



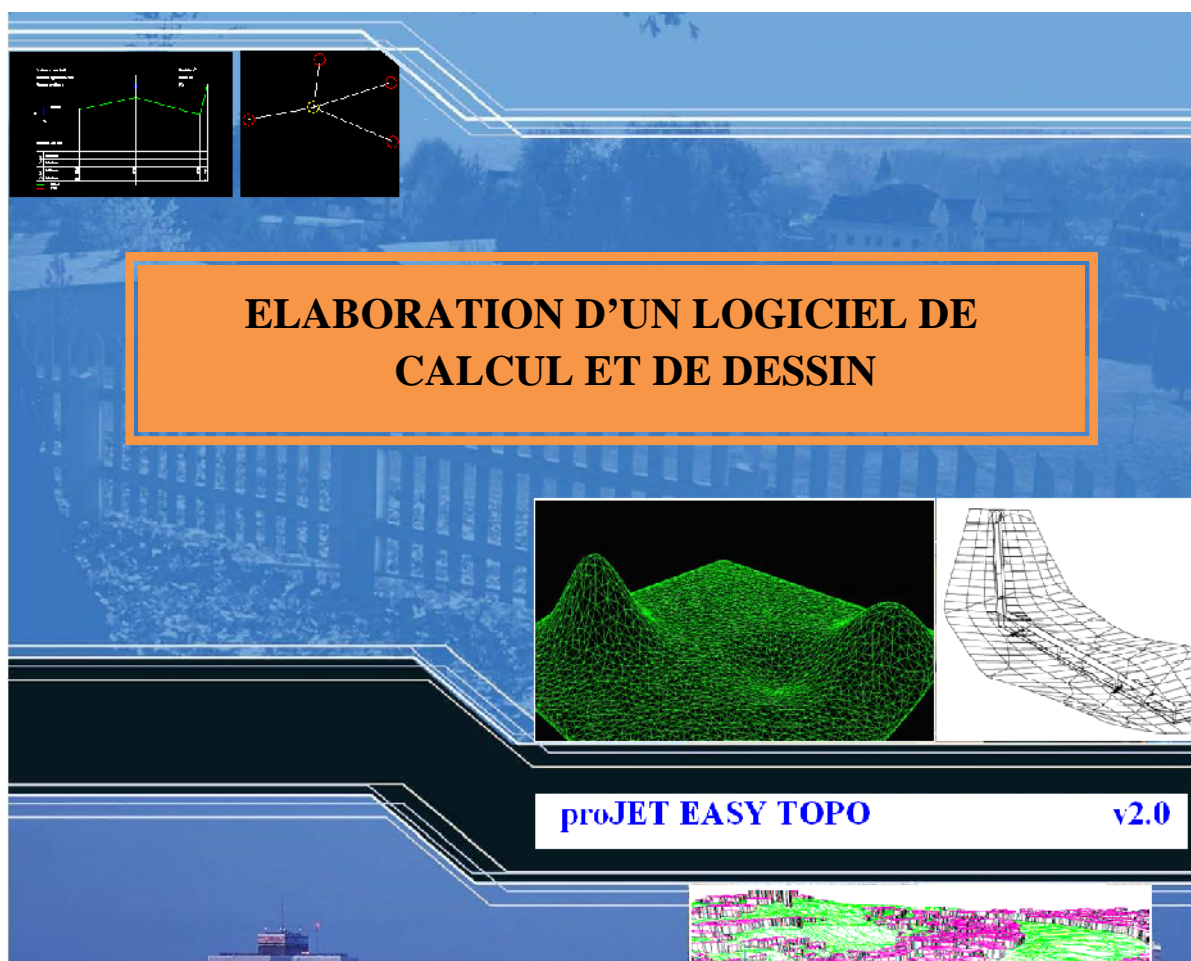
UNIVERSITE D'ANTANANARIVO

Ecole Supérieure Polytechnique d'Antananarivo

Département Information Géographique et Foncière



Mémoire de fin d'étude en vue de l'obtention de diplôme d'ingénieur Géomètre-Topographe



Présenté et soutenu par :

RASAMOELY Aina Natolotra Jonathan

Encadré par :

Monsieur RABETSIAHINY

Monsieur RABEMALAZAMANANA

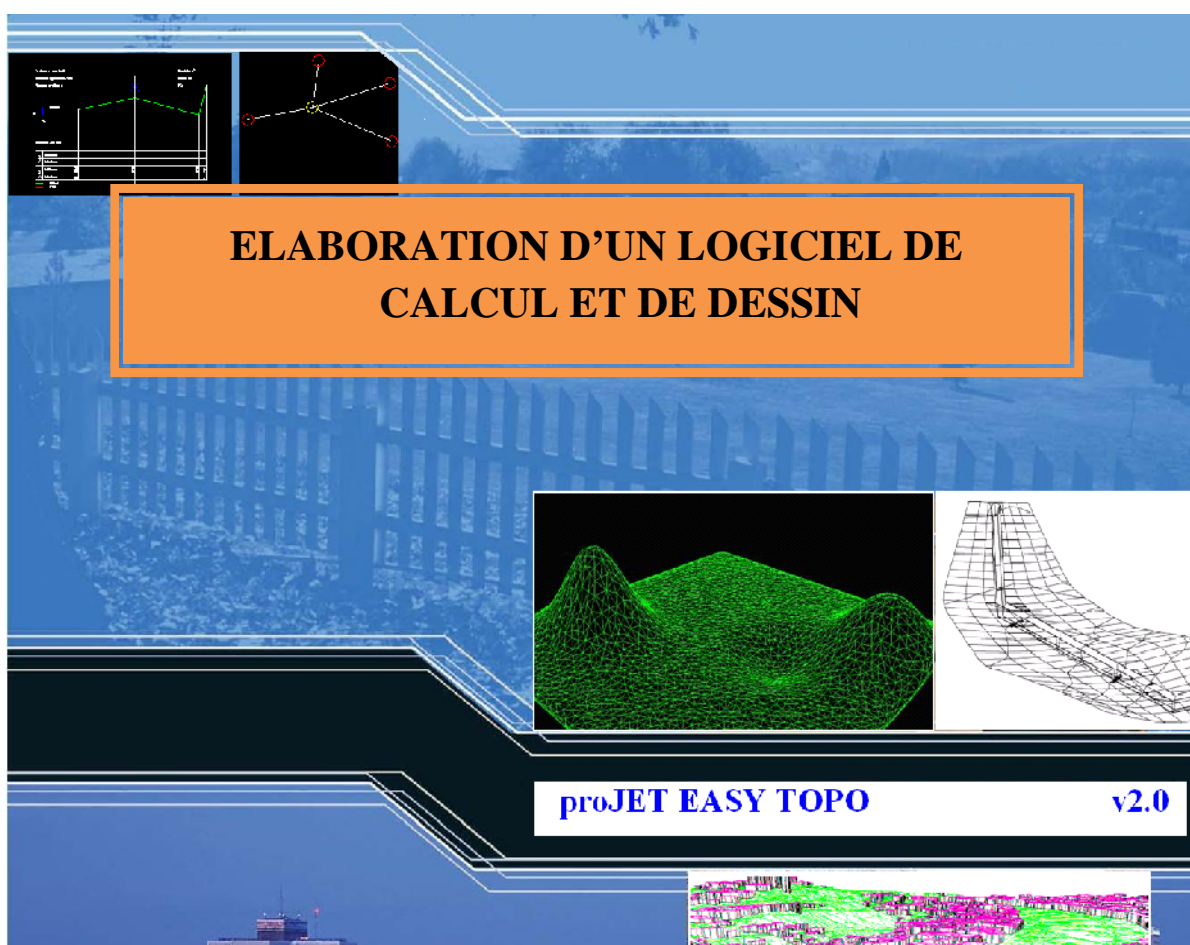
Promotion 2010



Ecole Supérieure Polytechnique d'Antananarivo
Département Information Géographique et Foncière



Mémoire de fin d'étude en vue de l'obtention de diplôme d'ingénieur Géomètre-Topographe



Présenté et soutenu par :

RASAMOELY Aina Natolotra Jonathan

Président : **Monsieur RABETSIAHINY**

Rapporteur : **Monsieur RABEMALAZAMANANA**

Examineurs : **Monsieur NARY Herilalao Iarivo**
Monsieur RAJAONARISON Jean Désiré

Date de Soutenance : 19 Avril 2011

REMERCIEMENTS

C'est avec une grande joie que je consacre cette page en signe de ma profonde gratitude et de reconnaissance à tous ceux qui ont contribué de près ou de loin à la réalisation de ce mémoire.

J'adresse particulièrement mes vifs remerciements :

A Monsieur ANDRIANARY Philippe Antoine,

Directeur de l'Ecole Supérieure Polytechnique d'Antananarivo (ESPA), qui n'a pas ménagé ses efforts pour assurer le bon fonctionnement de l'Etablissement ;

A Monsieur RABETSIAHINY,

Maître de conférences à l'Ecole Supérieure Polytechnique d'Antananarivo,

Chef de Département Information Géographique et Foncière, qui, malgré ses multiples responsabilités, a accepté de diriger ce mémoire.

A Monsieur RABEMALAZAMANANA,

Chef de Département Photogrammétrie et Orthophoto au FTM, qui a dépensé beaucoup de son temps de travail pour encadrer les recherches et les pratiques effectuées.

AUX EXAMINATEURS,

- **Monsieur RAJAONARISON Jean désiré**, Enseignant au sein du Département Information Géographique et Foncière
- **Monsieur NARY Herilalao Iarivo**, Enseignant au sein du Département Information Géographique et Foncière.

Je suis reconnaissant de leur accueil avec lequel ils ont accepté de siéger parmi les membres de jury et d'examiner ce travail malgré leurs multiples activités.

AUX ENSEIGNANTS,

Tous les Enseignants de l'ESPA et en particulier ceux de la filière IGF qui ont transmis leurs savoirs et leurs visions pour la réussite de mon ingénieur.

A MES PARENTS, MES SŒURS, MA FAMILLE ET MES AMIS,

Je n'oublierai pas d'honorer leur dévouement et leurs sacrifices fournis à mon égard durant ces longues années d'études.

TABLE DES MATIERES

REMERCIEMENTS	i
TABLE DES MATIERES	ii
TABLE DES ILLUSTRATIONS	v
LISTE DES ACRONYMES	vi
INTRODUCTION	1
PARTIE I - GENERALITES SUR NOS ETUDES	3
I - OBJET DE L'ETUDE	4
1) A propos de « Easy Topo Version 1.0 » :	4
2) Création de « Easy Topo Version 2.0 » :	4
II - PLACE DE LA TOPOGRAPHIE DANS L'ETUDE ET L'EXECUTION DES PROJETS DE GENIE CIVIL ET RURAL	7
1) Etude d'un projet :	7
2) Exécutions du projet :	7
III - DOCUMENTS TOPOGRAPHIQUES UTILES A L'ETUDE D'UN PROJET	8
1) Canevas de points :	8
2) Plan-coté :	8
3) Modèle Numérique de terrain :	8
4) Courbes de niveau :	9
5) Profil en long et profil en travers :	9
PARTIE II - LES PROCEDES DE CALCULS TOPOGRAPHIQUES	10
I - LES TRAITEMENTS TOPOGRAPHIQUES APPLIQUES AU LOGICIEL	11
1) Calculs de compensation des coordonnées planimétriques :	11
2) CALCUL et DESSIN de MNT :	11
a) Précision du MNT :	12
b) Solution pour la précision de l'exploitation du MNT :	12
3) Détermination des courbes de niveau :	13
4) Edition de profil en long et en profil en travers :	13
5) Différents rendus du relief en trois dimensions :	13
II - LES METHODES ET FORMULES DE CALCULS NECESSAIRES A L'EXECUTION DE CES TRAITEMENTS	14

1) Les procédés et méthodes des calculs de compensation par moindres carrés :	14
2) Les différentes méthodes de résolution du système d'équation après normalisation :	16
3) Le principe d'interpolation des courbes de niveau :	17
4) Les formules mathématiques de Lissage des courbes de niveaux :	18
a) Formule de base :	18
b) Reformulation mathématique	19
b.1. Equations de la courbe de Bézier :	21
b.2. Equations de la courbe B-Spline :	22
b.3. Equations des courbes C-Spline et T-Spline :	23
5) La production de MNT :	25
a) Les principales méthodes :	25
b) L'algorithme de DELAUNAY :	27
b.1. La triangulation par Destruction-construction :	27
b.2. La triangulation par Vérification : Côtés légaux- illégaux :	30
6) Les techniques du rendu virtuel en 3D :	32
PARTIE III - IMPLEMENTATION DES ALGORITHMES DE CALCULS TOPOGRAPHIQUES	33
I - REALISATION DU PROGICIEL DANS L'ENVIRONNEMENT DE DEVELOPPEMENT DU LANGAGE VISUAL BASIC	34
1) Choix de Visual Basic :	34
2) Les bases du langage Visual Basic :	34
3) Concepts de base de la programmation orientés objet :	34
a) Objet et classe :	35
b) Références d'objets pour notre programme :	35
b.1. OLE Automation :	35
b.1. Directx7 for Visual Basic Type Library:	35
b.2. Microsoft office 12.0 Object library :	35
b.3. AutoCAD 2008 Type Library :	37
II - LES DIFFERENTES ETAPES DU DEVELOPPEMENT	39
1) Modélisation de l'application	39
a) Diagramme des classes	39
2) Création de l'application :	41
a) Phase de codage : (Organigramme et Algorithme des calculs)	41
a.1. L'organigramme des calculs :	41
a.2. L'Algorithme des calculs :	43

b) Création de la source de données et gestion des fichiers.....	45
c) Création de l'interface : fenêtres, menu, commandes, tables... ..	47
d) Compilation de l'application.....	47
3) Phase test	48
III - Evaluation de la qualité et la précision de « proJET Easy-Topo » :.....	50
1) Calcul de MNT de 3917 points cotés de l'île Sakatia au Nord de Madagascar:	51
2) Edition de profils topographiques par le MNT :	53
3) Création de courbes de niveaux :	55
4) Calcul et rendu de maillage 3D :	56
PARTIE IV - INTEGRATION DU LOGICIEL DANS LE SYSTEME D'EXPLOITATION « WINDOWS »	57
I - PRESENTATION DU LOGICIEL :	58
II - INSTALLATION ET DESINSTALLATION :	59
1) Installation :	59
2) Désinstallation :	60
III - MODE D'EMPLOI.....	61
1) APERÇUES DU LOGICIEL :	62
a) Icône et identité :	62
b) Fenêtre d'accueil :	62
c) Fenêtre d'ouverture ou de sauvegarde de fichiers	62
d) Fenêtre principale :	63
e) Fenêtres filles :	63
f) Fenêtre de visualisation des plans topographiques :	64
CONCLUSION	65
REFERENCES.....	66
ANNEXES.....	67

TABLE DES ILLUSTRATIONS

Figure 1 : Etendue du logiciel « Easy Topo »	6
Figure 2 : Interpolation de courbes de niveau	17
Figure 3 : Courbe de Bézier	21
Figure 4 : Courbe B-Spline	22
Figure 5 : Courbes C-Spline, T-Spline	23
Figure 6 : Modèle numérique de terrain, Rectangulaire.....	25
Figure 7 : Modèle numérique de terrain, Triangulaire	25
Figure 8 : Vue en perspective du MNT et du MNE.....	26
Figure 9 : Triangulation par Destruction-construction.....	27
Figure 10 : [BC] un côté légal	30
Figure 11 : Transformation d'un côté illégal en côté légal.....	30
Figure 12 : Modèle d'objets Microsoft Excel.....	36
Figure 13 : Modèle d'objet Application AutoCAD	38
Figure 14 : Structure d'une classe sous forme UML	39
Figure 15 : Format d'Impression A0 des profils en travers	54
Figure 16 : Habillage du profil en travers	54
 Tableau 1 : Résultats sur la capacité de traitement du logiciel	 49
Tableau 2 : Mode d'emploi du Logiciel	61
 Diagramme 1 : Diagramme des classes du programme	 40
Diagramme 2 : Calcul de distance planimétrique entre 2 points topographiques.....	42

LISTE DES ACRONYMES

2D : Deux Dimensions.

