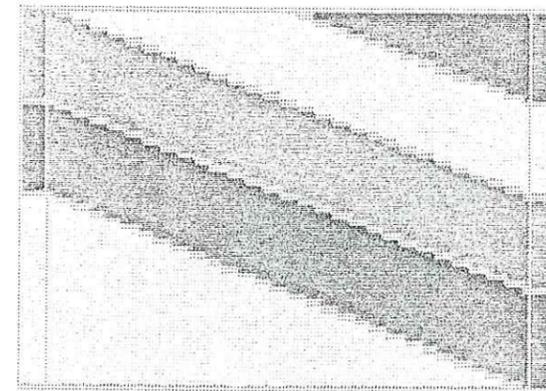


Figure 8.12-c: profil géologique N°3 (coté gauche)

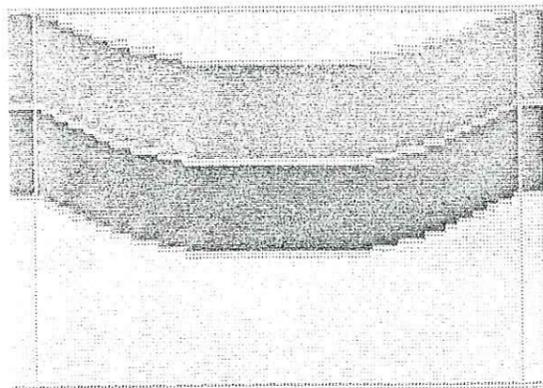


Figure 8.12-d: profil géologique N°4 (coté de derrière)

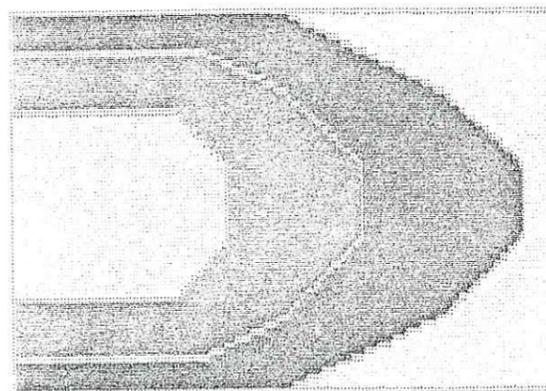


Figure 8.12-e : la surface de dessus.

La figure 8.13 représente une perspective du volume, modélisé sous quatre points de vue différents

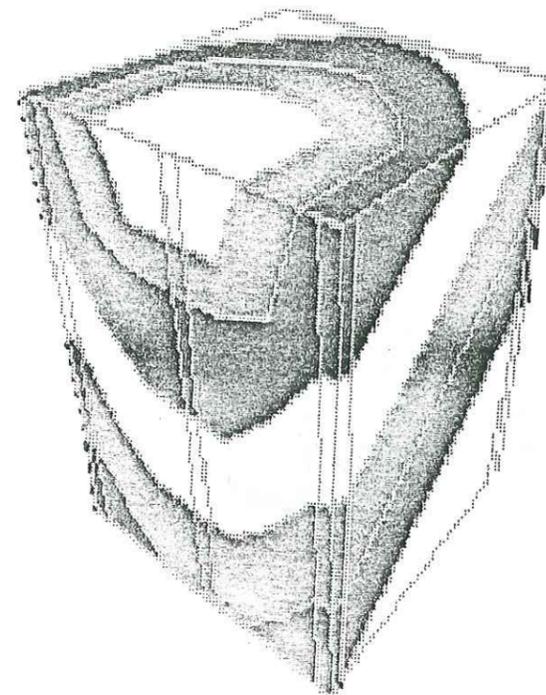


Figure 8.13-a : projection perspective haut-droite-face

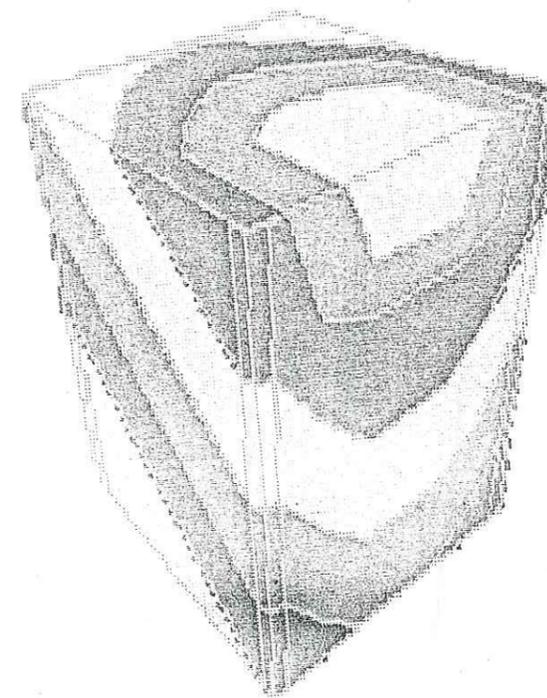


Figure 8.13-b: projection perspective haut-gauche-face

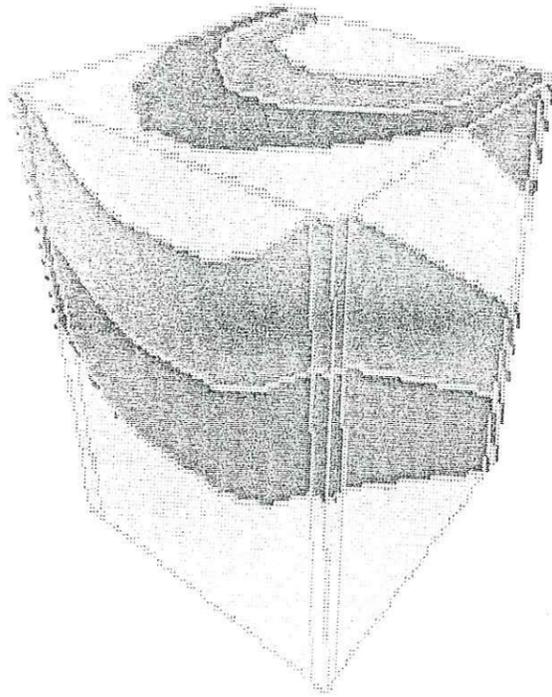


Figure 8.13-c: perspective haut-droite-derrière

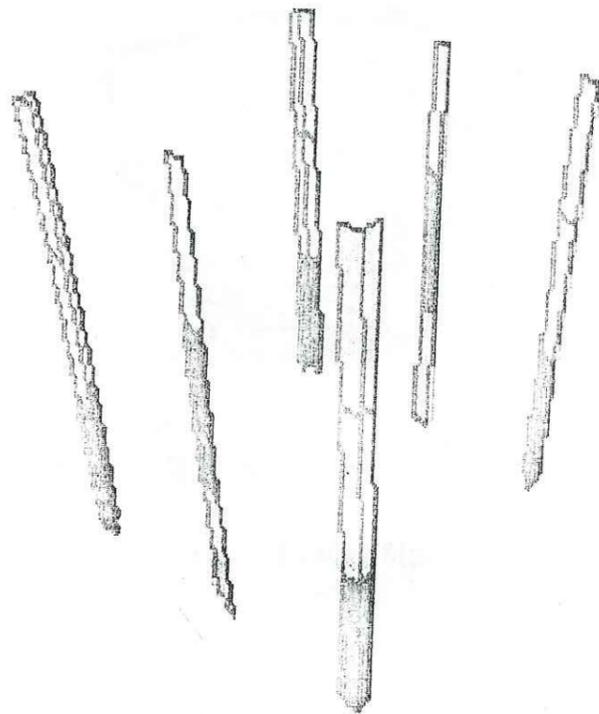


Figure 8.14-a: perspective des forages périphériques

La figure N 8.14 représente un modèle de terrain de structure antiforme .  
On représente ici les forages à partir desquels on définit les profils géologiques d'une part et une vue en perspective du volume ,d'autre part.

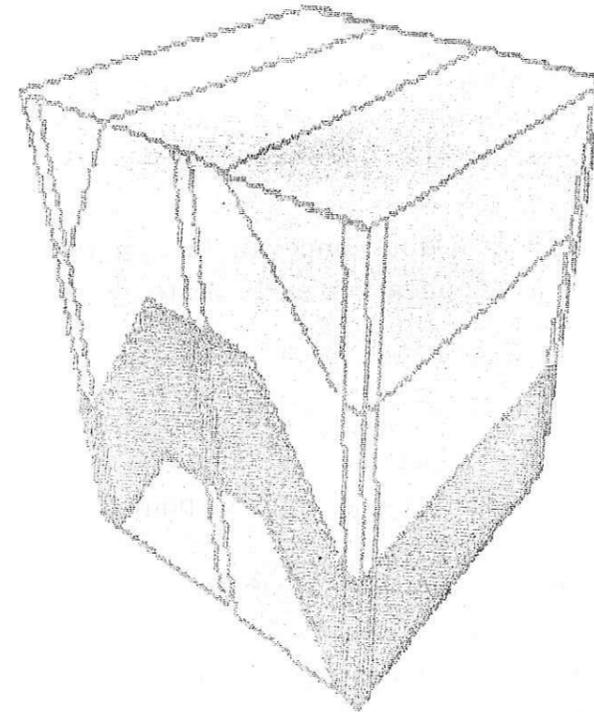


Figure 8.14-b: perspective haut-droit-face

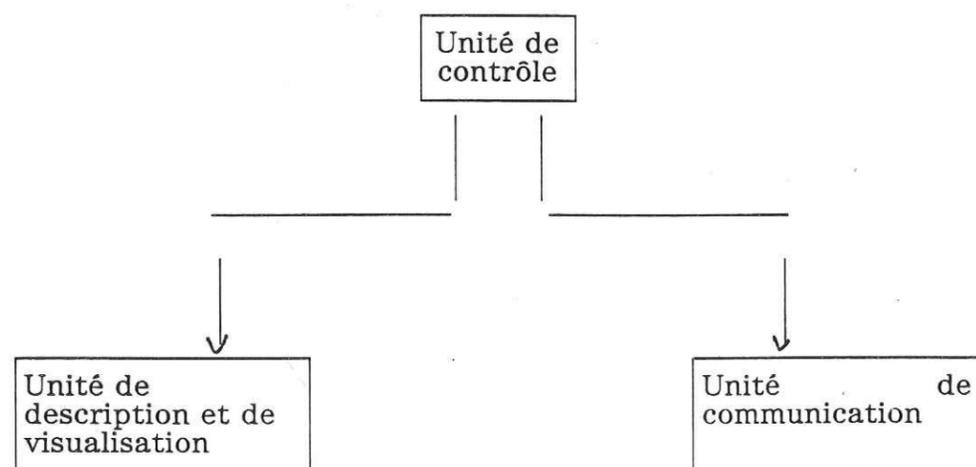
## IX. L'ARCHITECTURE LOGICIELLE

### A. ARCHITECTURE DU LOGICIEL GRAPHIQUE

La plus grande exigence que le logiciel doit satisfaire est l'adaptabilité . Cela se traduit par une indépendance vis à vis du matériel et de l'application . Le changement du matériel et le changement, ou l'ajout, de certaines fonctionnalités ne doivent pas affecter tout le système . On doit concevoir une partie liée au matériel, une autre partie liée à l'application et le reste du système doit être indépendant .