3D : Trois Dimensions.

VB : Visual Basic

VBA : Visual Basic for Application

COM : Component Object Model

MNT : Modèle Numérique de Terrain

DAO : Dessin Assisté par Ordinateur

TIN : Triangulated Irregular Network

MNE : Modèle Numérique d'élévation

DEM : Digital Elevation Model.

NURBS : Non Uniformes Rationnelles B-Splines

MDI : Multiple Document Interface

UML : Unified Modeling Language

INTRODUCTION

En accord avec l'évolution actuelle de la technologie de l'informatique, le Département Information Géographique et Foncière de l'ESPA s'est organisé d'une salle d'informatiques avec des matériels et des logiciels professionnels du domaine de la Topographie et de la Cartographie.

Cette étude a débuté par proposition du Département et lors des séances de travaux pratiques d'Informatiques-Topo. Nous avons demandé aux Enseignants de la matière, une suggestion pour la mémoire de fin d'étude, et ils nous ont proposé d'orienter l'étude dans l'informatique-Programmation pour les raisons suivantes :

Premièrement, l'informatisation devient actuellement la tendance dans le domaine de la Topographie. Or les utilisateurs des logiciels topographiques existants détectent que ces derniers sont pour l'usage global et ne permettent pas de personnalisation en accord avec les traitements des données en main. Ces logiciels sont incomplets et pas spécifiquement orientés face aux besoins particuliers de chaque utilisateur. Il serait donc intéressant de compléter et d'organiser ces processus de traitements en faisant intégrer de plus à ces derniers une application programmée qui répond aux tâches personnelles de l'utilisateur.

La deuxième raison est la lenteur de réalisation des documents topographiques après les travaux sur terrain. Nous comprenons les difficultés que les opérateurs doivent affronter : en effet, pour la réalisation de chaque document topographique, il faudrait répéter les mêmes calculs topométriques pour chaque point du terrain, et aussi d'engager plusieurs dessinateurs pour les multitudes de reports graphiques. Pour les travaux de grande envergure, ces traitements risquent de prendre plusieurs semaines. On ne tient pas encore compte des opérations qu'on ne peut pas faire à la main ou mentalement, comme la résolution d'une matrice à n inconnues avec m équations dans une compensation par moindres carrés.

La troisième raison est que le Département a l'ambition de développer son propre logiciel de traitement de l'Information Géographique.

La seule solution rentable est de recourir aux programmations informatiques. C'est dans ce cadre que s'inscrit le thème de ce mémoire : « **Elaboration d'un logiciel de calcul et de dessin topographique – proJET EasyTopo Version 2** ». Il se divise en quatre parties :

Nous nous portons en premier lieu à présenter nos objectifs et à spécifier les modules de la topographie à programmer.

En second lieu, nous entrons dans les détails des procédés, des méthodes et des formules de calculs nécessaires pour élaborer les différents algorithmes de notre programme.

Ensuite nous allons implémenter ces algorithmes en suivant les étapes dans un développement de logiciel. Ces étapes se terminent par des pratiques, des exemples et des comparaisons avec des logiciels professionnels déjà existants.

Enfin nous finalisons la programmation par l'intégration de « proJET Easy Topo » dans le système « WINDOWS » pour son utilisation lors des travaux pratiques du Département.

PARTIE I - GENERALITES SUR NOS ETUDES

I - OBJET DE L'ETUDE

1) A propos de « Easy Topo Version 1.0 » :

« Easy Topo Version 1.0 » a été l'objet du travail de fin d'étude de Monsieur MAMILALA Tolotra Nantenaina, Etudiant dans la filière Information Géographique et Foncière (IGF), Promotion 2009 ; et a été encadré par Monsieur RABEMALAZAMANANA, Enseignant à l'Ecole Supérieure Polytechnique d'Antananarivo (ESPA).

« Easy Topo Version 1.0 » a pour fonction l'automatisation des calculs topographiques tels que l'intersection, relèvement, cheminement planimétrique et altimétrique, calcul de surface. Il possède la capacité d'importer des données terrain Excel et il peut aussi faire des dessins automatiques sur AUTOCAD à partir des points calculés. [12]

« Easy Topo Version 1.0 » s'est concentré sur ces traitements en deux dimensions. Alors vient notre nouvelle étude, qui est d'actualité, le développement de la seconde version spécialisée dans les traitements topographiques en trois dimensions avec sa propre fenêtre de dessins et de visualisations, en plus de celle d'AUTOCAD.

2) Création de « Easy Topo Version 2.0 » :

Cette étude intéresse principalement le Département IGF qui espère ne plus utiliser d'autres logiciels dans son domaine, en élaborant un logiciel topographique répondant à ses besoins : ce sont les différentes activités entreprises, les matières étudiées en classe, et les traitements des travaux pratiques de terrain... Elle intéressera aussi tous les bureaux d'études et les entreprises qui utilisent la topographie avec leurs propres personnalisations.

Pour notre cas nous essayerons avec notre logiciel de ne plus utiliser COVADIS. Cette seconde version d'Easy Topo nommée «proJET Easy Topo » est orientée pour le Génie Civil et Rural grâce aux traitements en trois dimensions. Le calcul de compensation de points par la méthode des moindres carrés est un complément d'Easy Topo 1.0 qui traite la topométrie en deux dimensions.

L'étude se base donc sur la création du logiciel pour des résultats rapides, avec précision, et aux normes dans les traitements topographique en **2D** et en **3D** suivants :

- ❖ Compensations des points (X, Y) par méthode des moindres carrés.
- ❖ Calculs et dessin de MNT
- ❖ Calculs, Lissage et dessin de **courbes de niveaux** à partir de MNT
- ❖ Calculs et dessin de **profils topographiques** à partir de MNT
- ❖ Modélisation 3D (surfaces 3D, rendu 3D)
- ❖ Calculs et dessin de profils en long et profils en travers par des données levées sur terrain.

Le croquis ci-après nous montre l'étendue du logiciel « Easy Topo ». On y trouve les parties traitées dans « EasyTopo Version 1.0 », celles traitées dans ce mémoire « Version 2.0 », et celles à aborder dans les versions futures :

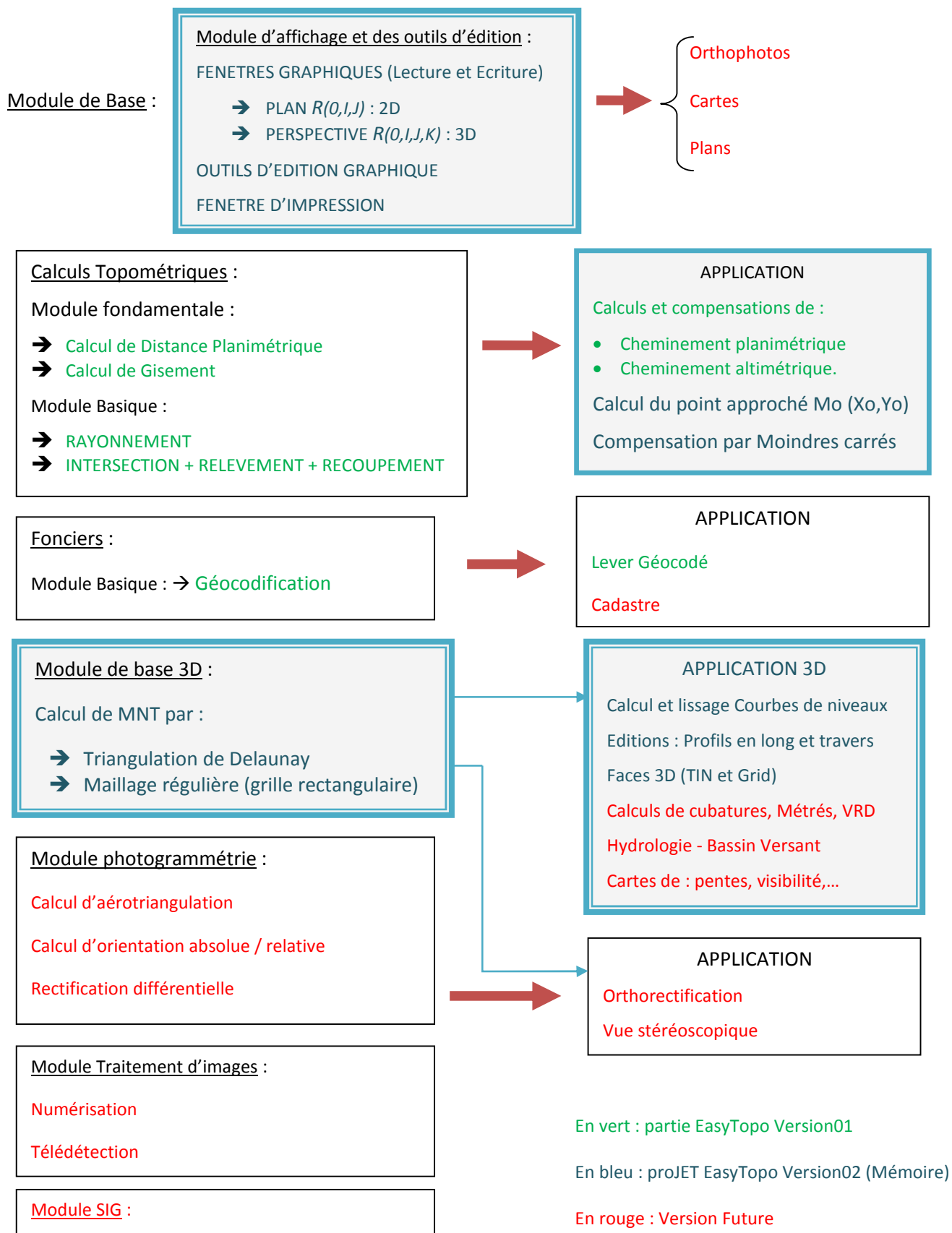


Figure 1 : Etendue du logiciel « Easy Topo »

II - PLACE DE LA TOPOGRAPHIE DANS L'ETUDE ET L'EXECUTION DES PROJETS DE GENIE CIVIL ET RURAL [10]

1) Etude d'un projet :

Dans l'étude d'un projet, la finalité de la topographie est la représentation d'une surface de terrain dans un document. Il doit comporter les détails planimétriques et altimétriques nécessaires à la réalisation du projet. Dans chaque phase de l'étude les documents nécessaires sont en générales comme suit :

Dans la phase de reconnaissance :

Le terrain est représenté à l'aide de cartes ou de photos aériennes ou de plans topographiques à petites échelles.

A ce stade, on cherche à déterminer les différents emplacements possibles du projet, en fonction de la technique adoptée et/ou des raisons d'ordre économique concernant le projet.

Dans l'avant projet sommaire :

Le terrain est représenté sommairement après adoption d'une solution définitive au projet. Il s'agit ici de la détermination de l'exécution, au moindre coût, du projet, par exemple la recherche du tracé « idéal » dans un projet routier.

Les détails planimétriques et altimétriques de cette phase sont souvent tirés à partir de carte ou plans topographiques à petites échelles.

Dans l'avant projet détaillé :

A ce stade on détermine en détail le coût et la technique du projet. Les documents topographiques ont donc besoin de plus de précision. Pour cela on effectue de levées à grandes échelles des représentations altimétriques et planimétriques précises selon le type du projet.

2) Exécutions du projet :

A la réalisation ou exécution du projet, la topographie entre comme pilote dans l'implantation de celui-ci. Elle assure la conformité de l'emplacement des détails du projet au plan préétabli.

La partie traitée dans ce mémoire est seulement la partie de la topographie utilisée à l'étude du projet par les cubatures et profils. Ceci concerne le « trois dimensions », mais nous développerons aussi la compensation par moindres carrés pour le contrôle des coordonnées planimétriques des ouvrages.

III - DOCUMENTS TOPOGRAPHIQUES UTILES A L'ETUDE D'UN PROJET

[10]

1) Canevas de points :

Il s'agit de points d'appuis, de rattachement, ou de calage si on traite des images. Ces points sont indispensables pour la fermeture et compensation des points levés. Le projet doit être géo-référencié c'est-à-dire situé dans le système de coordonnées du pays selon le type de projection qu'il utilise : « Laborde Madagascar », pour Madagascar.

2) Plan-coté :

Ce plan est utilisé pour l'installation des infrastructures (ponts, barrages, bâtiments, routes, etc....)

C'est un dessin topographique à une échelle donnée, dans lequel on indique les détails planimétriques du terrain (rivières, voie de communication, édifice,..) par des signes conventionnelles. Les cotes ou points de hauteur par rapport à un plan horizontal de comparaison, donnent les informations altimétriques du terrain (crêtes, talus, ondulations, dépressions...) le plan de comparaison peut être le niveau moyen de la mer, ou un plan de comparaison local. On ne cote bien sûr que les points essentiels caractéristiques du terrain.

3) Modèle Numérique de terrain :

Un **modèle numérique de terrain (MNT)** est une représentation de la topographie (altimétrie et/ou Bathymétrie) d'une zone terrestre (ou d'une planète tellurique) sous une forme adaptée à son utilisation par un calculateur numérique (ordinateur). [9]

En cartographie, les altitudes sont habituellement représentées par des courbes de niveaux et des points cotés. Suivant la taille de la zone couverte, la plupart des MNT utilisent pour les petites zones, un maillage régulier carré ou pour les grandes zones, un maillage pseudo carré dont les côtés sont des méridiens et des parallèles.

Le MNT permet ainsi : [7]

- de reconstituer une vue en images de synthèse du terrain ;
- de déterminer une trajectoire de survol du terrain ;
- de calculer des surfaces ou des volumes ;
- de tracer des profils topographiques : Profil en long / Profil en travers;
- d'étudier un bassin versant ;
- de produire les courbes de niveaux ;
- de produire des cartes d'ensoleillement, d'aménagement, carte des pentes, carte des reliefs, mise en place des réseaux d'antennes de Télécommunication.
- d'une manière générale, de manipuler de façon quantitative le terrain étudié.

Il sert aussi à caler des images orthophotos.

4) Courbes de niveau :

Les courbes de niveau, appelées isohypses, sont destinées à donner sur une carte et/ou un plan un aperçu du relief réel. Une courbe de niveau est l'intersection du relief réel avec un plan horizontal d'altitude donnée en cote ronde (généralement un nombre entier).