La modularité est le découpage du logiciel en plusieurs parties qui sont les modules . Un ensemble de modules proches par leurs fonctionnalités ou leurs structures de données peut former une unité .

Le système graphique sera formé par une unité de communication, une unité de description et de visualisation et une unité de contrôle . L'unité de contrôle et celle de description et de visualisation peuvent être considérées comme des banques d'opérateurs . L'unité de contrôle s'occupe de l'enchaînement des opérateurs selon les besoins de l'application . Ces unités sont organisées hiérarchiquement comme cela est illustré ci-dessous.



### 1. L'UNITE DE COMMUNICATION

C'est l'unité qui gère l'identité et la structure d'une image , elle assure les primitives suivantes :

-créer image : Cette primitive alloue la place mémoire pour une image nouvellement créée .

-détruire : Détruit une image et libère sa place mémoire

-nommer : Donne un nom à une image .

-structure: Gère la structure des images . Si on choisit une structure d'arbre ,on peut concevoir qu'une image est un ensemble d'images ; on définit alors une image-père , une image-fils et une image-frère

-parcours : Parcourt l'ensemble des images pour effectuer une action .

-info : Nous informe sur l'état d'une image ou l'un des attributs .

### 2. L'UNITE DE DESCRIPTION ET DE VISUALISATION

Cette unité est formée par l'ensemble des modules de la bibliothèque que nous avons réalisé (voir chapitre 7). Elle assure la liaison avec le matériel et gère les attributs suivants : La morphologie , l'aspect, la géométrie et l'éclairage de l'image.

\*La morphologie : Il s'agit surtout de la modélisation et de la construction de l'objet à visualiser .

Par exemple on a modélisé un bloc diagramme par un volume, défini par les facettes qui le délimitent .

\*L'aspect : Définit la couleur d'une image .

\*La géométrie : Elle est définie par une matrice qui traduit la position de l'image sur l'écran , les déplacements qu'elle a subis

(l'historique) ; pour les perspectives ,elle traduit la position de l'obsevateur . C'est la transformation de visualisation .

### 3. L'UNITE DE CONTROLE

Cette unité s'occupe des ordonnancements des processus (primitive), selon les besoins de l'application; elle assure la liaison avec celle-ci et effectue le transfert entre l'unité de communication et l'unité de description.

### **B. L'ARCHITECTURE DU LOGICIEL GENERAL**

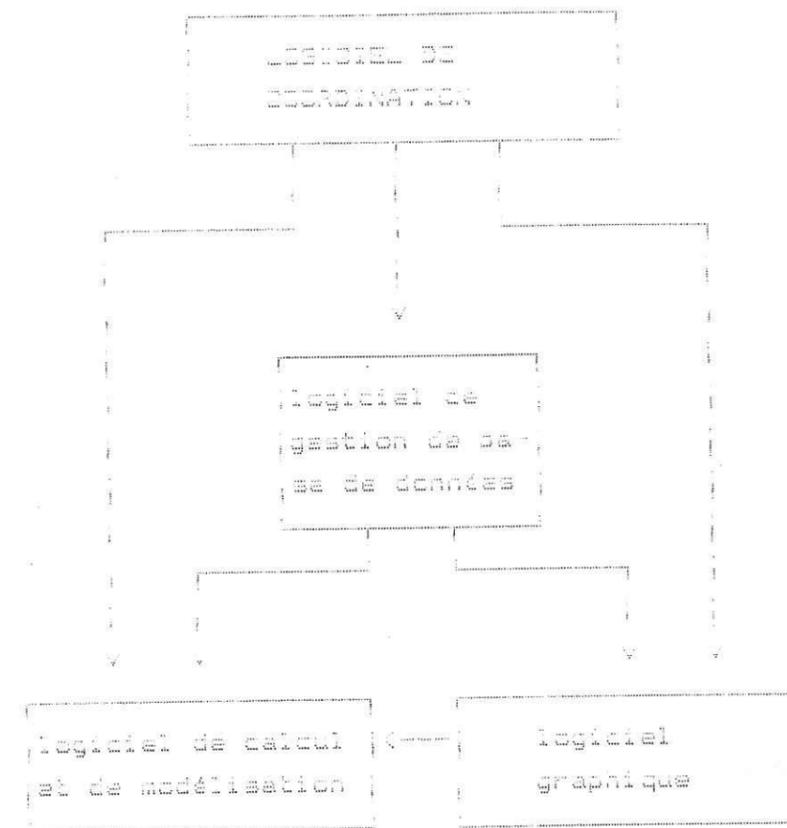
Comme le signale le cahier des charges, le logiciel graphique à réaliser fait partie d'un logiciel général de stockage et de traitement des données géologiques et géotechniques .Ce logiciel général comporte trois sous- logiciels communiquant entre eux par l'intermédiaire d'un logiciel de coordination . Les trois sous-logiciels sont:

-Un sous-logiciel de stockage et d'interrogation sélective des données géologiques et géotechniques .

-Un sous-logiciel de calcul numérique et géostatistique effectuant les estimations et les interpolations . Ce logiciel effectue la modélisation numérique et géométrique des données géologiques . Les modèles produits vont être transmis au sous logiciel graphique, pour la cartographie et la visualisation des données .

-Un sous-logiciel graphique qui utilise les résultats fournis par le logiciel de calcul et de modélisation, pour produire des cartes et des illustrations tridimensionnelles : c'est le sous-logiciel à réaliser.

La communication entre les différents logiciels peut être schématisée comme suit :



Cette architecture est modulaire et permet une certaine adaptabilité au logiciel . Mais elle pourrait être radicalement modifiée en cas d'utilisation en cas d'un langage orienté objet (exemple smaltalk) . Ceci est vivement conseillé . L'utilisation d'un langage orienté objet permet de ne plus avoir d'un côté une banque de données et de l'autre côté des logiciels de traitement de ces données . Le logiciel se présentera dans ce cas sous la forme d'une base de connaissances . Celle-ci est une collection d'objets . Chaque objet représente à la fois un savoir déclaratif (base de donnée locale ) et un savoir procédural(les procédures) . Ces objets ont des liens de spécification ,d'héritage et de généralisation . Cette conception améliorera l'architecture logiciel et lui permettra une meilleur capacité de modélisation, d'adaptabilité et d'évolution.

## X. CONCLUSION GENERALE

Nous avons accordé dans cette étude autant d'importance aux méthodes d'analyse et de conception qu'à la réalisation du système .

L'utilisation de la méthode System Analysis Data Technic nous a permis ,après la définition des besoins dans un cahier des charges , de constituer un dossier d'analyse des fonctionnalités et de spécifier un système général de stockage ,de traitement et de visualisation graphique et cartographique des données géologiques et géotechniques . Cette méthode nous a permis de simuler les fonctionnalités du système, de découvrir leurs interactions et d'établir la liste des données manipulées par le système .

L'application du modèle entité-association nous a conduit à la définition des entités correspondant aux données géologiques et géotechniques , et nous a permis de définir les relations entre ces entités (ou les ensembles d'entités) .Cela nous a préparé le terrain pour la phase de la réalisation d'une base de données relationnelle

La conception du système a été basée sur la méthode orientée objet . L'utilisation de cette méthode nous a conduit à un système modulaire qui facilite l'ajout et la modification des fonctionnalités, ainsi que la modification de l'architecture du système.

La définition des objets, par l'intermédiaire de la méthode orientée objet ,ainsi que la définition des entités dans le modèle entité-association nous ont permis de contribuer à la spécification formelle des données géologiques . Cette formalisation reste à compléter et elle sera utile pour la formation d'une base de connaissance dans un système plus intelligent

Cependant , nous remarquons que la simple utilisation de la méthode orientée objet n'est pas suffisante pour produire un système qui soit une image fidèle du monde réel , nous devons aussi disposer des moyens permettant d'exprimer et de réaliser notre conception . Il est donc nécessaire d'utiliser un langage adéquat ,capable d'exprimer le concept d'objet . Bien que le langage Pascal soit assez typé et permette la compilation séparée, son utilisation dans un contexte orienté objet ne

semble pas avantageuse ; il est nettement préférable d'utiliser un langage orienté objet (exemple : smalltalk) qui permet directement la création des objets, des groupes d'objets et des classes ainsi que leur manipulation par de simples instructions . Les langages orientés objets peuvent donc être utilisés à partir de la phase de conception , ce qui accroît leur intérêt.

L'utilisation d'un langage orienté objet pourrait nous pousser à rectifier certains éléments de la spécification . Il ne s'agira plus de construire d'un côté une base de données et de développer de l'autre côté un logiciel de traitement de données .Le système ne sera qu'une collection d'objets:c'est la base de connaissance . Chaque objet représente à la fois un savoir déclaratif(base de données locale) et un savoir procédural (le traitement) . La connaissance (données et programmes ) est distribuée entre chacun des objets.

La spécification effectuées concerne le logiciel général alors que la réalisation ne concerne que le sous-logiciel graphique. La réalisation du reste du logiciel doit être basée sur la spécification et la conception orientée objet effectuées

La bibliothèque graphique réalisée comporte un ensemble d'unités de modélisation et de visualisation en deux et en trois dimensions, avec élimination des surfaces cachées . La réalisation de cette bibliothèque a assuré une indépendance vis à vis de l'application ,en séparant celle-ci des problèmes graphiques . Ceci nous a permis de travailler à un niveau d'abstraction plus élevé . La hiérarchie que nous avons visée dans les modules de la bibliothèque a permis plus d'adaptabilité.

La modélisation des objets, ainsi que la structure de données choisies dans les modules réalisés,permettent des améliorations dans le sens de l'interactivité . Elles visent à permettre de corriger le dessin et de participer à son élaboration .