Pour rendre visible sur le plan les inégalités du sol, on rejoint parfois une suite de points de même altitude en une courbe qui sont les courbes de niveau. Souvent les points de même altitude en cote ronde ne se présentent pas directement dans le plan, donc on procède à des interpolations sur quelques semis de points voisins entre eux.

5) Profil en long et profil en travers :

Ces documents sont utilisés dans les tracées de routes, pistes, canaux et dans l'implantation des réseaux de télécommunication.

Le profil en long représente la coupe du terrain suivant un axe vertical. Dans un dessin de profil en long, on doit représenter le terrain naturel et le projet.

Les profils en travers représentent la coupe verticale du terrain suivant un plan perpendiculaire à l'axe du projet. A chaque point du profil en long correspond un profil en travers. Et on choisit tous ces points à chaque changement de pente et à chaque changement de direction.

Lors d'un avant-projet sommaire de l'étude d'un projet routier, ces deux types de graphiques permettent d'obtenir, après plusieurs études, un tracé « idéal » répondant aux impératifs du projet qui sont la visibilité, l'emprise sur le terrain, la déclivité maximale, le moindre coût, etc.

PARTIE II - LES PROCEDES DE CALCULS TOPOGRAPHIQUES

I - LES TRAITEMENTS TOPOGRAPHIQUES APPLIQUES AU LOGICIEL

1) Calculs de compensation des coordonnées planimétriques :

Notre logiciel permet de calculer :

- des points de triangulation,
- des points d'un réseau géodésique (position des sommets ; longueur et azimuth des côtés),
- des points d'une microtriangulation (auscultation de barrage...).

On mesure des angles et des distances ainsi que des orientations en nombre plus grand, c'est-à-dire des mesures surabondantes qui sont nécessaires à la bonne précision de résolution des triangles.

Les différentes formes de compensation sont : **[6]**

- Forme empirique : le plus simple et sans calculatrice.
- Compensation mécanique ou analogique
- Forme graphique : report sur minute à grande facteur d'échelle (exemple 10 fois) des directions des visées, et sensibilités au voisinage du point à déterminer.
- Forme numérique : on emploie ici la méthode des moindres carrés.

Nous avons choisi pour notre programme la compensation par méthode des moindres carrés.

2) CALCUL et DESSIN de MNT :

On peut distinguer les MNT selon le type de maillage utilisé :

- maillage carré/rectangulaire ;
- maillage hexagonal ;
- maillage triangulaire régulier ;
- maillage triangulaire quelconque.

En fonction du type de maillage, la représentation informatique du MNT varie. Dans le cas de maillages rectangulaires, on peut utiliser des tableaux, mais dans les autres cas, les structures de données sont plus complexes.

Pour notre application on a choisit la triangulation de Delaunay car elle permet de récupérer une bonne triangulation à partir de semis de points.

On n'aura pas à interpoler les points comme dans les maillages rectangulaires. L'interpolation diminue la précision et donc perd la vraie forme du relief. Mais nous avons programmé aussi les maillages rectangulaires car ils constituent un premier lissage de la surface à visualiser.

a) Précision du MNT :

Si les coordonnées du semi de points sont obtenues à 10^{-3} près (millimétriques), l'algorithme de Delaunay ne perd pas cette précision (il n'y a ni interpolation, ni arrondissement).

Par contre, le dessin sur plan, sur carte, ou à l'écran, du MNT présente une erreur graphique (E_g) qui varie **la précision linéaire** (en X et en Y) en fonction de l'échelle **E**. Par exemple, pour :

$E = 1/1000^{\text{ème}}$ et $E_g = 1/10^{\text{ème}}$ mm, la précision du MNT est 10cm ;

$E = 1/10.000^{\text{ème}}$ et $E_g = 1/10^{\text{ème}}$ mm, la précision devient 1m ;

$E = 1/100.000^{\text{ème}}$ et $E_g = 1/10^{\text{ème}}$ mm, la précision = 10m.

La précision altimétrique source reste inchangée du fait qu'on inscrit sur le dessin, les cotes respectives des points ou des courbes de niveaux.

b) Solution pour la précision de l'exploitation du MNT :

Notre logiciel va créer, avec les dessins, des fichiers qui contiennent une liste numérique des triangles avec la précision du semis de points (source). Les points sur l'écran auront comme attributs leurs coordonnées respectives dans les fichiers numériques, au lieu de leurs coordonnées mesurées à l'écran.

3) Détermination des courbes de niveau :

Par interpolation des points cotés et du MNT nous aurons des lignes vecteurs de même altitude (polylignes). Pour améliorer la représentation de courbe, on y applique la méthode de lissage qui consiste à générer et représenter des formes douces et ergonomiques.

Le logiciel effectuera toutes les opérations. Nous pouvons choisir entre les produits de lissages voulus, à citer :

- Les courbes de « Bézier »
- Les courbes « B-Splines » non rationnelles uniformes (NRU)
- Les courbes « B-Splines non uniformes non rationnelles (NUNR)
- Les courbes B-Splines non uniformes rationnelles: les NURBS
- Les courbes Splines de « Catmull-Rom » : (C-Spline) et (T-Spline)

4) Edition de profil en long et en profil en travers :

Notre logiciel génère et dessine automatiquement tous les éléments des tracés quelque soit le nombre des points des profils en long et en travers, par exemple 5000 profils en travers. Il va créer des fichiers correspondants au projet. Dans un projet routier le rôle de l'informatique est alors déterminant grâce au gain de temps sur les tracés des profils. Il est possible d'envisager une programmation qui aboutit automatiquement, en plusieurs passes de calcul, au tracé « idéal » en fonction de paramètres fixés par le projeteur.

5) Différents rendus du relief en trois dimensions :

Parfois les plans ont un degré élémentaire de lecture, c'est-à-dire qu'il faut un travailleur expérimenté pour décoder les différentes signes conventionnelles qui y sont inscrites. On arrive parfois jusqu'à utiliser une loupe si le plan ou la carte est surchargé et se présente comme une base de données.

Les rendus 3D virtuels du chantier nous aident à interpréter les codes et les signes conventionnelles des plans très techniques. Ils aident aussi les décideurs non expérimentés (Bailleurs de fonds, Ministères,...), à voir clairement l'importance ou le déroulement des projets.

Les rendus 3D peuvent remplacer les maquettes(*) traditionnelles et présentent alors un avantage au niveau de l'économie d'argent et de temps.

**Maquettes faites en bois à l'aide de colle et contreplaqués.*

II - LES METHODES ET FORMULES DE CALCULS NECESSAIRES A L'EXECUTION DE CES TRAITEMENTS

1) Les procédés et méthodes des calculs de compensation par moindres carrés :

[3]

En raison de l'imperfection des coordonnées des points d'appuis et des erreurs d'observation, les lieux déterminatifs de chacun des points nouveaux ne sont pas concourants. Les points définitifs sont fixés par rapport à ces lieux de telle façon que le réseau définitif soit aussi voisin que possible du réseau observé. Les écarts constatés entre ces réseaux ne doivent pas être supérieures aux erreurs susceptibles d'entacher des mesures. Pour palier ces imperfections, on a développé la méthode des moindres carrés.

Principe de la méthode :

C'est une méthode générale dont l'application présente plusieurs variables. Elle suppose que les fautes et les erreurs systématiques ont été enlevées préalablement soit par mode opératoire soit par des corrections appropriées. Seule restent les écarts ou résidu v entre éléments définitifs compensés et éléments mesurés. La méthode consiste à rendre minimum la somme des carrés des résidus v . Elle est la solution la plus probable au sens du calcul des probabilités.

Pose des relations d'observations :

Les relations suivent la formule

$$\text{VALEUR COMPENSEE} - \text{VALEURE MESUREE} = \text{RESIDU} \quad (2.1)$$

$$x - x_i = v \quad \sum v^2 \text{ minimum}$$

En stationnant sur A et en visant un point B on aura l'équation brute avec les Unités Système Internationale : les distances en mètre, les angles en radian.

$$-\frac{\Delta Y}{D^2} dx_A + \frac{\Delta X}{D^2} dy_A + \frac{\Delta Y}{D^2} dx_B - \frac{\Delta X}{D^2} dy_B + V_{calc} - V_{obs} - dV_0 = v \quad (2.2)$$

Equation en « i » :

Chaque relation d'observation comporte une inconnue dVo caractéristique d'un tour d'horizon. La méthode d'élimination de ces dVo consiste à ajouter pour chaque station une relation auxiliaire dont les coefficients et le terme constant sont respectivement égaux à la somme des coefficients et des termes constants des relations correspondantes. Cette relation auxiliaire est ensuite affectée d'un poids

$-1/n$ ce qui revient à multiplier ces termes constants par i/\sqrt{n} . Nous aurons les équations avec partie imaginaire appelées équations en « i ».

$$n = \text{nombre_points_visés} + \text{nombre_points pour la détermination du Vo station.}$$

(2.3)

Nous pourrions alors écrire les relations d'observations sans tenir compte des inconnues dVo.

Adaptation de l'équation selon notre besoin :

Nous savons que : **angle en radian = angle en seconde décimale * Sin (1'')**

Et une seconde décimale : **1'' = 0,0001 grade**, ou 1 déci-milligrade

On veut les angles en seconde, donc on aura une équation finale :

$$\left\{ -\frac{\Delta Y}{D^2} dx_A + \frac{\Delta X}{D^2} dy_A + \frac{\Delta Y}{D^2} dx_B - \frac{\Delta X}{D^2} dy_B \right\} / \sin 0.0001^G + \Delta'' = v''$$

(2.4)

$$\text{avec } \Delta'' = \Delta^{\text{Grade}} \cdot 10^{-4}$$

$$\sin 0.0001^G \approx 0,00000157$$

En faisant un changement de variables on aura l'équation prête à être programmée :

$$-\frac{\Delta Y}{D^2} \partial x_A + \frac{\Delta X}{D^2} \partial y_A + \frac{\Delta Y}{D^2} \partial x_B - \frac{\Delta X}{D^2} \partial y_B + \Delta'' = v''$$

(2.5)

$$\text{avec } \partial x_i = dx_i / \sin 0.0001^G \quad \Delta Y(m), \quad D(m), \quad \partial x(m)$$

Les coordonnées compensées s'obtiennent finalement par les formules :

$$\begin{cases} X_i (\text{compensée}) = X_{O_i} (\text{approchée}) + \partial x_i \end{cases} \quad (2.5.1)$$

$$\begin{cases} Y_i (\text{compensée}) = Y_{O_i} (\text{approchée}) + \partial y_i \end{cases} \quad (2.5.2)$$

Établissement du système d'équation :

On constitue le système d'équation dont le nombre d'équations est supérieur au nombre d'inconnus (**m>n**). Plus on a plusieurs équations, plus la détermination des points inconnus sera d'autant plus précise. On fait donc plusieurs visées en Intersection et en Relèvement.

2) Les différentes méthodes de résolution du système d'équation après normalisation (m=n) :

Tout en sachant que la somme des carrés des résidus est minimale $S = \sum v_i^2 = 0$, et que v_i sont fonctions des inconnues $\partial x_1, \partial x_2, \dots, \partial x_n$, les dérivées partielles de S par rapport à ces inconnues devront être nulles.

$$\boxed{\frac{\delta S}{\delta \partial x_1} = \frac{\delta S}{\delta \partial x_2} = \dots = \frac{\delta S}{\delta \partial x_n} = 0} \quad (2.6)$$

Chaque dérivée partielle, égale à zéro, constituera le système d'équation $m = n$, et plusieurs méthodes peuvent résoudre ce système en analyse numérique.

→ La première méthode est la résolution par élimination :

On peut citer les algorithmes de GAUSS, CHOLSKY, GAUSS-DOOLITTLE, BANACHEWITZ.

→ La deuxième méthode est la résolution itérative :

On peut citer les algorithmes de GRADIENTS CONJUGUES et de JACOBI.

On a adopté la méthode de GAUSS pour cette application.

Cette méthode, qui est aussi une méthode par pivot, se focalise à rendre la matrice symétrique (par rapport à la diagonale) et définie positive. Le déterminant est différent de zéro.

Soit la matrice (M) : $m = n$

$$\begin{bmatrix} M \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

Sous une autre forme elle se présente comme suit :

$$\begin{aligned} \rightarrow (A)(X) &= (K) \quad \text{avec } (A) : \text{coefficients, } (X) : \text{inconnus, } (K) : \text{termes constantes} \\ \underbrace{{}^T(A).(A)}_{(B)}.(X) &= \underbrace{{}^T(A).(K)}_{(Y)} \end{aligned}$$

(B) est une matrice carré ($i=j$).

3) Le principe d'interpolation des courbes de niveau : [10]

La connaissance du principe d'interpolation nous permet de choisir judicieusement le nombre et la position des points à lever.

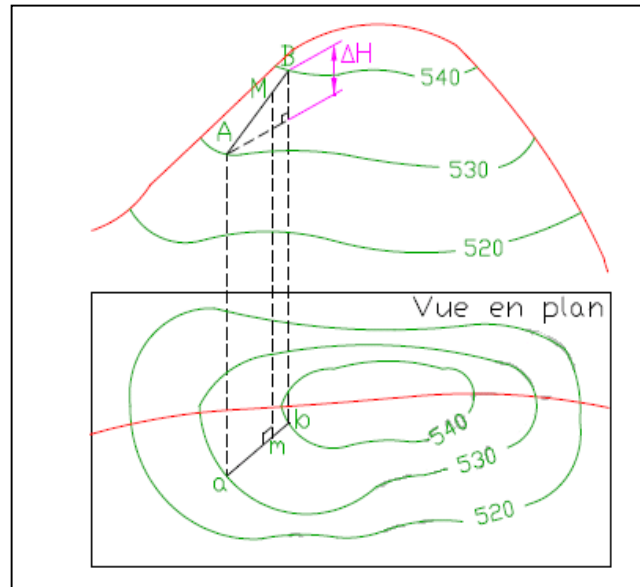


Figure 2 : Interpolation de courbes de niveau

L'altitude au point M, situé entre les courbes de niveau 530 et 540, est déterminée en considérant le terrain en pente constante entre A et B. Les points A et B sont les points les plus proches de M sur les courbes de niveau 530 et 540 ; ici $\Delta H = 10$ m.

La pente au point M vaut :

$$p = \frac{\Delta H}{ab} \quad (2.7)$$

La distance **ab** est la distance réelle, c'est-à-dire la distance mesurée sur le plan et divisée par l'échelle du plan.

L'altitude de M est :

$$H_M = H_A + am \cdot \frac{\Delta H}{ab} \quad (2.8)$$

Mais pour notre cas, nous effectuons un calcul inverse car on veut dessiner les courbes de niveau H_M sur le plan. H_M est donc connu, et il reste à déterminer les coordonnées X_M et Y_M . On aura la distance **am** par déduction de la formule **(2.8)** :

$$am = (H_M - H_A) \cdot ab / \Delta H \quad (2.9)$$

Par rayonnement à partir du point A on a :

$$\begin{cases} X_M = X_A + am \cdot \sin(gAB \cdot \pi / 200) \\ Y_M = Y_A + am \cdot \cos(gAB \cdot \pi / 200) \end{cases} \quad , \text{ gAB est le gisement de AB}$$

$$Z_M = H_M \quad (2.10)$$

4) Les formules mathématiques de Lissage des courbes de niveaux :

[5], [11]

a) Formule de base :

La représentation de courbes après lissage que ce soit du type **Bézier**, **B-Spline**, **C-Spline**, ou **T-Spline** est générée par des équations paramétriques cubiques de \mathbb{R}^3 .

Le système d'équations de base est donné ci-dessous :

$$Q(t) = \begin{cases} x(t) = a_x t^3 + b_x t^2 + c_x t + d_x \\ y(t) = a_y t^3 + b_y t^2 + c_y t + d_y \\ z(t) = a_z t^3 + b_z t^2 + c_z t + d_z \end{cases} \quad , \text{ avec } t \in \mathbb{R} \quad (2.11)$$

Pour obtenir des segments de courbes, on considère que **t** appartient à un intervalle [min,max] (fréquemment [0.0,1.0]).

b) Reformulation mathématique

Soit le vecteur $T = (t^3, t^2, t, 1)$, **(2.11)** s'écrit sous la forme

$$Q(t) = (x(t), y(t), z(t)) = T.C \text{ avec } C = \begin{pmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{pmatrix}, \text{ et } t \in \mathbb{R}. \quad (2.12)$$

Les courbes de cette famille ont pour propriétés d'être:

- continues,
- de dérivés premières en t continues,
- de dérivés secondes en t continues.

Quatre coefficients numériques constants sont nécessaires pour définir chacune des trois équations paramétriques.

→ On impose quatre contraintes mathématiques pour définir l'équation selon chacun des trois axes (OI, OJ, OK).

On transforme la matrice C donnée précédemment en une matrice $M.G$ où M est une matrice 4×4 (appelée matrice de base), et G est un vecteur colonne de quatre contraintes géométriques (appelé vecteur géométrie).

$$Q(t) = (x(t), y(t), z(t)) = T.C = T.M.G = \begin{pmatrix} t^3 & t^2 & t & 1 \end{pmatrix} \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{pmatrix} \begin{pmatrix} g_{1x} & g_{1y} & g_{1z} \\ g_{2x} & g_{2y} & g_{2z} \\ g_{3x} & g_{3y} & g_{3z} \\ g_{4x} & g_{4y} & g_{4z} \end{pmatrix} \quad (2.13)$$

La matrice G permet de représenter quatre contraintes géométriques s'appliquant à la courbe. Il s'agit habituellement de la position de 4 points différents dans l'espace de représentation. La courbe générée évoluera "entre" ces quatre points.

Soit une ligne polygonale de $n+1$ sommets P_i , $0 \leq i \leq n$ avec $n \geq 3$.

On définit $n-2$ vecteurs géométriques $\mathbf{G}_i = \begin{pmatrix} P_i \\ P_{i+1} \\ P_{i+2} \\ P_{i+3} \end{pmatrix}$, $0 \leq i \leq n-3$.

Ces $n-2$ vecteurs géométriques permettent la génération de $n-2$ morceaux de courbe jointifs qui une fois réunis formeront la courbe approximant la ligne polygonale.

La matrice \mathbf{M} permet d'attribuer des "poids" respectifs à chacun des coefficients \mathbf{t}^i s'appliquant à une contrainte géométrique, permettant ainsi de définir la "forme" de la courbe.

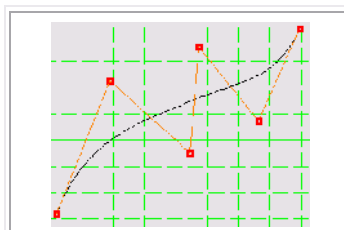
De bons choix sur la matrice \mathbf{M} permettront d'obtenir des tracés de courbes approximant la ligne polygonale à 4 sommets définis par les quatre points. On aura les courbes de Bezier, de Catmull-Rom, et les « **NURBS** ».

b.1. Equations de la courbe de Bézier :

Si $P_0, P_1, P_2, P_3, \dots, P_n$ sont les $n+1$ points d'une ligne polygonale. La courbe de Bézier définie par ces points est :

$$Q(t) = (x(t), y(t), z(t)) = T.C = T.M.G = \begin{pmatrix} t^3 & t^2 & t & 1 \end{pmatrix} \underbrace{\begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}}_M \begin{pmatrix} P_{0x} & P_{0y} & P_{0z} \\ P_{1x} & P_{1y} & P_{1z} \\ P_{2x} & P_{2y} & P_{2z} \\ P_{3x} & P_{3y} & P_{3z} \end{pmatrix} \quad (2.14)$$

Le coefficient t est compris entre 0 et 1.

Précision :

Non Précise : il perd la forme du relief à cause de l'exagération de lissage.

Application : Tracé de courbes lissées quand le nombre de points de contrôle n'est pas imposé.

Avantage : Contrôle local, forme "trop" lissée mais contrôlée ; ajout possible de points de contrôle.

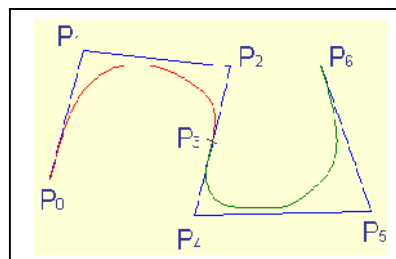


Figure 3 : Courbe de Bézier

b.2. Equations de la courbe B-Spline :

Les $n-2$ vecteurs géométriques permettent la génération de $n-2$ morceaux de courbe jointifs qui une fois réunis formeront la courbe approximant la ligne polygonale.

Pour chaque morceau la valeur de t variera obligatoirement et uniformément dans l'intervalle $[0.0 ; 1.0]$.

La matrice de base **M** utilisée est :

$$\frac{1}{6} \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix}$$

Les morceaux de courbe se joignent obligatoirement en leurs extrémités. En effet, soient les deux morceaux de courbe correspondant à G_i et G_{i+1} , le premier se finit à $t = 1$, le second commence à $t = 0$.

En ces points, on calcule que les coordonnées obtenues sont identiques et ont pour valeur :

$$\frac{P_{i+1} + 4P_{i+2} + P_{i+3}}{6}$$

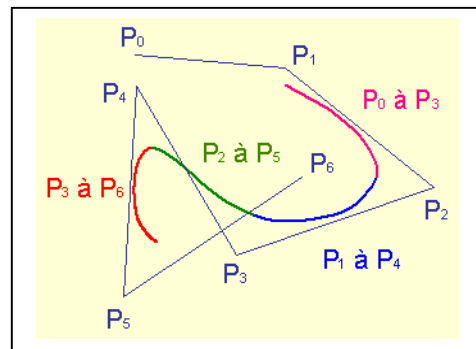
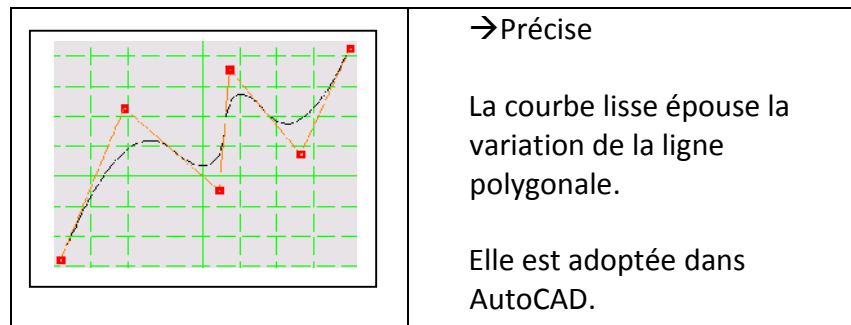


Figure 4 : Courbe B-Spline

Les courbes B-Splines non uniformes rationnelles, les NURBS, ont la particularité suivante

$$Q(t) = \left(\frac{x(t)}{w(t)} \quad \frac{y(t)}{w(t)} \quad \frac{z(t)}{w(t)} \right) \quad (2.15)$$

avec $x(t)$, $y(t)$, $z(t)$ et $w(t)$ des équations paramétriques cubiques représentant des coordonnées homogènes.

Précision :Avantage :

Permet une modification des paramètres de visualisation en perspective sans recalculer la B-Spline.

b.3. Equations des courbes C-Spline et T-Spline :

Ces courbes ont la propriété de passer par chacun des sommets de la ligne polygonale.

La matrice **M** de base est :

$$\frac{1}{2} \begin{pmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{pmatrix}$$

Pour le vecteur géométrique **Gi** la courbe commence pour $t = 0$ en P_{i+1} , et finit pour $t = 1$ en P_{i+2}

→ Tous les morceaux se rejoignent en leurs extrémités pour des intervalles $[0,1]$. Ces extrémités sont les points de contrôle.

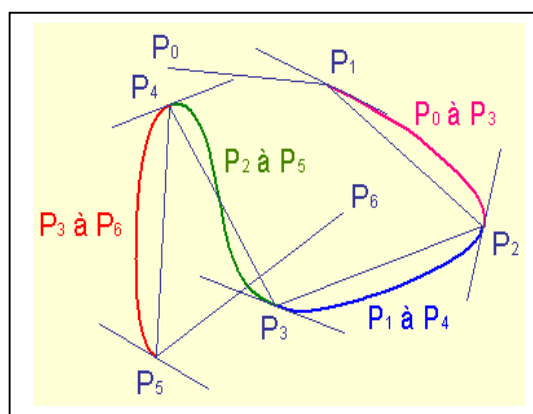
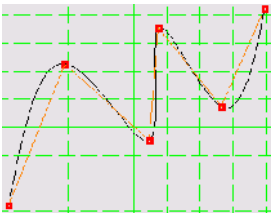
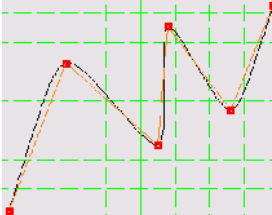
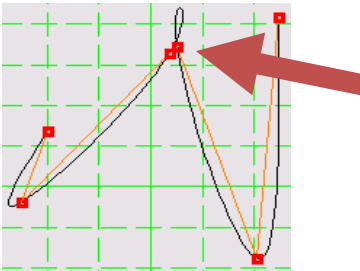


Figure 5 : Courbes C-Spline, T-Spline

Précisions :

	<p><u>C-Spline</u> : Non précise</p> <p>Elle n'est pas proportionnelle à la variation de la ligne polygonale originale.</p>
	<p><u>T-Spline</u> : Non précise</p> <p>Elle est très proche de la ligne polygonale donc le lissage ne se voit pas.</p>
	<p>Pour C-Spline et T-Spline, la courbe se déforme au niveau des points de contrôles très voisins.</p>

Les outils de haut niveau en modélisation pour la synthèse d'images font appel aux NURBS (Non Uniform Rational B-Splines) qui sont une évolution des NRUBs (« B-Splines » non rationnelles uniformes) pour permettre un affichage en perspective. C'est pourquoi « proJET Easy Topo » adopte les **Nurbs** comme type de lissage par défaut, mais l'utilisateur peut choisir parmi les autres formes selon la présentation des points de la ligne polygonale à lisser.

5) La production de MNT :

a) Les principales méthodes :

Il ya plusieurs procédés pour la création de MNT : [8]

- Interpolation sur les courbes de niveaux : on aura un MNT rectangulaire.

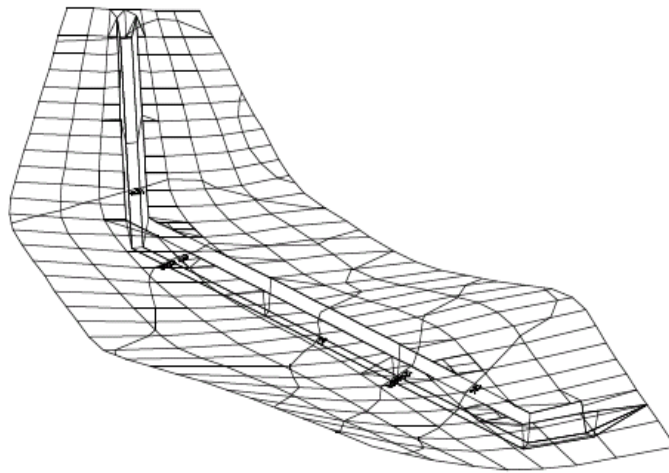


Figure 6 : Modèle numérique de terrain, Rectangulaire

- Triangulation de DELAUNAY ou TIN (Triangulated Irregular Network): on aura un MNT triangulaire.

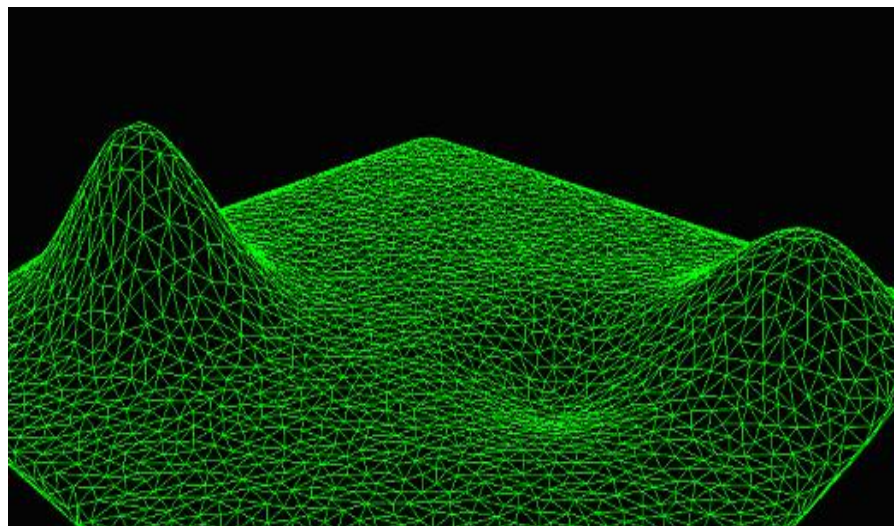


Figure 7 : Modèle numérique de terrain, Triangulaire

- Corrélation automatique et appariement d'images :

Le traitement se fait par couple de photos (stéréoscopie). Le logiciel de traitement des images détermine la valeur biométrique des pixels et calcule le coefficient de corrélation. Le traitement est beaucoup plus long si la résolution de l'image est grande.

- Télémétrie LASER (ou LIDAR : Light Detecting And Ranging) : on aura un MNE ou modèle numérique d'élévation ou DEM. (Digital Elevation Model)



Figure 8 : Vue en perspective du MNT et du MNE

Nous avons choisi pour notre logiciel de programmer l'algorithme de DELAUNAY qui est la plus efficace et la plus utilisée dans le domaine du Génie Civil.

Avec le nombre de points cotés que nous disposons, nous aurons à travers l'Algorithme de Delaunay, le nombre de triangles donné par les formules suivantes :

$$\text{nb_triangles} = 2 * (\text{nb_points} - 1) - \text{nb_côtés_du_périmètre} \quad (2.16)$$

$$\text{nb_segments} = 3 * (\text{nb_points} - 1) - \text{nb_côtés_du_périmètre} \quad (2.17)$$

b) L'algorithme de DELAUNAY : [8]

Nous traiterons ici la présentation de l'algorithme de triangulation de Delaunay à travers notre programme permettant de générer des terrains 3D.