## XI. ANNEXE A : PRESENTATION DES TECHNIQUES GRAPHIQUES

### A. LES TRANSFORMATIONS GRAPHIQUES

#### 1. INTRODUCTION

La modélisation des objets, la simulation de leurs déplacements et de leurs manipulations dans le plan ou dans l'espace, ainsi que leurs affichages impliquent des transformations linéaires telles que les translations, les rotations, les changements d'échelles, les symétries et les glissements. Les projections sont les transformations les plus complexes, elles permettent de passer de l'espace au plan et ont la particularité de ne pas posséder de transformations inverses; elles seront traitées à part.

#### 2. LES TRANSFORMATIONS LINEAIRES

Les transformations linéaires permettent de ne pas calculer la transformée de chaque point de l'objet, mais seulement de quelques points caractéristiques. Par exemple, considérons le segment de droite AB. La transformée de ce segment est un segment de droite dont les extrémités A' et B' sont les transformées des points A et B.

##### a) LA TRANSFORMATION LINEAIRE D'UN POINT

L'objet graphique le plus élémentaire est le point. Il est caractérisé par deux coordonnées cartésiennes dans le plan et par trois coordonnées dans l'espace. Déterminer le point transformé d'un point revient à calculer ses coordonnées, à l'aide d'une formule de transformation.

La transformation est donnée par une équation linéaire par coordonnée:

$$\begin{aligned} X' &= a_{11} X + a_{12} Y + a_{13} Z + d_1 \\ Y' &= a_{21} X + a_{22} Y + a_{23} Z + d_2 \\ Z' &= a_{31} X + a_{32} Y + a_{33} Z + d_3 \end{aligned} \quad (1)$$

##### b) LA NOTATION MATRICIELLE

Dans la notation matricielle, on met en évidence les coefficients représentant la transformation. Le point et sa transformée sont représentés par des vecteurs colonnes.

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix} \quad (2)$$

Sous forme abrégé : (p', p, d sont des vecteurs)

$$P' = A P + d$$

##### c) LES PROPRIETES GEOMETRIQUES

-Le point d'intersection de deux droites donne par transformation le point d'intersection de leurs transformées

-Deux droites parallèles AB et CD restent parallèles après la transformation.

#### 3. LA MATRICE HOMOGENE

La transformation exprimée par l'équation (2) peut être représentée par une matrice  $M_{3 \times 4}$  de trois lignes sur quatre colonnes. En multipliant un vecteur U à quatre éléments par cette matrice, on obtient un vecteur V à trois éléments

$$\begin{vmatrix} X' \\ Y' \\ Z' \end{vmatrix} = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \begin{vmatrix} d_1 \\ d_2 \\ d_3 \end{vmatrix} \begin{vmatrix} X \\ Y \\ Z \\ 1 \end{vmatrix} \quad (3)$$

Rappelons que les transformations possèdent des transformations inverses, or seule une matrice carrée peut être inversée. Ce qui nous oblige à présenter toute transformation par une matrice carrée.

On peut remarquer par ailleurs que l'équation (4) est équivalente aux équations (3) et (2)

$$\begin{aligned} X' &= a_{11} X + a_{12} Y + a_{13} Z + d_1 \\ Y' &= a_{21} X + a_{22} Y + a_{23} Z + d_2 \\ Z' &= a_{31} X + a_{32} Y + a_{33} Z + d_3 \\ 1 &= 0 X + 0 Y + 0 Z + 1 \end{aligned} \quad (4)$$

Ce qui est équivalent à l'équation (5)

$$\begin{vmatrix} X' \\ Y' \\ Z' \\ 1 \end{vmatrix} = \begin{vmatrix} a_{11} & a_{12} & a_{13} & d_1 \\ a_{21} & a_{22} & a_{23} & d_2 \\ a_{31} & a_{32} & a_{33} & d_3 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} X \\ Y \\ Z \\ 1 \end{vmatrix} \quad (5)$$

L'homogénéisation est donc réalisée en représentant toute transformation dans l'espace par une matrice carrée d'ordre 4, la dernière ligne valant 0 0 0 1 : c'est la matrice homogène.

La matrice homogène d'une transformation dans le plan est une matrice carrée d'ordre 3 dont la dernière ligne est 0 0 1

Avec cette notation, la composition de deux transformations U et V, représentées par des matrices de ce type, correspond à une matrice homogène W égale au produit dans l'ordre inverse des deux matrices U et V. La composition de U et V est représentée par la matrice  $W = V \cdot U$

#### 4. LES COORDONNÉES\_HOMOGENES

La représentation universelle d'une transformation linéaire par une matrice carrée correspond à l'usage des coordonnées homogènes. Un point d'un espace à N dimensions est décrit par un vecteur de N+1 coordonnées. La coordonnée supplémentaire est nommée w.

- Dans le cas du plan :

Les coordonnées cartésiennes d'un point sont : x, y

Les coordonnées homogènes d'un point sont X, Y, W avec  $X=W \cdot x$ ,  $Y = W \cdot y$

- Dans le cas de l'espace :

Les coordonnées cartésiennes d'un point sont x,y,z.

Les coordonnées homogènes sont X,Y,Z,W avec :

$X = W \cdot x$   $Y = W \cdot y$ ,  $Z = W \cdot z$ .

La valeur de W doit être différente de zéro, on choisit simplement  $W = 1$

Le passage inverse des coordonnées homogènes aux coordonnées cartésiennes, se réalise en divisant chacune des coordonnées par W :  $x = X/W$ ,  $y = Y/W$  et  $z = Z/W$

Les coordonnées homogènes permettent de représenter un point à l'infini (de vecteur directeur P), par le vecteur homogènes [ Px Py Pz 0 ] (la composante W étant nulle), et facilitent le traitement d'un certain nombre de problèmes dont la composition des transformations.

#### 5. LES TRANSFORMATIONS GEOMETRIQUES ET LES TRANSFORMATIONS DE COORDONNÉES

On peut décrire les transformations par deux points de vue complémentaires. Prenons par exemple, le cas du déplacement d'un objet.

Le premier point de vue est de considérer que l'objet se déplace par rapport à un système de coordonnées fixes. Ce déplacement s'effectue par transformations géométriques

Le deuxième point de vue part considère un objet restant immobile dans un système de coordonnées en mouvement par rapport à cet objet. Cette opération s'effectue par transformations de coordonnées. C'est un changement de système de coordonnées.

Pour illustrer la différence entre les deux points de vues, on va étudier le cas de la translation.

- Point de vue de transformation géométrique (Figure A1.1) : Considérons un objet "obj" dans le plan, si l'objet est déplacé vers une nouvelle position, il peut être regardé comme un nouvel objet obj'. Un point P(X,Y) de l'objet obj peut être considéré comme un vecteur, soit P'(X',Y') le point de obj' image de P par la translation T. On a P' = P + T

On a alors  $X' = X + Tx$  et  $Y' = Y + Ty$

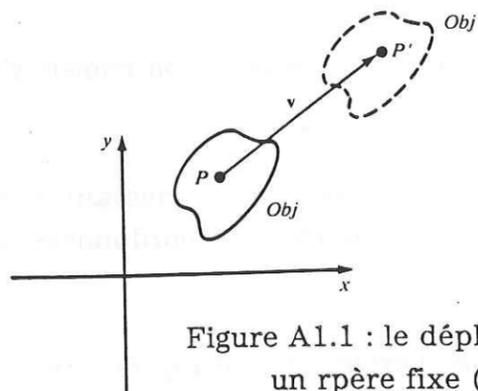


Figure A1.1 : le déplacement d'un objet dans un repère fixe (Plastock 87)

- Point de vue de transformation de coordonnées (changement de repère):

Dans la figure A1.2, le système de coordonnées (x y) est déplacé vers une nouvelle position. La translation T affecte le repère. Le point p peut être considéré comme un vecteur P dans le repère (x y) et comme le vecteur P' dans le repère (x'y')

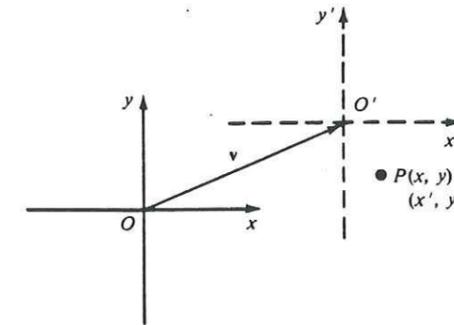


Figure A1.2 : changement de repère. (Plastock 87)

$$\begin{aligned} op = oo' + o'p & \quad \langle \rangle P = T + P' \quad \langle \rangle P' = P - T \quad \langle \rangle \\ x = Tx + x' & \quad \langle \rangle x' = x - Tx \\ y = Ty + y' & \quad \langle \rangle y' = y - Ty \end{aligned}$$

$$\text{Or } -T = T^{-1}$$

En coordonnées homogènes, la transformation T qui transforme le point o en o', ainsi que les axes x en x' et y en y', est représentée par une matrice de transformation M. Le passage du repère (x y) à celui (x' y') revient à transformer les coordonnées d'un point par la transformation inverse  $T^{-1}$ .

Le passage d'un repère à un autre se fait en transformant les coordonnées des points dans le nouveau repère, à partir de ces coordonnées dans le repère d'origine, par une matrice généralisée  $M^{-1}$  :  $P' = M^{-1} P$ . Cette matrice est l'inverse de la matrice M qui transforme le repère 1 en repère 2 c'est à dire l'extrémité d'un vecteur directeur du repère 1 en l'extrémité du vecteur correspondant du repère 2 (exprimée dans le repère 1)  $i' = M i$   $j' = M j$   $k' = M k$

#### 6. LES TRANSFORMATION GEOMETRIQUES EN COORDONNEES HOMOGENES

Une transformation dans l'espace (3 dimensions) nécessite une matrice carré d'ordre 4. Dans un plan, toute transformation se représente par une matrice d'ordre 3. On représente la transformation inverse par la matrice inverse. La matrice d'une projection présente une ligne nulle, ce qui annule le déterminant et empêche l'inversibilité.

La dernière ligne de la matrice d'une transformation comporte des zéro et se termine par 1, sauf pour les perspectives .