La triangulation de Delaunay présente deux algorithmes qui adoptent la technique incrémentale, c'est-à-dire nous partons d'une triangulation et, en ajoutant un point coté, nous recalculons les triangles. Ces deux algorithmes sont :

- Triangulation par Destruction-construction
- Triangulation par Vérification : Côtés légaux- illégaux

b.1. La triangulation par Destruction-construction :

Elle se fait à partir d'une triangulation partielle du semis de points, et se poursuit par la triangulation existante augmentée d'un point. Donc, en partant de quatre points de départ formant deux triangles (donc quatre points triangulés : ABCD), l'algorithme rajoute des points (exemple : E) qui sont intérieurs à la triangulation et qui sont triangulés au fur et à mesure.

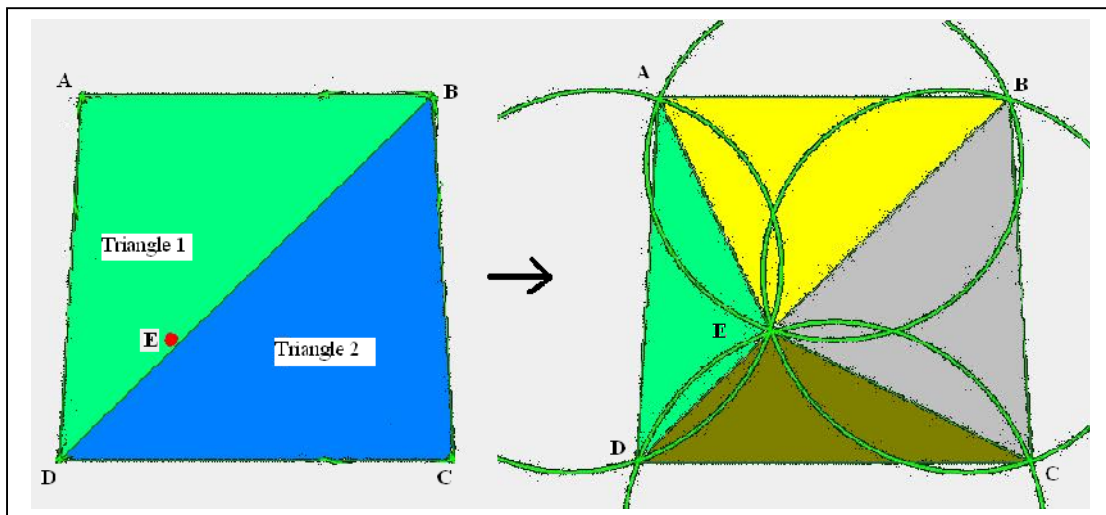


Figure 9 : Triangulation par Destruction-construction

Nous procédons ainsi : on utilise un triangle et un point choisi comme indiqué ci-dessus, puis tous les triangles dont le cercle circonscrit contient le point considéré sont détruits. Une fois les triangles illégaux détruits, on construit ceux qui sont engendrés par ce nouveau point, en ajoutant des arêtes reliant au point inséré les extrémités de celles qui n'appartenaient qu'à un seul triangle présent dans la liste de ceux qui ont été supprimés.

On utilise donc principalement deux méthodes : une fonction récursive qui détermine les triangles à éliminer, l'autre les triangles à construire. Voici l'algorithme général en pseudo-code : [8]

Algorithme général

```

Fonction inserePoint()
{
//On choisit le plus grand triangle
Triangle t = PremierTriangle();

//Choisir un point dans le triangle
Point p = ChoisirPoint(t);

//Initialiser une liste vide pour
la destruction
Liste ld = InitialiseListeVide() ;
lc = InitialiseListeVide();

//On detruit les triangles qui
posent problemes
detruireTriangles(p,t,ld);

//On construit les triangles
construireTriangles(p,ld,lc);

//On ajoute les triangles
AjouteTriangles(lc,lt);
}

```

(2.18)

Cet algorithme suit bien le paragraphe précédent. Montrons maintenant l'algorithme de la fonction de destruction : [8]

Destruction des triangles

```

Fonction detruireTriangles(Point p, Triangle t,
Liste ld)
{
//Ajoute le triangle t a la liste l
AjouteTriangle(l,t);

//On regarde les triangles voisins
Pour chaque triangle tv voisin de t
Si le triangle tv n'appartient pas a ld alors
Si p est inclu dans le cercle circonscrit de tv
alors
detruireTriangles(p,tv,ld) ;
FinSi
FinSi
}

```

(2.19)

Cette fonction est donc très simple. On ajoute le triangle à la liste des triangles à détruire, et on traite chaque triangle voisin. Si le cercle circonscrit d'un triangle voisin inclut le nouveau point, on appelle la fonction récursivement (donc on l'ajoutera à la liste et on regardera ses voisins). On regarde le cercle circonscrit seulement si le triangle n'appartient pas déjà dans la liste.

Enfin, voici l'algorithme de la fonction de construction. [8]

Construction des triangles

```

Fonction construireTriangles(Point p, Liste ld, Liste lc)
{
    Pour chaque triangle t de la liste ld
    Pour chaque voisin tv du triangle t
    Soit p1,p2 les deux points du côté en commun avec t et tv
        nouveaut = CreerTriangle(p,p1,p2)

    //On met a jour l'adjacence entre les triangles tv et
        nouveaut
        GererAdjacence(tv,nouveaut)

    //Ajouter le triangle nouveaut dans une liste lc
        AjouteTriangle(lc,nouveaut)
        FinPour
    FinPour

    //Gerer les adjacence entre les triangles
        GererAdjacenceEntreTriangles(lc)
}

```

(2.20)

b.2. La triangulation par Vérification : Côtés légaux- illégaux :

Cet algorithme est plus rapide que le premier puisqu'il a, en principe, moins de travail à faire. C'est une grande fonction réursive (qui pourrait être implémentée en itératif), qui vérifie si les côtés sont légaux. Un côté est défini comme étant l'arête commune à deux triangles. Pour savoir s'il est illégal, nous regardons les points qui ne sont présents que dans un des triangles (donc les points opposés). La figure suivante montre un côté légal entre deux triangles, comme on le voit, les cercles circonscrits ne contiennent pas le point opposé.

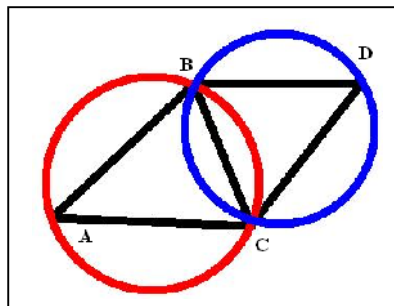


Figure 10 : [BC] un côté légal

Un côté est considéré comme étant illégal si au moins un de ces deux points opposés est contenu dans le cercle circonscrit du triangle opposé.

Si c'est le cas, le côté illégal est remplacé par un côté relié par les deux autres points opposés. Et on testera les quatre côtés qui restent pour voir si ce changement a provoqué un autre conflit. La figure qui suit montre un côté illégal et la transformation qui permet de rendre ce côté légal :

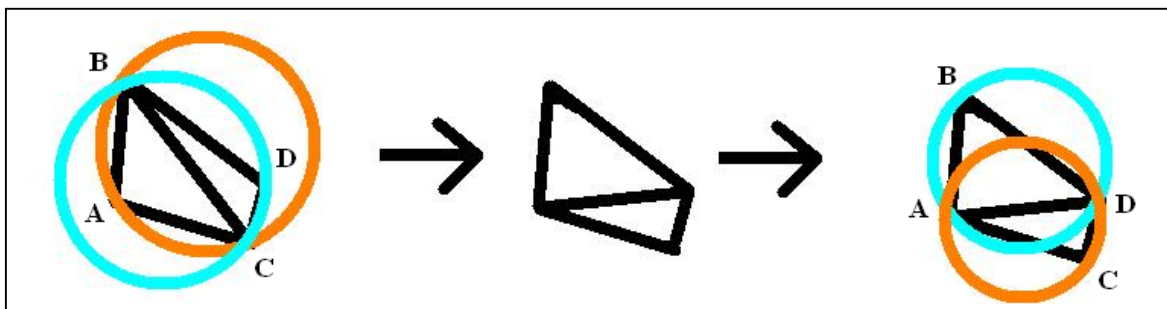


Figure 11 : Transformation d'un côté illégal en côté légal

Donc l'algorithme peut être décrit par :

- Insérer le point, détruire le triangle le contenant
- Créer trois triangles qui ont ce nouveau point comme sommet
- Vérifier chaque côté externe pour une illégalité.

La vérification des côtés peut être donnée par l'algorithme : [8]

Algorithme de vérification

Pour les deux points opposés, vérifier s'ils sont dans le
cercle circonscrit du triangle opposé

Faire

Si le point est inscrit dans le cercle circonscrit du
triangle opposé alors

Inverser le côté.

Vérifier les quatre côtés externes pour une illégalité.

FinSi

Fin Pour

(2.21)

6) Les techniques du rendu virtuel en 3D :

Nous avons utilisé pour le rendu, le « package » DirectX7 for Visual Basic Type Library. Cette une extension d'application qui est utile pour la conception des fenêtres d'affichages graphiques 3D.

Ce package renferme des fonctions numériques préétablies que notre programme va appeler. Ces fonctions sont :

- Le dessin et la coloration de surfaces (triangle, quadrilatère, sphère, cône, cylindre)

```

Set f = frmFull3D.RMC.mDrm.CreateFace
f.AddVertex X1, Y1, Z1
f.AddVertex X2, Y2, Z2
f.AddVertex X3, Y3, Z3
f.SetColorRGB 0, 0, 1
Part.AddFace f
  
```

(Pour dessiner une triangle)

- La spécification de la position d'un repère (O, I, J, K) dans l'espace : translation, rotation.

```

frmFull3D.RMC.mFrO.GetPosition Nothing, xVecteur
frmFull3D.RMC.moovepoint.SetBack 500
MForm.RMC.mVpt.SetFront -500
frmFull3D.RMC.mFrO.SetPosition Nothing, xVect.x, xVect.y, xVect.z
frmFull3D.SetZoom
mFrO.SetRotation Nothing, dY, dX, 0, Theta
  
```

- L'ajout de lumière virtuelle qui permet des effets comme l'éclairage du soleil et l'ombrage due aux reliefs (MNT).

```

Set DxL1 = mDrm.CreateLightRGB(D3DRMLIGHT_DIRECTIONAL, 0.7, 0.7, 0.7)
Set DxL2 = mDrm.CreateLightRGB(D3DRMLIGHT_AMBIENT, 0.8, 0.8, 0.8)
mFrL.AddLight DxL1
mFrL.AddLight DxL2
  
```

- La modification de la qualité du rendu :

```

Part.SetQuality D3DRMRENDER_FLAT
Part.SetQuality D3DRMRENDER_GOURAUD
Part.SetQuality D3DRMRENDER_PHONG
Part.SetQuality D3DRMRENDER_UNLITFLAT
Part.SetQuality D3DRMRENDER_WIREFRAME
  
```

PARTIE III - IMPLEMENTATION DES ALGORITHMES DE CALCULS TOPOGRAPHIQUES

I - REALISATION DU PROGICIEL DANS L'ENVIRONNEMENT DE DEVELOPPEMENT DU LANGUAGE VISUAL BASIC

1) Choix de Visual Basic :

Visual Basic est l'un des langages de programmation les plus performants et les plus simples à utiliser. Il a été créé par John G. Kemeny et Thoams E. Kurtz en 1963 et devient vite un langage populaire. Bill Gates a adapté Visual Basic sur PC, au milieu des années 70. Depuis, plusieurs versions améliorées pour PC ont vu le jour par exemple Microsoft QuickBasis et MS-DOS Qbasic.

La simplicité du langage explique son choix pour le développement d'interfaces.

Notre principal outil de développement est le **Visual Basic 2006 Entreprise Edition**, compte tenu de sa capacité de contrôler les objets par la technologie « OLE Automation » en utilisant la fonction du modèle d'objet composant COM (Component Object Model).

2) Les bases du langage Visual Basic : [2]

Le développement d'un programme sous VB se base sous trois étapes :

- Création de l'interface utilisateur à l'aide des contrôles VB
- Définition des caractéristiques ou propriétés des éléments qui composent l'interface
- Ecriture du code de programmation pour un ou plusieurs éléments de l'interface en fonction des besoins.

Les instructions suivront toujours la relation :

VARIABLE=VALEURS

(3.1)

On a à déclarer ces variables selon leur type voulu, par exemple Entier ou Double ou Caractère ou Date, etc.

3) Concepts de base de la programmation orientés objet : [1]

Le processus est assimilable au jouet d'enfant dit « LEGO ». C'est un jeu de construction en plastique, à pièces emboîtables de couleurs variables par lesquels l'enfant construit par exemple une maison en assemblant ces différents petits objets. On développe donc notre logiciel grâce aux objets comme : boutons, fenêtres, tableaux, lignes 2D-3D, fichiers numériques...

Ces objets ont leurs propriétés ou attributs, par exemple la couleur de la ligne N°1 est jaune. Donc celle-ci est représentée : **line1.color = vbYellow**

a) Objet et classe :

Un objet est une entité autonome qui regroupe un ensemble de propriétés cohérentes et traitement associé, d'où il dispose d'une partie statique et d'autre dynamique.

Une classe est un type de donnée abstrait caractérisé par des propriétés et des méthodes communes à des objets. Une classe est un modèle commun à un ensemble d'objet de même caractéristique.

b) Références d'objets pour notre programme :**b.1. OLE Automation :**

Contrôle les différents types d'objets d'une application qui supporte "Automation", par exemple l'application AutoCAD.

b.1. Directx7 for Visual Basic Type Library:

Cette référence est utile pour le contrôle des fenêtres graphiques en 2D et 3D de « proJET Easy Topo ».

b.2. Microsoft office 12.0 Object library :

Celle ci permet à notre logiciel d'importer ou de lire les fichiers d'extensions « *.xls » ou « *.xlsx ». Nous pouvons aussi exporter nos données issues des traitements vers un classeur Excel.

Ainsi pour l'importation des données Excel, on emploie les collections d'objets suivantes :

Workbooks qui veut dire Classeurs, actuellement ouverts dans Microsoft Excel.

Voici quelques exemples le plus utilisés :

Workbooks.Open « VONTOVORONAxzy.XLS » : notre programme ouvre le fichier « vontovoronaxyz.xls » sous la forme d'un classeur.

Workbooks.Add : notre programme ajoute automatiquement un nouveau classeur vierge dans Microsoft Excel.

Workbooks.Cell (i, j) : notre logiciel exporte une donnée traitée vers Excel. Il va l'écrire sur la cellule (i, j) = (Ligne, Colonne) du classeur Excel.

Worksheet(1) et **Worksheet(2)** sont par exemple la feuille1 et la feuille 2 dans le Classeur1.

La figure suivante représente le Modèle d'objet Application MICROSOFT EXCEL.

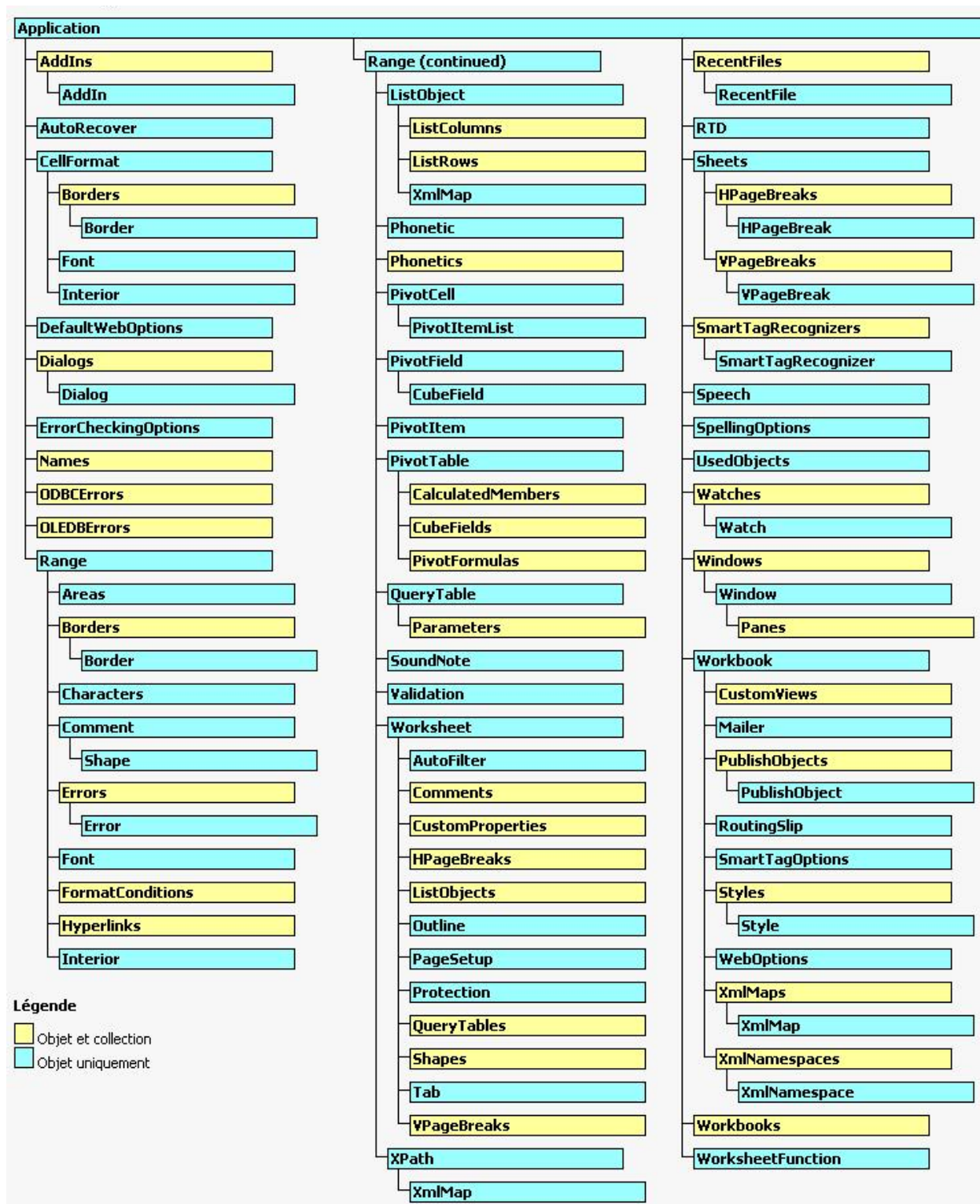


Figure 12 : Modèle d'objets Microsoft Excel

b.3. AutoCAD 2008 Type Library :

Vu que « proJET easytopo » comprend quelques fenêtres de dessin, on l'a conçu aussi pour pouvoir utiliser la fenêtre graphique d'AutoCAD 2008 pour le report automatique des points traités. On a choisi AutoCAD 2008 car des sondages entrepris au niveau des utilisateurs nous ont fait savoir sa facilité de manipulation. Son interface 3D aussi est très performante et plus complète que des versions d'AutoCAD antérieures.

Les principales instructions utilisées dans notre application sont :

Object.AddPoint (point) : crée un objet Point dans l'espace objet à la coordonnée (X, Y, Z)

Object.AddLine (StartPoint, EndPoint) : permet de tracer une ligne avec un point de départ et un point d'arriver.

Object.AddAttribute(Height, Mode, Prompt, InsertionPoint, Tag, Value) : Cette méthode requiert la saisie de six valeurs : la hauteur du texte de l'attribut, le mode de l'attribut, une chaîne de message, le point d'insertion, la chaîne d'étiquette et la valeur par défaut de l'attribut.

Object.Add3DFace (Point1, Point2, Point3[, Point4]) : permet à notre logiciel de créer des surfaces 3D à partir de quatre points de MNT, ou trois points s'il s'agit d'une surface triangulaire.

La figure ci-après représente le Modèle d'objet Application AutoCAD 2008 :

[5]

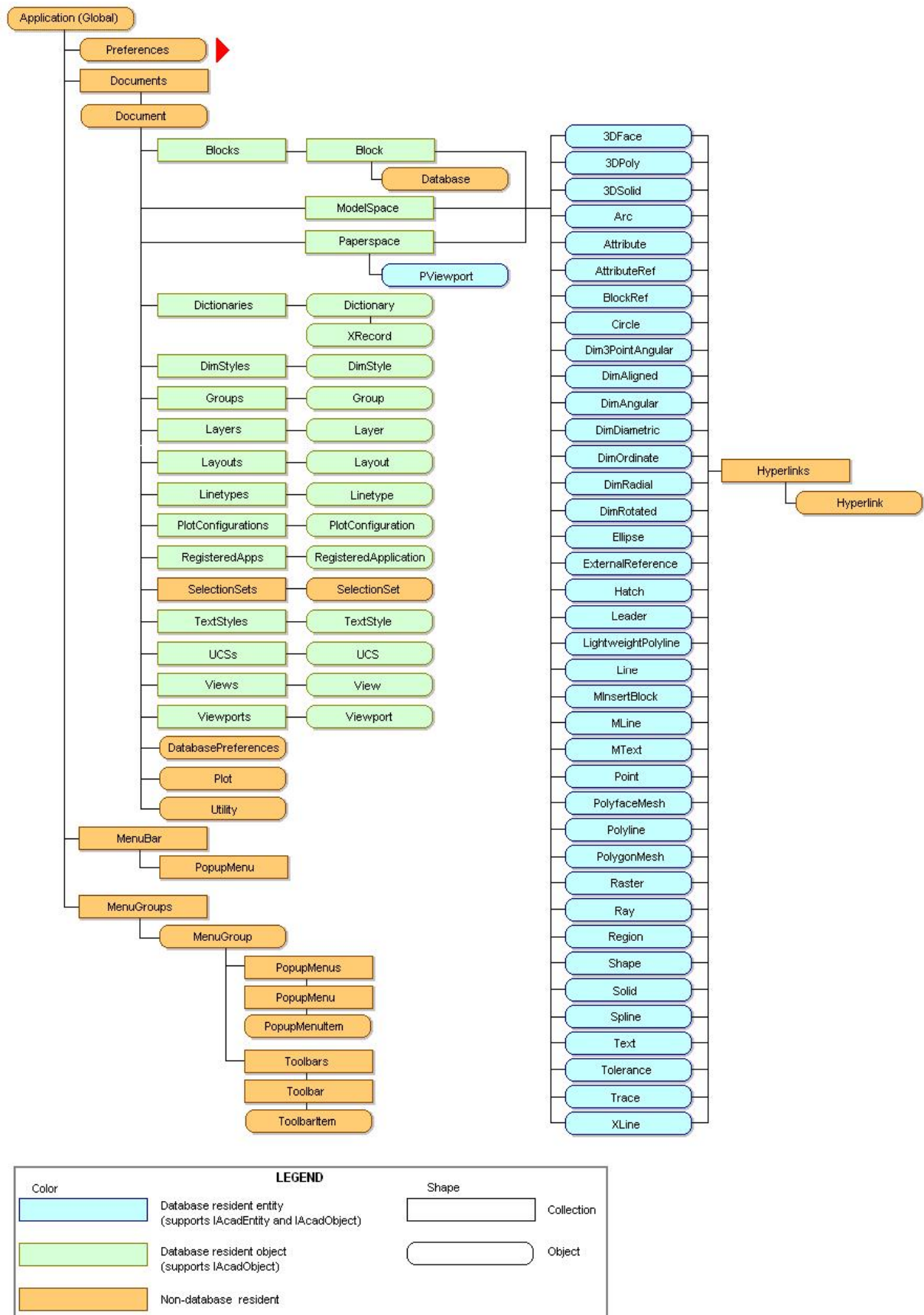


Figure 13 : Modèle d'objet Application AutoCAD

II - LES DIFFERENTES ETAPES DU DEVELOPPEMENT

1) Modélisation de l'application

a) Diagramme des classes

Le mode de présentation du diagramme des classes est la méthode « Unified Modeling Language » abrégée par UML. Nous avons utilisé cette modélisation en UML car notre application présente non seulement les données mais aussi les opérations ou les calculs à effectuer. Ces caractéristiques diffèrent la présentation UML des autres formes comme : le schéma HBDS, la méthode Merise,... Ces autres formes mettent seulement en relation les données comme dans la création des bases de données ordinaires.

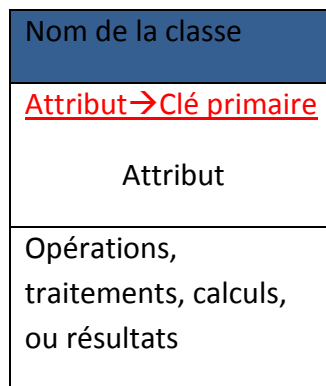


Figure 14 : Structure d'une classe sous forme UML

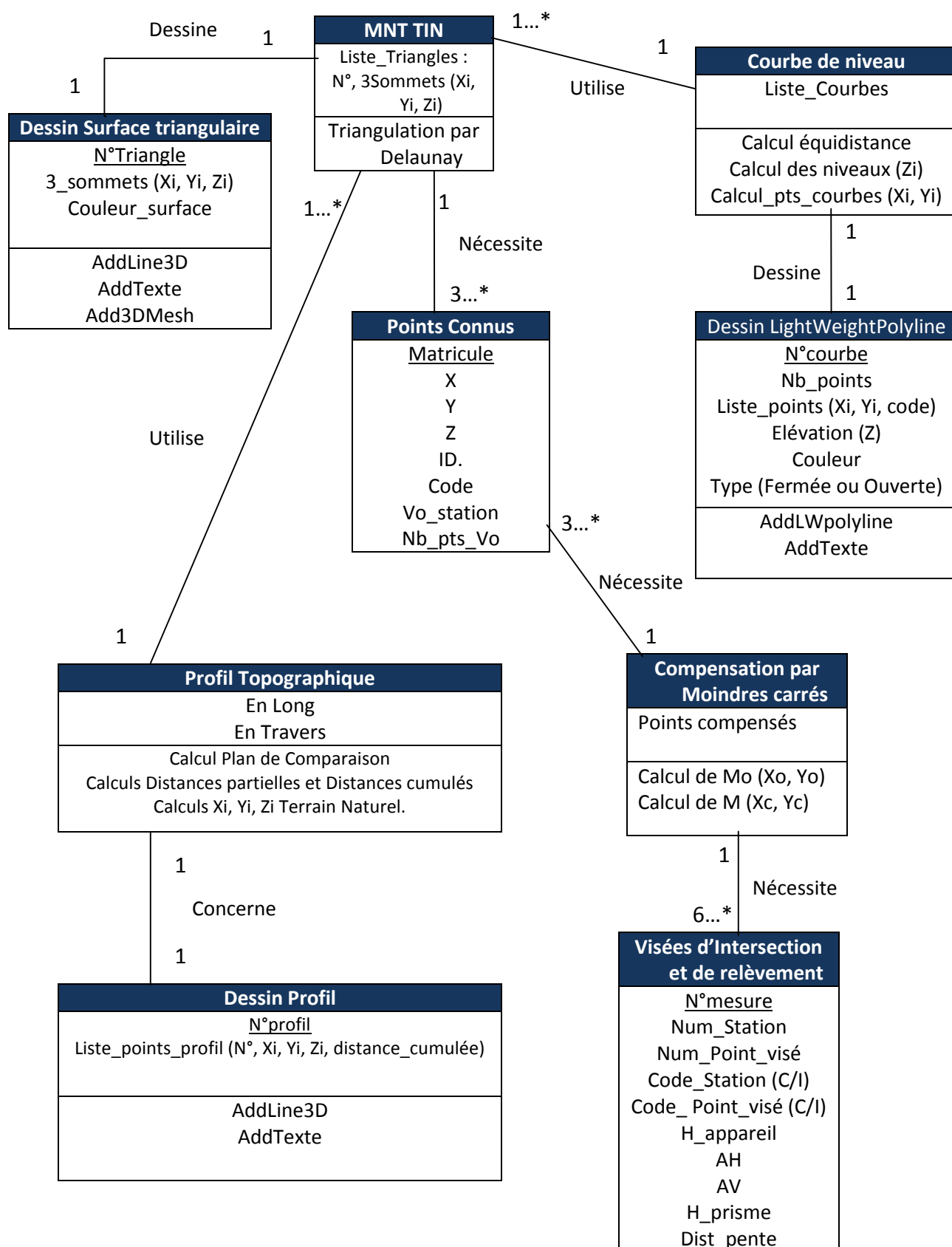


Diagramme 1 : Diagramme des classes du programme

2) Création de l'application : [4]

a) Phase de codage : (Organigramme et Algorithme des calculs)

a.1. L'organigramme des calculs :

L'organigramme demeure le premier outil important pour la mise au point d'un programme, même si le programmeur est expérimenté ou doué. Elle est fondée sur la présentation initiale du problème, mais elle se limite seulement à l'organisation. La description du programme vient après.

L'organigramme est composée de quelques symboles graphiques comme rectangle, losange, cercle...