Dans les paragraphes suivants , on définit pour chacune des transformations, la matrice associé

#### a) LES TRANSLATIONS

La translation du point p(x,y) par la translation T donne le point p'(x',y')

$$\begin{aligned}x' &= x + Tx \\ y' &= y + Ty\end{aligned}$$

Ce qui équivaut aux équations suivantes :

$$\begin{aligned}x' &= 1 * x + 0 * y + 1 * Tx \\ y' &= 0 * x + 1 * y + 1 * Ty \\ 1 &= 0 * x + 0 * y + 1 * 1\end{aligned}$$

Ce qui se traduit par

$$\begin{vmatrix} x' \\ y' \\ 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & Tx \\ 0 & 1 & Ty \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} x \\ y \\ 1 \end{vmatrix}$$

La colonne de droite de la matrice homogènes correspond au vecteur de translation

plan	espace
$\begin{vmatrix} 1 & 0 & Tx \\ 0 & 1 & Ty \\ 0 & 0 & 1 \end{vmatrix}$	$\begin{vmatrix} 1 & 0 & 0 & Tx \\ 0 & 1 & 0 & Ty \\ 0 & 0 & 1 & Tz \\ 0 & 0 & 0 & 1 \end{vmatrix}$

La transformation inverse est la translation - T

plan	espace
$\begin{vmatrix} 1 & 0 & -Tx \\ 0 & 1 & -Ty \\ 0 & 0 & 1 \end{vmatrix}$	$\begin{vmatrix} 1 & 0 & 0 & -Tx \\ 0 & 1 & 0 & -Ty \\ 0 & 0 & 1 & -Tz \\ 0 & 0 & 0 & 1 \end{vmatrix}$

#### b) LES CHANGEMENTS D'ECHELLE

Les termes diagonaux d'une matrice homogène correspondent aux facteurs d'échelle ( ces termes sont aussi utilisés par les rotations)

plan	espace
$\begin{vmatrix} Ex & 0 & 0 \\ 0 & Ey & 0 \\ 0 & 0 & 1 \end{vmatrix}$	$\begin{vmatrix} Ex & 0 & 0 & 0 \\ 0 & Ey & 0 & 0 \\ 0 & 0 & Ez & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$

Un changement d'échelle En par rapport à un centre d'homothétie c différent de l'origine nécessite un changement de repère permettant de ramener le point c à l'origine (la translation C ). Ce changement d'échelle se représente par une matrice qui résulte de la composition de la translation - C , du changement d'échelle à l'origine E et de la translation C .

Par exemple ,dans le plan ,  $En = (C) * (E) * (-C)$  soit

$$\begin{vmatrix} 1 & 0 & Cx \\ 0 & 1 & Cy \\ 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} Ex & 0 & 0 \\ 0 & Ey & 0 \\ 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} 1 & 0 & -Cx \\ 0 & 1 & -Cy \\ 0 & 0 & 1 \end{vmatrix} =$$

$$\begin{vmatrix} Ex & 0 & -Ex * Cx + Cx \\ 0 & Ey & -Ey * Cy + Cy \\ 0 & 0 & 1 \end{vmatrix}$$

### c) LES ROTATIONS

#### (1) rotation dans le plan

La matrice homogène de rotation d'un angle  $\theta$  autour de l'origine dans le plan est :

$$\begin{vmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

La transformation inverse est une rotation d'angle  $-\theta$  et se représente par la matrice suivante:

$$\begin{vmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

#### (2) rotation dans l'espace

Une matrice de rotation peut facilement être reconnue par le fait que son déterminant est unitaire. Les trois rotations de base autour des axes de coordonnées sont :

- axe Z :  $R_{z,\theta}$

$$\begin{vmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

- axe Y :  $R_{y,\theta}$

$$\begin{vmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

- axe x :  $R_{x,\theta}$

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

## B. TRANSFORMATION DE VISUALISATION ET FENETRAGE

### 1. INTRODUCTION

L'affichage de l'image d'un modèle implique la mise en correspondance des coordonnées des points et des lignes du modèle, avec les coordonnées appropriées du dispositif ou de la station de travail où l'image doit être visualisée. On passe d'un système de coordonnées à un

autre grâce aux procédures de transformations de coordonnées appelé ici "transformation de visualisation".

## 2. DEFINITION DES DIFFERENTS SYSTEMES DE COORDONNEES

-Le système de coordonnées absolues: C'est le système de coordonnées cartésiennes dans lequel on décrit les objets dont on veut afficher l'image .

-Le système des coordonnées du dispositif physique : il s'agit du système de coordonnées lié au dispositif sur lequel l'image de l'objet est affichée.

-Le système du dispositif normalisé : C'est le système de coordonnées de la surface d'affichage; cette surface est considérée comme un carré unité dont le coin inférieur gauche est l'origine du système de coordonnées

## 3. FENETRAGE ET CLOTURE

L'espace décrit par le système des coordonnées absolues est infini alors que la surface du dispositif de la visualisation est finie . On définit en général une région finie du système des coordonnées absolues appelée clôture. La clôture ainsi défini est une région finie de l'espace utilisateur: Elle correspond à l'espace que l'on veut visualiser.

La clôture peut être mise en correspondance directe avec la zone d'affichage ou avec un sous-ensemble de cette zone que nous appelons fenêtre .

En d'autre termes la clôture définit l'échelle des objets qui apparaissent dans la fenêtre

## 4. TRANSFORMATION DE NORMALISATION ET TRANSFORMATION DU DISPOSITIF PHYSIQUE

La transformation de normalisation N met en correspondance les coordonnées absolues avec les coordonnées du dispositif normalisé, mais la transformation du dispositif physique D fait correspondre les

coordonnées du dispositif normalisé aux coordonnées du dispositif physique (voir figure A1.3).

La transformation de visualisation (V) est donnée par  $N \cdot D$

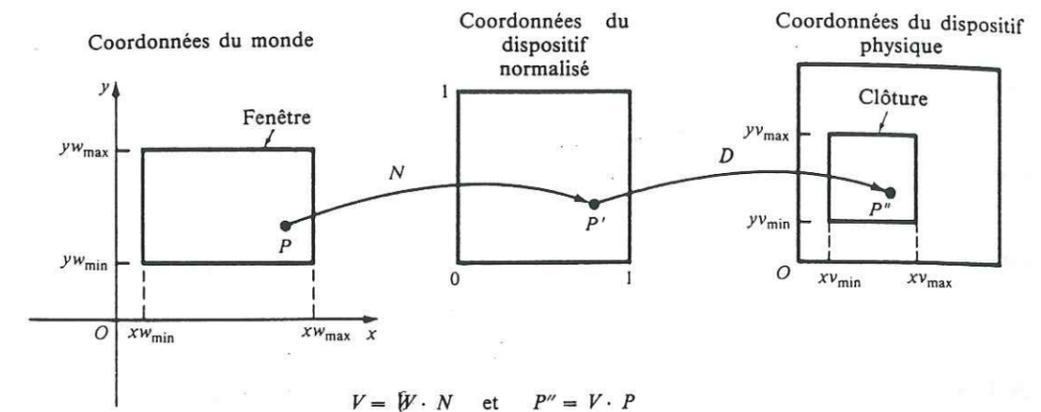


Figure A1.3 : la transformation de visualisation

(Plastock 87)

## 5. COORDONNES ECRAN-CLOTURE-NORMALISEES

On peut définir une fenêtre de deux façons:

-définir les coordonnées dans le repère écran de son coin supérieur droit ( $x_{csd}, y_{csd}$ ) et de son coin inférieur gauche ( $x_{cig}, y_{cig}$ )

-définir sa taille et les coordonnées de son centre

Supposons qu'on a un écran de  $x_{emax}$  pixels de largeur et de  $y_{emax}$  de hauteur

On peut déduire les coordonnées du centre de la fenêtre:

$$cx = x_{cig} + x_{csd} / 2$$

$$cy = y_{cig} + y_{csd} / 2$$

Les demies tailles  $s_x$  et  $s_y$  de la fenêtre sont exprimées par les expressions suivantes :

$$sx = (xc_{sd} - xc_{ig}) / 2$$

$$sy = (yc_{sd} - yc_{ig}) / 2$$

On peut donc définir un repère fenêtre dont l'origine est le centre  $c$  et dont les axes sont définis par les demis tailles  $x_{max} = sx$  et  $y_{max} = sy$  exprimées en nombre de pixels. Cependant, on peut exprimer les demis tailles en une unité de mesure, par exemple en centimètres :  $px$ ,  $py$  ou en coordonnées normalisées  $x_n$ ,  $y_n$  tels que  $x_n = 1$  pour  $px$  et  $y_n = 1$  pour  $py$  :

On peut dresser le tableau comparatif suivant :

coordonnées écran	fenêtre	normalisée
C ( $cx, cy$ )	(0,0)	(0,0)
A ( $cx+sx, cy$ )	( $px, 0$ )	( $px, 0$ )
B ( $cx, cy+sy$ )	( $0, py$ )	(0,1)
D ( $cx+sx, cy+sy$ )	( $px, py$ )	(1,1)
P ( $x_e, y_e$ )	( $x_p, y_p$ )	( $x_n, y_n$ )

$$\begin{aligned} sx & \text{ correspond à } px \\ x_e - cx & \text{ correspond à } x_p \quad > sx / (x_e - cx) = px / x_p \quad (1) \\ px & \text{ correspond à } x_n = 1 \\ x_p & \text{ correspond à } x_n \quad > px / x_p = 1 / x_n \quad (2) \end{aligned}$$

à partir de (1) et (2) on peut déduire que :  
 $sx / (x_e - cx) = 1 / x_n \Leftrightarrow sx * x_n = x_e - cx$  d'ou  
 $x_e = sx * x_n + cx$

Le même raisonnement nous conduit à l'expression suivante :  
 $y_e = sy * y_n + cy$

d'autre part on peut déduire à partir de l'expression (2) que :  
 $x_n = x_p / px$  et  $y_n = y_p / py$

## C. LES PROJECTIONS

### 1. INTRODUCTION

L'affichage des objets tridimensionnels est nécessairement plus complexe que le traitement des objets bidimensionnels. En effet, les dispositifs d'affichages (écran, feuille de papier) sont des domaines bidimensionnels ; il est alors nécessaire de transformer la représentation tridimensionnelle des objets en une représentation bidimensionnelle pour le traçage. On désire simuler en deux dimensions des objets tridimensionnels tels que la vision humaine les perçoit.

La transformation qui effectue le passage de l'espace 3D au plan contenant la surface d'affichage est une projection.

Toute projection de l'espace sur le plan peut être représentée par une matrice homogène d'ordre 4, caractérisée par un déterminant nul correspondant à la perte d'une dimension. De ce fait les projections ne sont pas réversibles.