L'exemple ci-après montre un organigramme de notre programme :

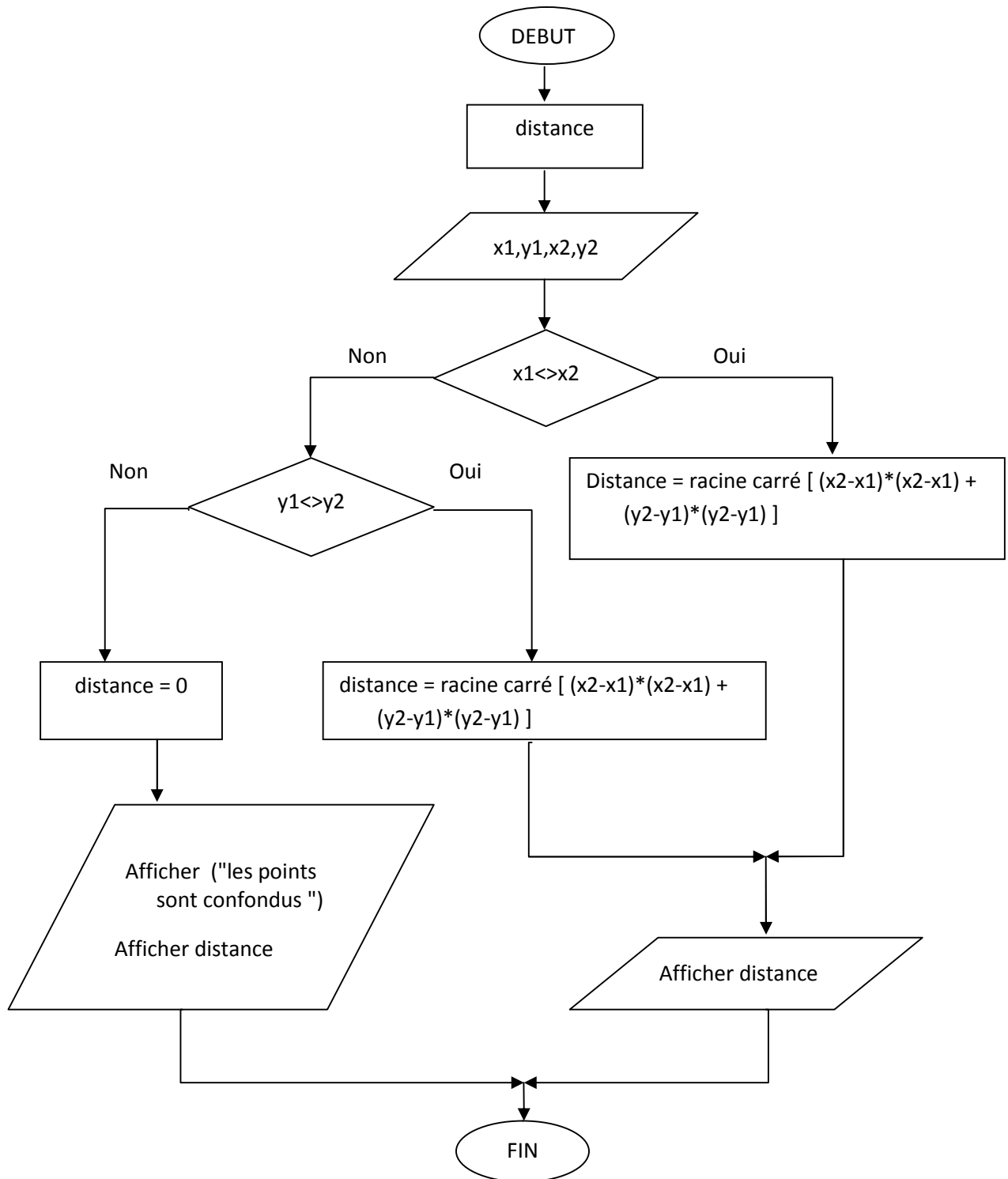


Diagramme 2 : Calcul de distance planimétrique entre 2 points topographiques

a.2. L'Algorithme des calculs :

Un algorithme fait passer d'un état initial à un état final de façon déterministe. Il doit respecter les règles suivantes :

- Il est **défini** sans ambiguïté, lisible et compréhensible par un non-informaticien.
- Il se termine après un nombre fini d'opération et ne doit pas dépasser une page, si oui, on va décomposer le problème en sous-problèmes.
- Il doit être **effectif** c'est-à-dire toutes les opérations doivent pouvoir être effectuées par un homme utilisant des moyens manuels.
- Il doit être de **haut niveau** : cela veut dire qu'on peut le traduire en n'importe quel langage de programmation.
- Il manipule des **objets** qui doivent être définis de manière précise.

Dans un algorithme on doit identifier 3 parties successives :

- La partie indication du nom de l'algorithme
- La partie déclaration des variables, précédée du mot clé « variable »
- La partie instruction des opérations, commençant par le mot clé « Début » et finissant par le mot clé « fin ».

Exemple d'algorithme de notre programme :**Algorithme :** calcul_de_la_distance_planimétrique_entre_2_points_topographiques**Variables :** x1, y1, x2, y2, distance : Réelles ;**DEBUT**

{ Saisir : x1, y1, x2, y2 ;

Tester que : $x1 \neq x2$

Si oui :

 { distance = racine carré [$(x2-x1)^2 + (y2-y1)^2$] ;

Afficher (distance) ;

}

Si non :

 { **Tester que :** $y1 \neq y2$

Si oui :

 { distance = racine carré [$(x2-x1)^2 + (y2-y1)^2$] ;

Afficher (distance) ;

}

Si non :

{ distance = 0 ;

Afficher (distance) ;

Afficher ("les points sont confondus ")

}

}

}

FIN**(3.2)**

b) Création de la source de données et gestion des fichiers

Le logiciel travaille dans les types de fichiers suivants :

➤ Fichiers Easy-Topo V.02 :

Ce sont des fichiers textes d'extensions : *.top, *.mnt, *.tri, *.xyz, *.obs, *.lon, *.zzz .

➤ Fichiers Excel : *.xls, *.xlsx

➤ Fichiers d'échange international : *.DXF

➤ Fichiers AUTOCAD : *.DWG

Les fichiers en entrées pour notre logiciel exigent les systèmes de formats comme suit :

➤ Pour les calculs de compensation par Moindres carrées :

➔ Fichier d'observations sur terrain :

Station/Point_visé/code_station/code_point_visé/Hauteur_d'appareil/Angle_horizontal/Angle_vertical/Hauteur_prisme/distance

```
Station, Ptvisé, CodeSt, CodePv, Happareil, AH, av, Hprisme, dist
-----
"Ankitsika","Bendravola","C","I",1.5,179.03704,0,1.5,0
"Ankitsika","Antsongosongo","C","I",1.5,59.88849,0,1.5,0
"Ambohitsimanaiky","Bendravola","C","I",1.5,68.10921,0,1.5,0
"Ampamoizana","Bendravola","C","I",1.5,382.09121,0,1.5,0
"Ambatomirahavavy","Bendravola","C","I",1.5,389.89554,0,1.5,0
"Ambatomirahavavy","Ankarahara","C","I",1.5,84.89938,0,1.5,0
"Andranotokana","Ampandihizina","C","I",1.5,311.20833,0,1.5,0
```

➔ Fichier des points connus en coordonnées :

Station/Xv/Yv/Z/Vo_station/nombre de points pour la détermination de Vo

```
station Xs Ys Zs v0 nv0
-----
"vohilomotra",597672.04,986679.5,0,210.19658,11
"Ambatomirahavavy",612497.5,996185.87,0,216.44706,6
"Andrepatza",617035.17,1027509.49,0,322.32666,0
"vohitrambo",640389.36,1027654.19,0,81.07728,0
```

- Pour le calcul de MNT :

➔ Fichier de points cotés :

Matricule/X/Y/Z

```
PointN° x y z
-----
"1",214726.55,7913981.98,980.43
"2",214714.03,7913974.78,977.64
"3",214675.352,7914102.795,1008.191
"4",214704.47,7913987.7,977.52
"5",214651.09,7914000.1,971.93
```

- Pour l'édition de Profils en long :

Matricule/cote_terrain_naturel/cote_projet/Distance_cumulée/distance_partielle
/X/Y

```
N°Profil Cotes_TN Cotes_projet Dist_Cumul Dist_partiel X Y
-----
"A",53.59,0,0,0,585871.35,1416649.36
"0",53.95,0,0,9.93,585876.2,1416658.03
"1",54.19,0,0,1.24,585876.81,1416659.11
"2",54.12,0,0,1.07,585877.33,1416660.04
"3",53.84,0,0,6,585880.26,1416665.28
"4",53.81,0,0,4.66,585882.54,1416669.34
```

- Pour l'édition de profils en travers :

Matricule/distance d'application/abscisses des points de travers/cote des
points de travers

```
N°Profil Dh(m) abscisses Cotes_projet abscisses Cotes_TN
-----
"1",0,0,0,-10,110.95
"1",0,0,0,-5,111.05
"1",0,0,0,0,110.82
"1",0,0,0,5,110.76
"1",0,0,0,10,110.45
```

c) Création de l'interface : fenêtres, menu, commandes, tables...

L'interface comprend :

- Une fenêtre d'accueil : nom de marque, copyright, mot de passe d'utilisateur, logo, bouton 3D.
- Une fenêtre principale : MDI (*Multiple Document Interface*)
- Des fenêtres filles : MDI Child
- Des fenêtres d'ouverture et de sauvegarde de fichiers.
- Une fenêtre de visualisation des plans topographiques.

(Cf. Capture d'écran à la page 62)

d) Compilation de l'application

La compilation nous indique si les codes que nous avons tapé sont correctes. Cette phase révèle au programmeur si des oublis ont eu lieu, comme l'oubli de déclaration de variable par exemple, et elle corrige aussi les erreurs de syntaxe.

Si le compilateur ne détecte plus d'erreurs, on aura l'exécutable du logiciel qui sera d'extension « EXE-STANDARD », applicable sous Microsoft Windows.

3) Phase test

A la suite de ces étapes, le logiciel est presque prêt à l'utilisation. Nous faisons alors différents tests sur tout l'environnement du logiciel à citer :

L'interface : on vérifie l'esthétique, les dimensions de la fenêtre, la cohérence des commandes avec le code source.

Les types de données et fichiers : quelle type de donnée nous pouvons saisir à telle fenêtre, exemple : il y a erreur si au lieu d'employer « virgule » comme séparateur des décimales, l'utilisateur a saisi au clavier un « point ». On doit respecter les types de variables : on effectue des opérations sur des « nombres » alors que les résultats sont affichés en « caractères ».

Les algorithmes de calculs : un algorithme donne toujours des résultats, mais il reste à vérifier la véracité de ces résultats en les comparant avec des étalons. On peut se demander ici lors de résultats erronés si on n'a pas saisi la vraie formule de calcul.

La Compatibilité du logiciel (SOFTWARE) au système d'exploitation : pour notre logiciel, le système d'exploitation agréable est WINDOWS XP. On vérifie si nos algorithmes ne provoquent pas des erreurs fatales qui peuvent endommager WINDOWS.

Les Comportements des matériels (HARDWARE) pendant l'exécution du logiciel :

On pourra voir directement la normalité de nos algorithmes :

Une boucle infinie de traitements entraîne l'ordinateur à se planter et les programmes ne répondent plus aux instructions (le Processeur marche à font 100%).

L'utilisation des tableaux dynamiques se fait directement sur la mémoire vive (RAM). Cela va retarder le traitement si les données numériques à stocker dans ces tableaux sont volumineuses, à moins que nous ayons une carte RAM élevée.

Exemple de teste effectué sur ProJET EasyTopo sur un ordinateur pris comme échantillon :

Caractéristiques matériels : CPU **1,6 GHz**, RAM 1Go, Graphiques 384Mo

→ L'échantillon a une puissance faible au niveau du processeur

Calcul de MNT par l'algorithme de Delaunay :

Vitesse de traitements : 1 triangle / milliseconde

Programmes en exécution	Nombre de points topo	Durée de traitements	Nombre de triangles	Taille de mémoire RAM occupée	Etat du processeur CPU
ProJET Easy topo.exe	1000	15s	1961	25Mo	90%
AutoCAD2008.exe		45s		30Mo	
ProJET Easy topo.exe	5000	2min	9978	27Mo	100%
AutoCAD2008.exe		1min30s		35Mo	
ProJET Easy topo.exe	+ 5000	2min	-	27Mo	100% + (PAS DE REPONSE durant un petit laps de temps)
AutoCAD2008.exe		3min	-	45Mo	

Tableau 1 : Résultats sur la capacité de traitement du logiciel

III - Evaluation de la qualité et la précision de « proJET Easy-Topo » :

Un programme informatique donne toujours des résultats mais il reste à savoir si ces résultats sont justes et dans les normes professionnelles. Pour connaître la qualité et la précision de « proJET Easy-Topo », nous avons traité des données réelles issues des travaux topographiques sur terrain par deux méthodes :

La première comparaison s'est faite avec des calculs manuels et des méthodes graphiques analogiques. (Méthodes traditionnelles)

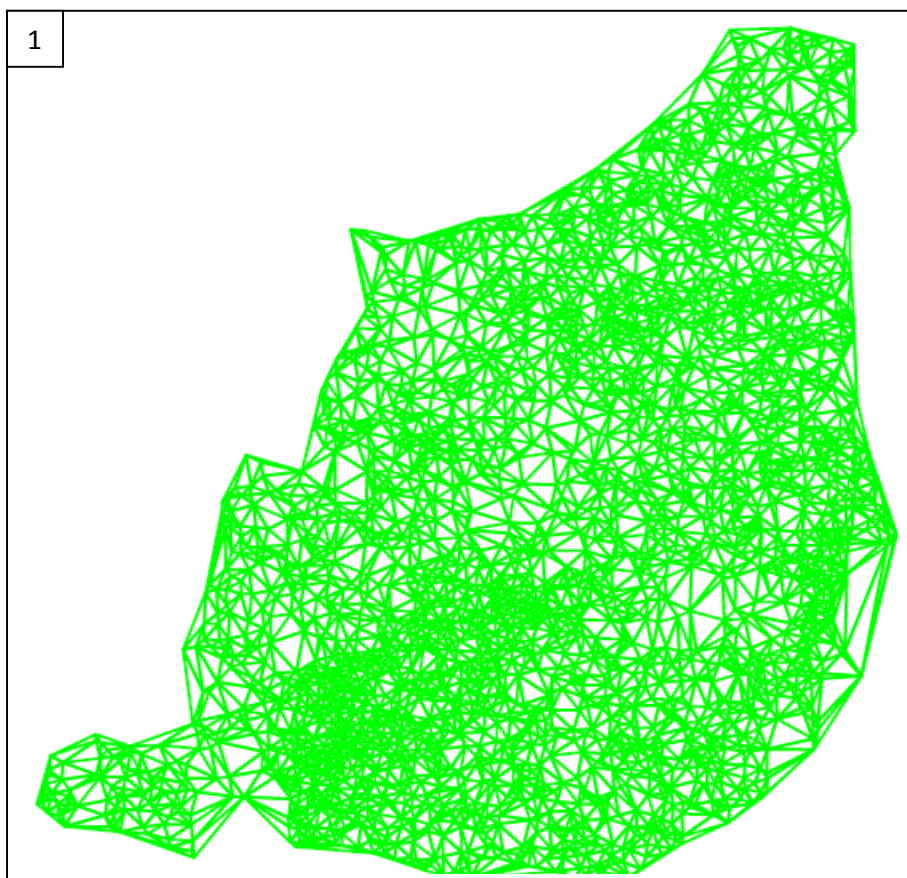
La deuxième comparaison a été faite par des logiciels professionnels.