Une projection géométrique plane est définie par un plan de projection et un centre de projection. Chaque point de l'objet étant projeté sur cette surface selon le rayon rectiligne issu du centre de projection. La projection transforme un segment de droite en un segment de droite.

### 2. LA CLASSIFICATION DES PROJECTIONS

Les projections géométriques planes se divisent en deux catégories suivant la position du centre de projection (figure A1.4):

- les projections parallèles
- les projections perspectives

Univ. J. Fourier - O.S.U.G.  
 MAISON DES GEOSCIENCES  
 DOCUMENTATION  
 B.P. 53  
 F. 38041 GRENOBLE CEDEX  
 Tél. 04 76 63 54 27 - Fax 04 76 51 40 58  
 Mail : ptalour@ujf-grenoble.fr

11 SEP. 2003

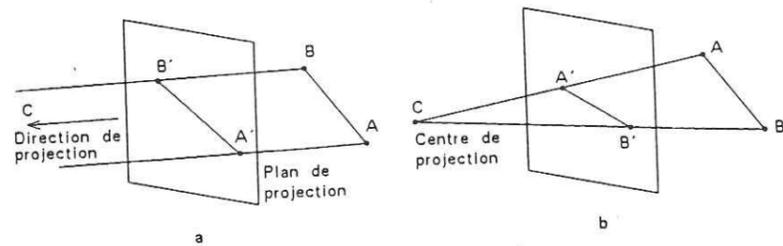


Figure A1.4: projections parallèle(a) et perspective (b) (Schweizer 87)

Le tableau suivant(voir figure A1.5) donne une topologie des familles de projections perspectives et parallèles

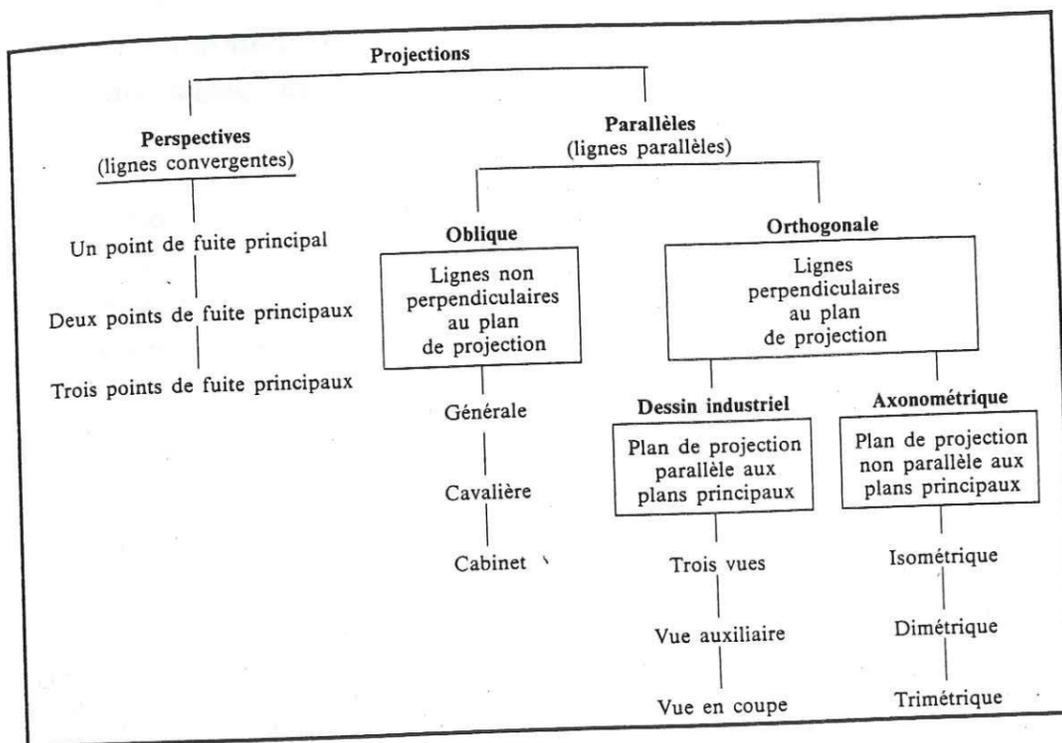


Figure A1.5(plastock 87)

a) LES PROJECTIONS PARALLELES

Une projection parallèle est définie par une direction de projection, le centre de projection est placé à l'infini . L'image d'un point est obtenue à l'intersection du plan de projection et de la ligne parallèle à la direction de projection et issue du point de l'objet . Les projections parallèles comportent deux sous-groupes :

- Les projections orthogonales
- Les projections obliques

(1) les projections orthogonales

Elles sont caractérisées par une direction de projection perpendiculaire au plan de projection et sont divisées en deux sous-groupe :

(a) les projections orthographiques

Ce sont les projections les plus utilisées . Une projection orthographique est caractérisée par une direction de projection parallèle à l'un des axes principaux (voir figure A1.6) . On obtient des vues de face ,de dessous ou de côté quand la direction de projection est parallèle respectivement à l'axe des x, des y ou des z

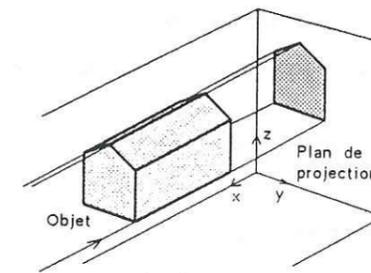


Figure A1.6 :projection par élimination d'une coordonnée.(Lucas 77)

On obtient donc l'une des trois vues en éliminant simplement une des coordonnées :

- élimination de z : vue de dessus
- élimination de y : vue de côté
- élimination de x : vue de face

(b) les projections axonométriques

Ce sont des projections orthogonales mais non orthographiques . Les plans de projections ne sont pas parallèles aux plans principaux . On distingue les projections suivantes :

- Les projections isométriques
- les projections dimétriques
- les projections trimétrique

(2) les projections obliques

Se sont des projections parallèles mais la direction de la projection n'est pas perpendiculaire au plan de projection

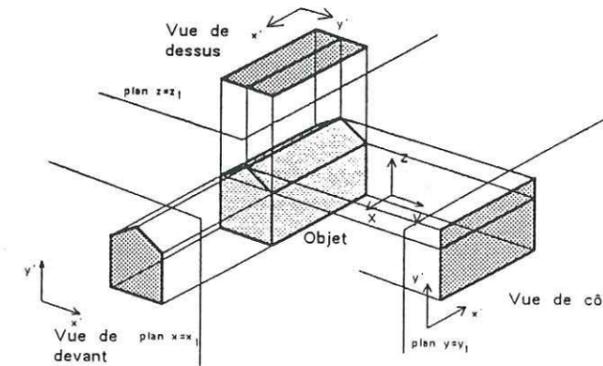
### b) LES PROJECTIONS PERSPECTIVES

Pour les projections perspectives , le centre de projection est placé à une distance finie du plan de projection , c'est la position de l'oeil de l'observateur . Les dessins en perspective sont caractérisés par le raccourci et les points de fuites . L'effet de raccourci est la réduction de la taille des objets en fonction de leurs éloignements , c'est aussi l'effet de profondeur . Cela augmente l'effet du réalisme .

Une perspective donne l'impression que des ensembles de droites parallèles semblent se rencontrer en un point , c'est le point de fuite . Les point de fuites principaux sont formés par la convergence apparente des lignes parallèles vers l'un des trois axes principaux x,y, ou z . Le nombre d'axes principaux intersectants le plan de projection définit le nombre des points de fuites principaux.

### 3. LES PROJECTIONS ORTHOGRAPHIQUES

Ces projections sont très simples à réaliser , elle sont basées sur le principe de la suppression d'une coordonnée permettant le passage de l'espace tridimensionnel au plan .Ces projections peuvent produire des vues de dessus, de face , ou de coté.



Vues utilisées en dessin technique

Figure 1A.7

#### a) VUE DE DESSUS

Cette vue correspond à la projection selon la direction de l'axe z, sur le plan de projection  $Z=0$ , donc la coordonnée z de tout point devient égale à zéro, les coordonnées x et y restent inchangées : on élimine la coordonnée z.

La matrice homogène correspondante est:

$$M_{\text{orth},z} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

#### b) VUE DE COTE

Cette vue correspond à la projection selon l'axe  $y$  sur le plan  $Y=0$ , sa matrice homogènes est

$$M_{\text{orth},y} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

### c) VUE DE FACE

Cette vue est une projection sur le plan  $X=0$  selon l'axe  $x$ . La matrice correspondante est:

$$M_{\text{orth},x} = \begin{vmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

## 4. LES PROJECTIONS PERSPECTIVES

### a) INTRODUCTION

La perspective produit un meilleur effet de réalisme que les autres projections. Cela est dû à l'effet de profondeur correspondant à notre vision naturelle.

Contrairement aux projections parallèles qui suivent les règles géométriques affines, les perspectives ont une géométrie spéciale, en effet, deux lignes parallèles ne sont plus parallèles après projection et la longueur d'un segment est réduite proportionnellement à sa distance à l'observateur. On ne peut alors plus extraire aucune information dimensionnelle d'une vue perspective.

La projection d'un point se calcule comme toute transformation en multipliant des coordonnées par une matrice d'ordre 4 dont les trois premiers termes de la dernière ligne correspondent à la perspective.

### b) CAS PARTICULIER

La projection perspective standard est illustrée dans la figure figure 1A.8

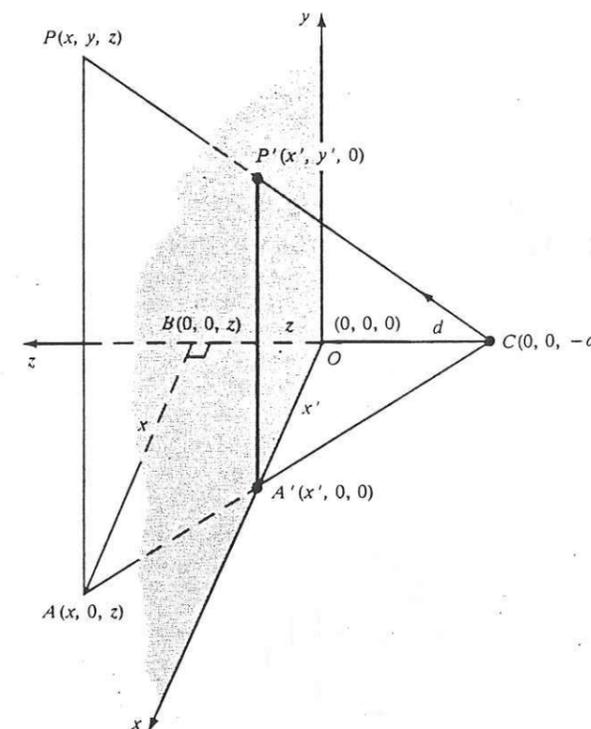


Figure A1.8 (Plastock 87)

Le point objet  $p$  est placé aux coordonnées absolues en  $(x,y,z)$ , le problème est de déterminer les coordonnées  $(x',y',z')$  du point image  $p'$ . Le plan de projection est le plan  $xy$  et le centre de projection est le point  $c(0, 0, -d)$  sur les  $z$  négatifs. Les triangles semblables  $ABC$   $A'OC$  nous permettent d'écrire  $x' = d \cdot x / (z+d)$ ,  $y' = d \cdot y / (z+d)$ ,  $z' = 0$

La transformation perspective d'un objet en son image n'est pas linéaire et ne peut donc pas être représentée par une matrice de transformation d'ordre 3. Cependant, en utilisant les coordonnées homogènes, la transformation perspective peut être représentée par une matrice d'ordre 4:

$$\begin{vmatrix} x' \\ y' \\ z' \\ 1 \end{vmatrix} = \begin{vmatrix} d * x \\ d * y \\ 0 \\ z + d \end{vmatrix} = \begin{vmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & d \end{vmatrix} * \begin{vmatrix} x \\ y \\ z \\ 1 \end{vmatrix}$$

c) TRANSFORMATION PERSPECTIVE SELON UN AXE PRINCIPAL

- Le modèle de la chambre noire

On considère le modèle de la chambre noire. C'est une simple boîte parallélépipédique qui sert à capter des images 2D à partir des objets 3D (voir figure A1.9). L'effet de perspective découle du principe de la propagation rectiligne des rayons lumineux : Pour tout point d'un objet, seuls les rayons passant par l'ouverture de la boîte noire peuvent atteindre son fond (plan de projection  $Z=L$ ). L'image obtenue est une projection perspective mais elle est inversée

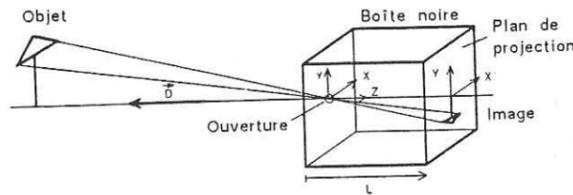
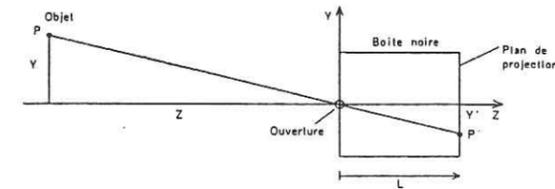


Figure A1.9: le principe de la boîte noire  
(Schweizer 87)

On choisit un repère direct à l'ouverture, l'axe Z est colinéaire avec la direction de vision. Il s'agit d'exprimer ici la projection perspective d'un point.

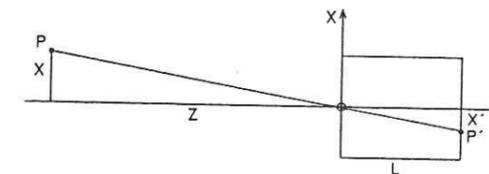
Un point P de coordonnée  $(x=0, y)$  a pour projection le point  $p'$  dont les coordonnées sont facilement déterminées à l'aide des triangles semblables dans une vue en coupe selon le plan  $x=0$  (voir figure A1.10) on a  $y' = y * L/z$



Vue en coupe selon le plan  $X=0$

Figure A1.10

Pour un point de coordonnées  $(x, y=0)$ , on applique le même raisonnement dans le plan  $y=0$  (voir figure A1.11) on a  $x' = x * L/z$



Vue en coupe selon le plan  $Y=0$

Figure A1.11

On peut imaginer que l'écran, c'est à dire le plan de projection, est une vitre placée entre l'observateur et l'objet (voir figure A1.12). La projection perspective est donc simplement l'image perçue par l'observateur à travers la vitre

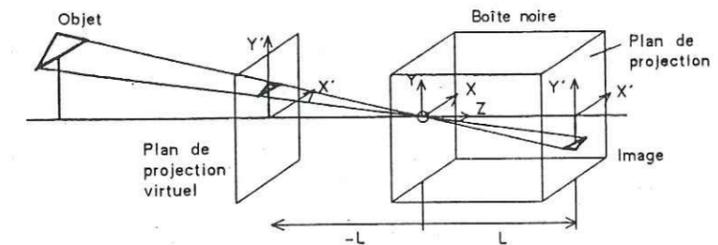


Figure A1.12: redressement de l'image  
(Schweizer 87)

Les mêmes relations sont encore vraies mais le plan de projection est devenu  $Z = -L$ . Avec ces conventions, on obtient un premier modèle de la perspective selon l'axe Z

Centre de projection:  $[0 \quad 0 \quad 0 \quad 1]$   
 direction de vision :  $[0 \quad 0 \quad -1 \quad 0]$   
 plan de projection :  $Z = -L$   
 point :  $[x \quad y \quad z \quad 1]$   
 projection :  $[x' \quad y' \quad -L \quad 1]$

$$x' = x * -L/z \quad y' = y * -L/z \quad z' = -L$$

Si on exprime toutes les coordonnées dans le repère associé au plan de projection (voir figure A1.13), la perspective sera décrite comme suit:

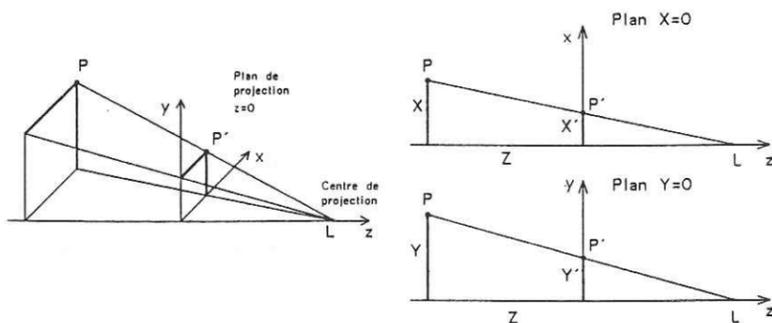


Figure A1.13 : les triangles semblables (Schweizer 87)

Centre de projection:  $[0 \quad 0 \quad L \quad 1]$   
 direction de vision :  $[0 \quad 0 \quad -1 \quad 0]$   
 plan de projection :  $Z = 0$   
 point :  $[x \quad y \quad z \quad 1]$   
 projection :  $[x' \quad y' \quad 0 \quad 1]$

$$x' = x * -L/z-L \quad \langle \rangle \quad x' = x * 1/(z/-L + 1) \quad : (1)$$

$$y' = y * -L/z-L \quad \langle \rangle \quad y' = y * 1/(z/-L + 1) \quad : (2)$$

$$z' = 0 \quad : (3)$$

d) MATRICE DE TRANSFORMATION PERSPECTIVE

La matrice homogène qui correspond à la transformation perspective exprimée par les équations (1) (2) (3) est

$$M_{per,z} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & r & 1 \end{vmatrix} \quad \text{avec } r = 1/-L$$

$$p' = M_{per,z} P$$

Seul le terme d'indice w,z différencie cette matrice de la matrice unité, il exprime donc la perspective, dont la direction de vision est l'axe Z. Cette matrice correspond à une transformation perspective et non à la projection perspective

e) MATRICE DE PROJECTION PERSPECTIVE

Pour obtenir une projection, Il faut multiplier la matrice de la transformation perspective par la matrice de la projection orthographique, cela revient à éliminer la coordonnée z du résultat de la transformation perspective :

$$M_{proj,z} = M_{orth,z} * M_{per,z} =$$

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & r & 1 \end{vmatrix} =$$

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & r & 1 \end{vmatrix}$$

On a donc  $p' = M_{proj,z} * P$

$$\begin{vmatrix} x' \\ y' \\ z' \\ w' \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0-1/L & 1 & 1 \end{vmatrix} * \begin{vmatrix} x \\ y \\ z \\ w \end{vmatrix} \langle \rangle \begin{vmatrix} x' \\ y' \\ z' \\ w' \end{vmatrix} \begin{vmatrix} x \\ y \\ 0 \\ (-z/L)+w \end{vmatrix}$$

Ce résultat est obtenu en coordonnées homogènes, il doit être converti en coordonnées cartésiennes

$$x'' = x'/w' = x^* 1/(-z/L+w)$$

$$y'' = y'/w' = y^* 1/(-z/L+w)$$

$$z'' = 0$$

Pour  $w = 1$ , on retrouve les équations de la perspective

## BIBLIOGRAPHIE

Bailly .C, Challine.J.F, Ferri .H.C , Gloess P.Y ,Marcheson .B : Les langages orientés objets ; Cepadues édition 1987.

Bertin .J : La graphique et le traitement graphique de l'information ; Flammarion, Nouvelle Bibliothèque Scientifique, Paris, 1977.

Booch .G : Ingénierie du logiciel avec ADA , de la conception à la réalisation ; Interédition ,paris 1988.

Boule . PH :étude et réalisation d'algorithme pour la visualisation de scènes composées de facettes planes ; thèse à l'Institut National polytechnique de Grenoble (1980).

Buisson J.L :Le fichier des données géotechnique . B.L.L.P.C , N° 84 juil-aout 76 ref 1905 ; p 149-162.

Bulletin de Liaison des Laboratoires des ponts et chaussées ; numéro spécial : logiciels(production,exploitation,maintenance) Novembre 1974

Colas .G : Stockage des données géologiques et géotechniques ; B.L.L.P.C N° 56 Dec 71 .

Cua Charles : Amélioration des Maillages par des Méthodes de sous-Gradient ; Thèse I.N.P.G ,Grenoble 1985

Damak M., Dayre M. : Visualisation, par Microordinateur, des données géologiques et géotechniques du sous-sol ; Premières Journées Tunisiennes de Géologie Appliquée, Sfax - Tunisie , Juin 1984 .

De Ragueneil .A : Une expérience de géotechnique sur ordinateur B.L.L.P.C , N° 67 sept-oct 73 ref 1332 .

Durand .M : La micro-informatique comme aide à la création des cartes géologiques Bulletin de l'Association Internationale de Géologie De l'Ingénieur .

Fabre .D, Lettelier.J , Mercieca .G : Essai de constitution et d'utilisation d'une banque de données sur le sous-sol d'une plaine alluviale (région de Grenoble, France) . Revue française de la géotechnique ,N° 13 Nov 1980 .

Franco P.Preparata, Michael Ian Shamos : Computational Geometry -An Introduction ; Springer-Verlag New York Inc 1985 (390 pages)

Galtier B : La Cartographie Automatique dans les sciences de la terre ; informatique dans les sciences de la terre pour la caractérisation des ressources naturelles . Revue sciences de la terre ; informatique géologique N° 20 1984 ; part 1 . P 309-320

Grelot J-P : Avantages et Inconvénients de la Cartographie Assistée par Ordinateur . Actes de Micad'85 sur la C.F.A.O et l'infographie;4° conférence européenne P. 70-78 .

Gros. G : La console de visualisation , outil de représentation et de traitement des données géologiques d'un site . B.L.L.P.C , N° 76 ref 1603.

Guiennet .B : Le fichier géologique et géotechnique du laboratoire d'Aix-en-provence , B.L.L.P.C , N° 80, p.163-171.

Guillemin .R , Ducloux .A : Application du dessin automatique au laboratoire d'Autun ; B.L.L.P.C ; N° 80 , Nov-Dec 75 ref 1702.

Havard.H , Mignot . N , Potdevew .M : Suggfrid pour la gestion sur ordinateur d'un fichier géotechnique de granulats, B.L.L.P.C; N° 84 juillet aout 76 ref 1870

Hermeline . F : Triangulation automatique d'un polyèdre en dimension N ; R.A.I.R.O - Numerical Analysis 16 1982 ; pages 211-242 .

Laffitte . P :traité d'informatique géologique ; Masson 1972 ; 620 pages

Lettelier .J : Synthèse cartographique géotechnique de la région grenobloise . Documentation graphique et traitement informatique des données .,Thèse à l'Université Scientifique et Médicale de Grenoble ; 1981; 236 p.

Lucas .M : contribution à l'étude des techniques de communication graphique avec un ordinateur - élément de base des logiciels graphiques interactifs ;thèse à I.N.P.G (1977)

Lucas .M : La réalisation des logiciels graphiques interactifs; Travaux dirigés de l'école d'été d'informatique tenue au bréau-sans-nappe du 7 au 27 juillet 1979 Eyrolles

Mahieu J.L : Essai méthodologique d'utilisation d'un fichier de données géotechniques . Documentation , cartographie, traitement statistiques . B.L.L.P.C , N° 76 , p. 123-131.

Mallet J.L : préparation d'un ensemble de méthodes de la cartographie automatique numériques. sciences de la terre, informatique géologique N4-octobre 1974 .

Maranda .R : établissement des cartes géotechniques au ministère des richesses naturelles du Québec . Direction, générale des mines , Québec 1977.

Martinez Francis : La Synthèse d'Image -concepts,matériels et logiciels; éditest paris 1984

Mercieca .G : Constitution d'un fichier géologique et géotechnique sur ordinateur , application à l'étude du remblaiement alluvial de la vallée de l'Isère. Thèse 3° cycle, Grenoble ,1977, 153 p

Mignot.Y, Lorain . J-M : Etablissement d'un fichier géologique des zones urbaines B.L.L.P.C ,N° 50 , Jan-Fev 1971 .

Newman .W, Sproull .R.F : Principle of Interactive Computer Graphics ; Mc Graw Hill , New York ,Second Edition, 1979.

Palen .E : Correlation Entre Sondages de Diagraphie ; informatique dans les sciences de la terre pour la caractérisation des ressources naturelles . Revue sciences de la terre ; informatique géologique N° 20 1984 ; part 1 . P 277-294.

Plastock .R.A ,Gordon Kalley: infographie cours et problèmes série Schaum ; édition McGraw-Hill 1987

Sanejouand. R .: La cartographie géotechnique en France . Ministère de l'équipement et du logement .(D.A.F.U),1972 ; 95 p.

Sanejouand. R : Fichier de données géotechnique sur ordinateur dans les laboratoires des ponts et des chassées . B.L.L.P.C ;N° 76 p. 107-116.

Savoysky .S, Guillon M.P :L'emploi de l'informatique pour la gestion des données B.L.L.P.C N° 83 , Mai-Juin 76 ref 1869 p. 105-112 .

Scholl P.C, Lucas.M, Peyrin.J.P : Algorithmique et Représentation des Données ; Vol 1,3 Masson 1983

Schweizer Philippe : infographie ;2 volumes; collection informatique ; édition :Presses polytechniques normandes 1987

Talon Jean-Yves : Algorithmes de Génération et d'Amélioration de Maillages en 2 Dimensions ; Atelier de Recherche sur les Techniques Mathématiques et Informatiques des Systèmes ; Rapport technique 20;Mars 1987

Ussegloi Polatera J-M : Contribution à la cartographie géotechnique de la ville de paris. Etude de la butte de belleville , these Paris 6 , Mai 83.

William R.Mallgren :Formal Spécification of Graphic Data Types; ACM Transactions on Programmng Languages and systems , Vol 4, N° 4, October 1982, Pages 687-710.

## TABLE DE MATIERE

I. INTRODUCTION	1
II. LE CONTEXTE HISTORIQUE	4
III. LA METHODOLOGIE DE DEVELOPPEMENT LOGICIEL.	18
A. INTRODUCTION	18
B. LA METHODE ORIENTE OBJET	19
C. "LA METHODOLOGIE ORIENTEE OBJET"	20
1. DEFINIR LE PROBLEME : spécification et analyse (s.a.d.t)	21
2. LA DEFINITION DES ENTITES DU MONDE REEL	22
3. LA CONCEPTION ORIENTE OBJET	22
IV. DEFINITION D'UN SYSTEME GENERAL	24
A. SPECIFICATION GENERALE DU SYSTEME:CAHIER DES CHARGES	24
1. CONTEXTE SCIENTIFIQUE ET ENVIRONNEMENT DU SYSTEME	24
a) PRINCIPES GENERAUX DE LA CARTOGRAPHIE GEOTECHNIQUE	24
(1) définition	24
(2) utilité	24
(3) contenu d'un dossier géotechnique	25
b) DESCRIPTION DE L'INFORMATION BRUTE NECESSAIRE A LA PRODUCTION D'UN DOSSIER GEOTECHNIQUE	25
c) L'INFORMATISATION DU PROCESSUS	26
2. PRINCIPALES FONCTIONNALITES DU SYSTEME	27
a) LA REALISATION D'UNE BASE DE DONNEES	27
b) L'ESTIMATION ET LE CALCUL	27
(1) l'interpolation et l'estimation	27
(2) le calcul	27
c) LE DESSIN ET LA PRODUCTION DES GRAPHIQUES	28
(1) les documents cartographiques	28
(2) les coupes graphiques de sondage	29
(3) les profils géologiques	29
(4) les blocs diagrammes	29
3. L'INFORMATION MANIPULEE PAR LE SYSTEME	29
a) L'INFORMATION BRUTE	29
b) LES DONNEES GRAPHIQUES	30
c) L'INFORMATION INTERMEDIAIRE	30
4. CONTRAINTES ET EXIGENCES	30
a) LES CONTRAINTES	30
(1) la nature sémantique de l'information	30
(2) l'utilisation d'un micro-ordinateur de type PC	31
(3) l'utilisation du turbo pascal	32
(4) l'utilisation d'un S.G.B.D relationnel	32
(5) contrainte de place mémoire	32
b) LES EXIGENCES	32
(1) la transparence du logiciel	32
(2) la portabilité du système	33
(3) la maintenance et l'adaptabilité du système	33
(4) l'interactivité du logiciel	33
(5) l'ergonomie	34
5. LA PRIORITE DE REALISATION	34
B. ANALYSE GENERALE DES FONCTIONNALITES	35
1. PRESENTATION DE L'ANALYSE FONCTIONNELLE	35
a) INTRODUCTION	35
b) LA METHODE S.A.D.T	35
(1) présentation	35

(2) justification	35
2. DECOMPOSITION FONCTIONNELLE DU SYSTEME	37
a) MANIPULER LA BASE DE DONNEES	37
(1) but	38
(2) principe	38
(a) Les besoins de l'utilisateur	39
(b) besoin du système	39
(3) contrôle	40
(4) mécanismes	40
b) DESSINER	40
(1) produire un dessin	40
(a) but	41
(b) principe	41
(c) dessiner une carte	42
i) dessiner une carte d'isovaleur	42
ii) dessiner une carte lithologique	42
(d) dessiner un profil géologique	43
(e) réaliser un bloc-diagramme	43
(2) manipuler un dessin	44
(3) modifier un dessin	44
(a) but	44
(b) principe	44
c) ESTIMER	44
(1) but	44
(2) principe	45
d) INTERPOLER, LISSER, CALCULER	45
(1) but	45
(2) principe	45
C. LA DEFINITION DES ENTITES REELLES : LE MODELE ENTITE-ASSOCIATION	46
1. LA PRESENTATION DU MODELE ENTITE-ASSOCIATION	46
a) LA DEFINITION DU MODELE	46
b) LA JUSTIFICATION	47
(1) approche relationnelle de base de données	47
(2) passage à la méthode orientée objet	47
c) L'APPLICATION SELECTIVE	47
2. L'ETUDE DE L'INFORMATION COLLECTEE LORS D'UNE CAMPAGNE DE RECONNAISSANCE	49
3. L'APPLICATION DU MODELE	49
a) LA DETERMINATION DES ENSEMBLES D'ENTITES COMPOSANT LE SYSTEME D'INFORMATION	49
b) LES ASSOCIATIONS ENTRE ENTITES	53
(1) définition	53
(2) notation	54
(3) présentation	54
c) LE SCHEMAS-Z GENERALISE	62
V. LA METHODE ORIENTEE OBJET : DOSSIER DE CONCEPTION	65
A. PRESENTATION DE LA METHODE ORIENTEE OBJET	65
B. SPECIFICATION DES OBJETS GRAPHIQUES	66
1. LES DEFINITIONS DE BASE	66
a) LA REGION	66
b) L'IMAGE	67
c) LA TRANSFORMATION GRAPHIQUE	68
d) LA STRUCTURE HIERARCHIQUE DE L'IMAGE	69
2. LA SPECIFICATION DES OBJETS GRAPHIQUES	70
C. SPECIFICATION DES OBJETS DE MODELISATION	75
1. UN DOMAINE	75
2. LA TRIANGULATION	75

3. LE MAILLAGE	76
4. L'ENVELOPPE CONVEXE	76
5. LA COURBE	76
6. LA CARTE ISO	77
D. SPECIFICATION DES OBJETS GEOLOGIQUES	77
1. LA COUCHE	77
2. LE FORAGE	78
3. LE TERRAIN	79
4. LA REPARTITION LITHOLOGIQUE	80
5. UN PROFIL GEOLOGIQUE	81
6. UN BLOC-DIAGRAMME	81
E. CONCLUSION	82
VI. LA REALISATION DES PROCEDURES DE VISUALISATION	83
A. LA REALISATION DE LA PROJECTION PERSPECTIVE GENERALE	84
1. LE PRINCIPE DE L'ALGORITHME	84
2. LE PASSAGE DU REPERE OBJET AU REPERE OBSERVATEUR	86
3. LA REALISATION DE L'ALGORITHME	90
4. INFLUENCE DE LA POSITION DE L'OBSERVATEUR	91
B. LA REALISATION DES PROJECTIONS ORTHOGONALES	95
C. LES ETAPES DE LA VISUALISATION D'OBJETS TRIDIMENSIONNELS	98
D. LE DECOUPAGE ET LA NORMALISATION EN TROIS DIMENSIONS	100
00	
1. INTRODUCTION	100
2. DEFINITION DU VOLUME DE VISUALISATION	100
3. FENETRAGE AVEC UN VOLUME DE PROJECTION FINI	101
4. DECOUPAGE ET FENETRAGE	101
5. DEFINITION DU SYSTEME DE COORDONNEES NORMALISEES 3D	102
6. ELIMINATION D'UN POINT CACHE	102
7. DEFINITION DE LA MATRICE DE NORMALISATION 3D	103
8. ALGORITHME DE DECOUPAGE EN 3D	106
a) DECOUPAGE POUR UNE PROJECTION PERSPECTIVE	106
(1) découpage d'un point	106
(2) découpage de segment de droite	107
-la représentation paramétrique:	108
(3) l'algorithme de découpage	108
b) DECOUPAGE POUR UNE PROJECTION ORTHOGONALE	108
E. L'ELIMINATION DES PARTIES CACHEES	109
1. INTRODUCTION	109
2. ETUDE ET CLASSIFICATION DES ALGORITHMES	110
a) ) Z-BUFFER	112
b) LIGNE DE BALAYAGE	113
c) SUBDIVISION	114
d) PEINTRE	114
3. CHOIX DE L'ALGORITHME	115
4. L'ADAPTATION DE L'ALGORITHME DE N.N.S	118
a) LE PRINCIPE DE L'ALGORITHME	118
c) AFFECTATION DES PRIORITES	119
d) LE TEST DES FACES CACHEES	119
e) L'ALGORITHME GENERAL	121
f) L'ANALYSE ET LA REALISATION DE L'ALGORITHME	122
(1) le modèle de description	122
(a) description des éléments	122
(b) représentation des éléments traités	122

(2) tri initial de données	123
(3) les opérations géométriques utilisées	123
(4) la fonction stratégie	123
(5) l'ensemble des représentations intermédiaires	124
(6) le modèle de visibilité	124
VII. LE DOSSIER DE REALISATION DE LA BIBLIOTHEQUE GRAPHIQUE	127
A. INTERET DE LA BIBLIOTHEQUE GRAPHIQUE	127
B. PRESENTATION DES MODULES	128
1. LE MODULE POINT	131
2. LE MODULE DROITE	132
3. LE MODULE VECTEUR	134
4. LE MODULE MATRICE	134
5. LE MODULE RECTANGLE	136
6. LE MODULE TRANSFORMATION	138
7. LE MODULE POLYGONE	142
8. LE MODULE VOLUME	148
9. LE MODULE D'ELIMINATION DE SURFACES CACHEES	148
10. LE MODULE DES DECLARATIONS GLOBALES	151
11. LE MODULE DES OUTILS	153
VIII. LA MODELISATION GEOMETRIQUE DES DONNEES	153
A. LA MODELISATION DES VOLUMES	153
1. SURFACE PLANE - SURFACE MATHEMATIQUE	154
2. LA MODELISATION	154
a) LE PRINCIPE DE LA MODELISATION	155
b) LE SCHEMA	156
3. LA REALISATION DE LA MODELISATION	158
B. LA TRIANGULATION	158
1. CRITIQUE DE LA METHODE DES RESEAUX REGULIERS	159
2. TRIANGULATION ET MAILLAGE EN ELEMENTS FINIS	160
3. LES DIFFERENTES METHODES DE TRIANGULATIONS	161
a) LA METHODE TRIVIALE DES SEGMENTS	161
b) LA METHODE DE VORONOI	161
c) AUTRES METHODES	162
4. c) PRESENTATION ET AMELIORATION DE L'ALGORITHME CHOISI	163
a) L'ALGORITHME DE TALON	163
(1) Algorithme de construction de D sans points intérieurs:	163
(a) présentation	163
(b) performances	164
(c) simplification	165
(2) Algorithme de triangulation d'un domaine D muni de la discrétisation Vint :	165
(a) principe	165
(b) présentation	166
(c) performance	167
(d) amélioration	167
i) La première boucle	167
ii) La deuxième boucle	169
C. L'ALGORITHME DE LA CARTOGRAPHIE	170
D. LA REALISATION DES PROFILS GEOLOGIQUES	171
1. LE PROBLEME DE CORRELATION	171
2. L'ALGORITHME	172
E. LES BLOCS-DIAGRAMMES	174
1. LES BLOCS-DIAGRAMMES RELIEF	174
a) CAS DE MAILLE REGULIERE	174
b) CAS D'AMAS IRRÉGULIER:	176

2. LES BLOCS-DIAGRAMMES TERRAIN	177
IX. L'ARCHITECTURE LOGICIELLE	184
A. ARCHITECTURE DU LOGICIEL GRAPHIQUE	184
1. L'UNITE DE COMMUNICATION	185
2. L'UNITE DE DESCRIPTION ET DE VISUALISATION	185
3. L'UNITE DE CONTROLE	186
B. L'ARCHITECTURE DU LOGICIEL GENERAL	186
X. CONCLUSION GENERALE	188
XI. ANNEXE A : PRESENTATION DES TECHNIQUES GRAPHIQUES	190
A. LES TRANSFORMATIONS GRAPHIQUES	190
1. INTRODUCTION	190
2. LES TRANSFORMATIONS LINEAIRES	190
a) LA TRANSFORMATION LINEAIRE D'UN POINT	190
b) LA NOTATION MATRICIELLE	191
c) LES PROPRIETES GEOMETRIQUES	191
3. LA MATRICE HOMOGENE	191
4. LES COORDONNEES HOMOGENES	193
5. LES TRANSFORMATIONS GEOMETRIQUES ET LES TRANSFORMATIONS DE COORDONNEES	193
6. LES TRANSFORMATIONS GEOMETRIQUES EN COORDONNEES HOMOGENES	195
a) LES TRANSLATIONS	196
b) LES CHANGEMENTS D'ECHELLE	197
c) LES ROTATIONS	198
B. TRANSFORMATION DE VISUALISATION ET FENETRAGE	199
1. INTRODUCTION	199
2. DEFINITION DES DIFFERENTS SYSTEMES DE COORDONNEES	200
3. FENETRAGE ET CLOTURE	200
4. TRANSFORMATION DE NORMALISATION ET TRANSFORMATION DU DISPOSITIF PHYSIQUE	200
5. COORDONNES ECRAN-CLOTURE-NORMALISEES	201
C. LES PROJECTIONS	203
1. INTRODUCTION	203
2. LA CLASSIFICATION DES PROJECTIONS	203
a) LES PROJECTIONS PARALLELES	205
(1) les projections orthogonales	205
(a) les projections orthographiques	205
(b) les projections axonométriques	206
(2) les projections obliques	206
b) LES PROJECTIONS PERSPECTIVES	206
3. LES PROJECTIONS ORTHOGRAPHIQUES	207
a) VUE DE DESSUS	207
b) VUE DE COTE	207
c) VUE DE FACE	208
4. LES PROJECTIONS PERSPECTIVES	208
a) INTRODUCTION	208
b) CAS PARTICULIER	209
c) TRANSFORMATION PERSPECTIVE SELON UN AXE PRINCIPAL	210
d) MATRICE DE TRANSFORMATION PERSPECTIVE	213
e) MATRICE DE PROJECTION PERSPECTIVE	213

Thèse de Doctorat de l'Université Joseph Fourier  
Grenoble I

*Titre de l'ouvrage :*

**UN LOGICIEL DE STOCKAGE, DE TRAITEMENT ET  
DE VISUALISATION GRAPHIQUE ET  
CARTOGRAPHIQUE DES DONNEES GEOLOGIQUES  
ET GEOTECHNIQUES**

*Title :*

**SOFTWARE DEVELOPMENT FOR STORAGE,  
TREATMENT, MAPPING AND GRAPHIC DISPLAY OF  
GEOLOGICAL AND GEOTECHNICAL DATA  
ABSTRACT**

*Nom de l'auteur :* Mohamed DAMAK

*Etablissement :*

**Institut de Recherches Interdisciplinaires de Géologie et de  
Mécanique**

### RESUME

Cette thèse présente un travail de spécification, de conception et de réalisation partielle d'un logiciel de stockage, de traitement et de visualisation graphique et cartographique des données géologiques et géotechniques.

Nous avons accordé dans cette étude autant d'importance aux méthodes d'analyse et de conception qu'à la réalisation du système. En effet, nous avons établi la spécification en utilisant la méthode S.A.D.T. (System Analysis Data Thecnic) qui nous a permis de constituer un dossier d'analyse de fonctionnalités du système.

Par ailleurs, l'application du modèle entité-association nous a conduit à la définition des entités correspondant aux données géologiques ainsi que les relations entre ces entités.

La conception du système a été basée sur la méthode orientée objet ce qui permet plus d'adaptabilité et de portabilité.

La priorité a été donnée à la réalisation d'un sous-logiciel de modélisation géométrique des données et de leur visualisation tridimensionnelle. La modélisation a été basée sur un choix adéquat des structures de données. Une étude des algorithmes de triangulation et d'élimination des parties cachées a été réalisée, les algorithmes choisis ont été améliorés et adaptés à l'application présente.

### MOTS CLES

Elimination des surfaces cachées ; Entités-Associations ; Visualisation-Tridimensionnelle ; Cartographie; Méthode orientée objet ; Géologie et géotechnique ; Méthode d'analyse et de conception.