Ces logiciels professionnels que nous avons utilisé sont :

COVADIS Projet Routier,

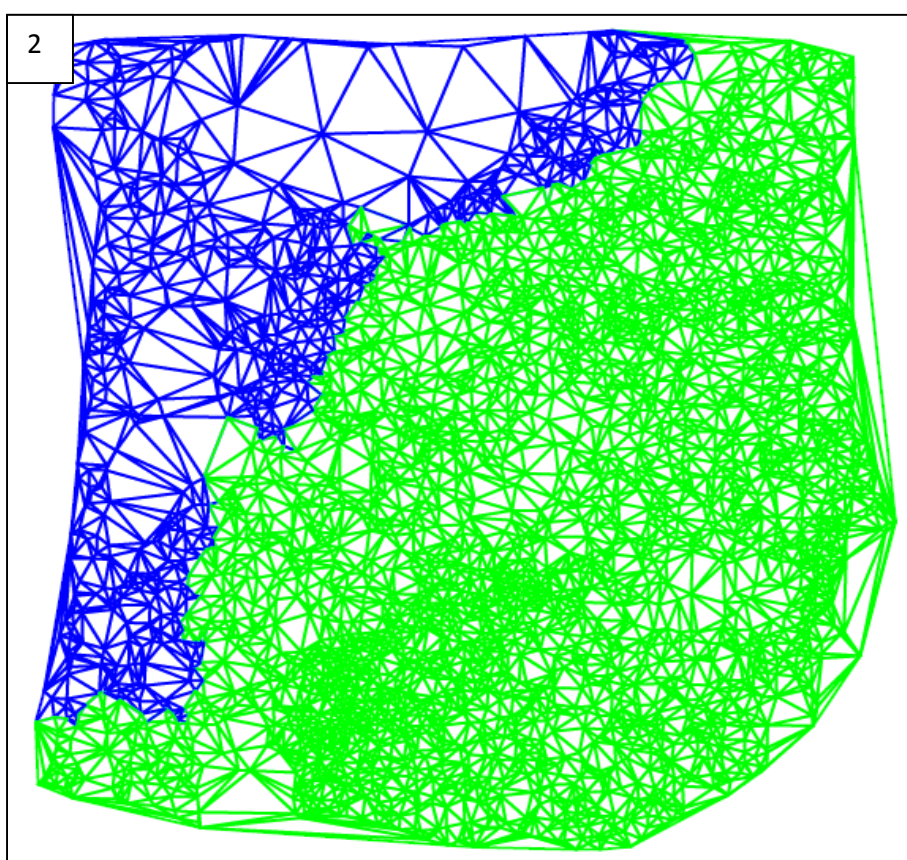
ArcView GIS 3.2,

Nous présentons ci-après ces résultats et ceux obtenus avec notre logiciel.

1) Calcul de MNT de 3917 points cotés de l'île Sakatia au Nord de Madagascar:

On voit que COVADIS a bien délimité la zone et a éliminé les triangles faux aux extrémités, mais il n'a pas tenu compte des points de niveau zéro qui représentent la surface de la mer.

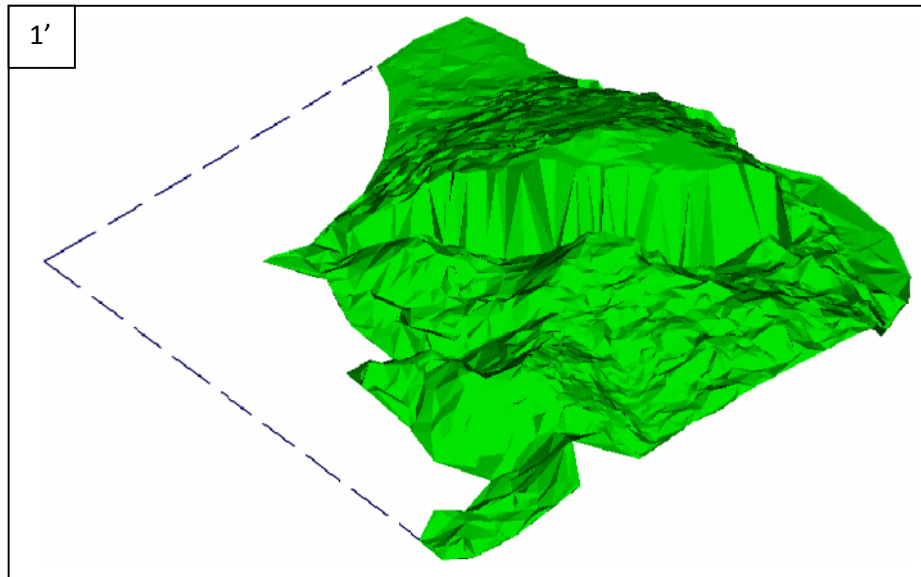
(Figure 1)



Easy Topo a filtré ces points et a affecté la couleur bleu pour un rendu 3D réaliste.

Nous voyons quelques différences sur les contours. Un ajout d'algorithme complémentaire au logiciel va permettre d'effacer ces imperfections.

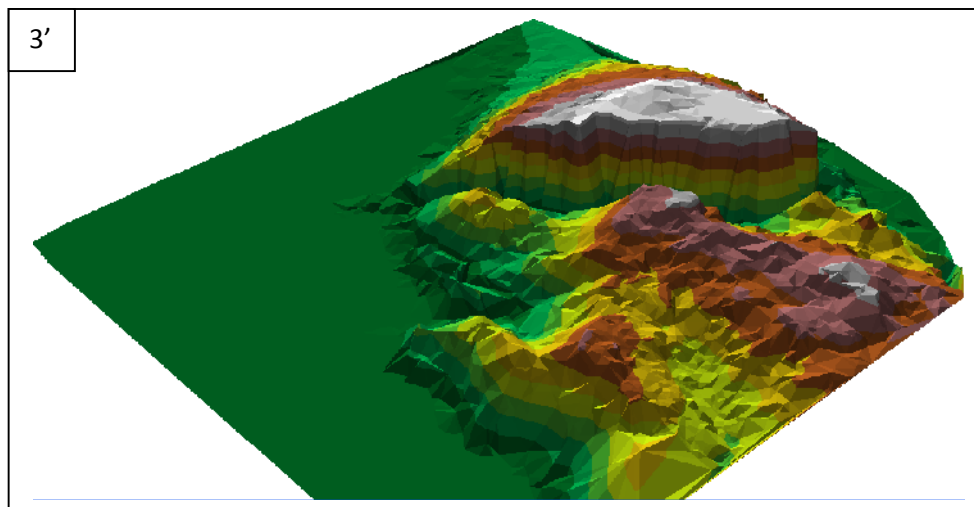
(Figure 2)



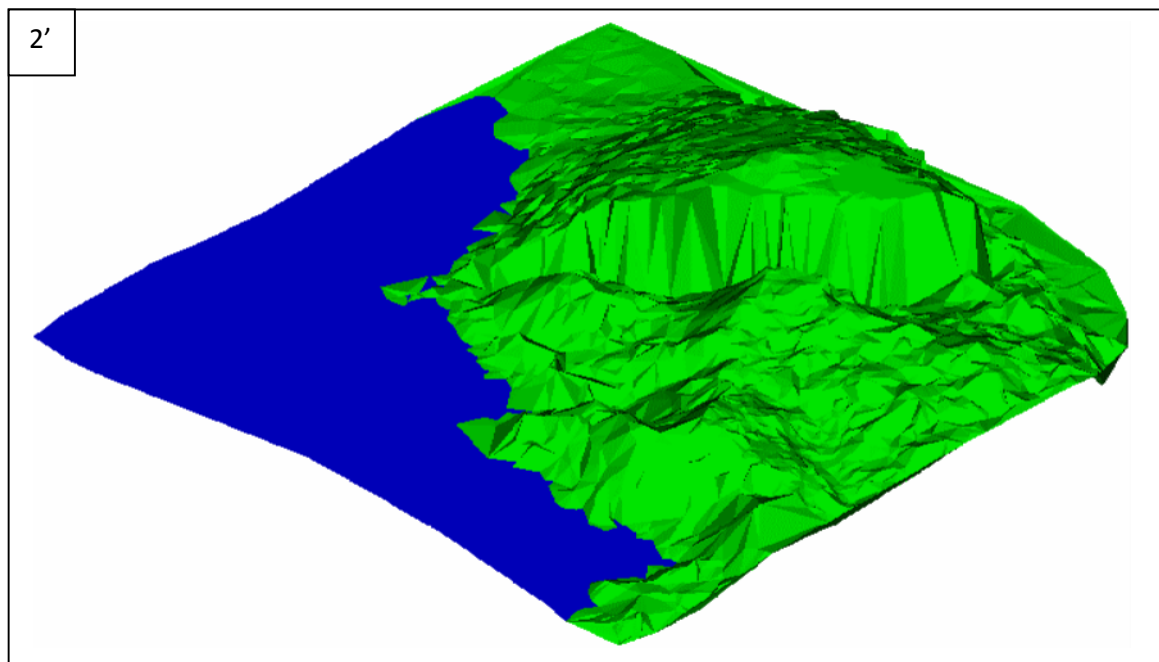
(Figure 1') Rendu 3D du MNT par COVADIS.

(Figure 3') Rendu 3D du MNT par ArcView.

Nous pouvons distinguer que l'île de SAKATIA présente une falaise.

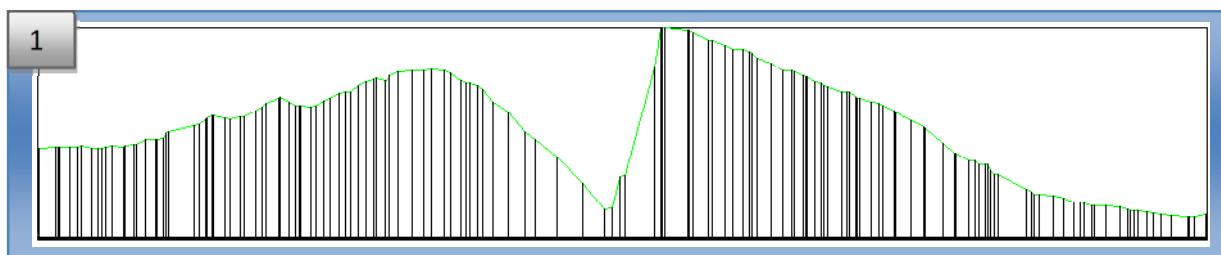


(Figure 2') Module modélisation 3D de « proJET Easy Topo », la surface coloriée en bleu représente celle de la mer.



2) Edition de profils topographiques par le MNT :

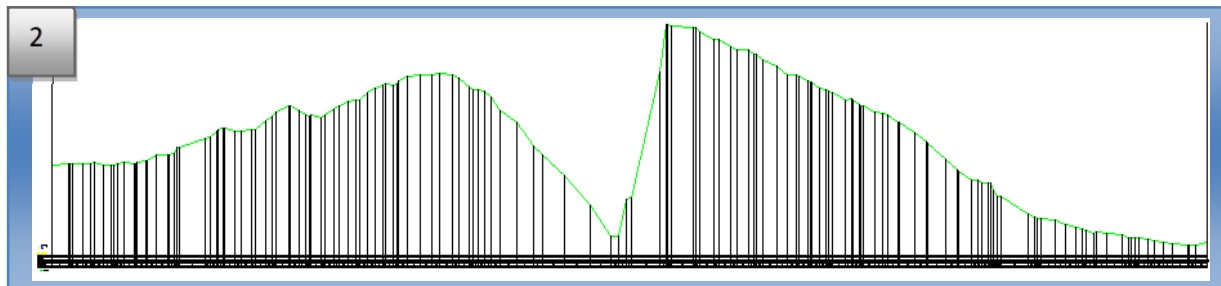
Profils en long Terrain Naturel :



1'	Echelle en X: 1/10 Echelle en Y: 1/10	
	PC : 0.00 m	
	Numéros des points TN	
	Altitudes TN	
	Distances cumulées TN	
	Distances partielles TN	
	Pentes et rampes TN	
	Altitudes Projet	
	Distances cumulées Projet	
	Distances partielles Projet	
	Alignements et courbes	

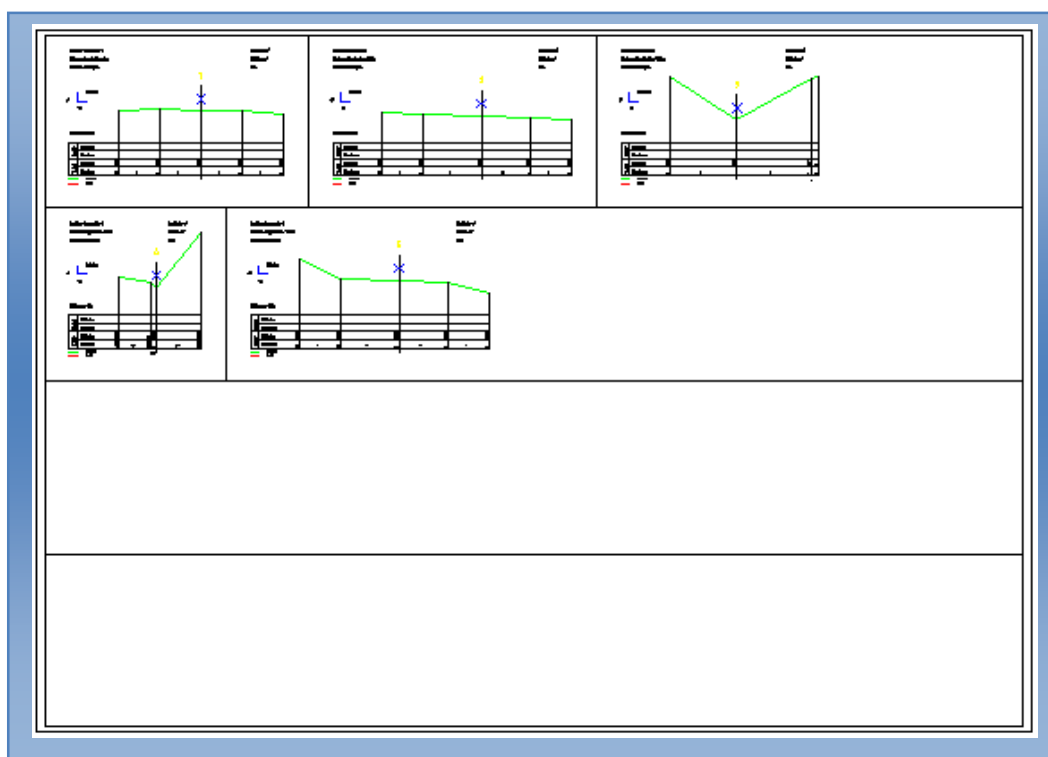
Les figures 1 et 1' sont les représentations du profil en long TN effectué par COVADIS :
Echelle : 10/10

Les figures 2 et 2', celles du profil en long TN effectué par EasyTopo : Echelle : 1/1 (à imprimer sur un rouleau de 40m par un traceur)



2'	Echelles (x,y)				
	1m				
	Plan de comparaison 7,00m				
	Cotes Terrain Naturel	53,95	54,15	54,15	
	Cotes Projet				
	Numéros des Profils	1	2	3	4
	Distances Cumulées	9,93	11,17	12,24	
	Distances Partielles	9,93	1,24	1,07	
	Informations				
	Terrain Naturel				

Profils en travers Terrain Naturel avec format d'Impression A0 :



Quand la première surface A0 sera remplie, notre logiciel va créer un autre format de papier A0 pour les profils qui suivent.

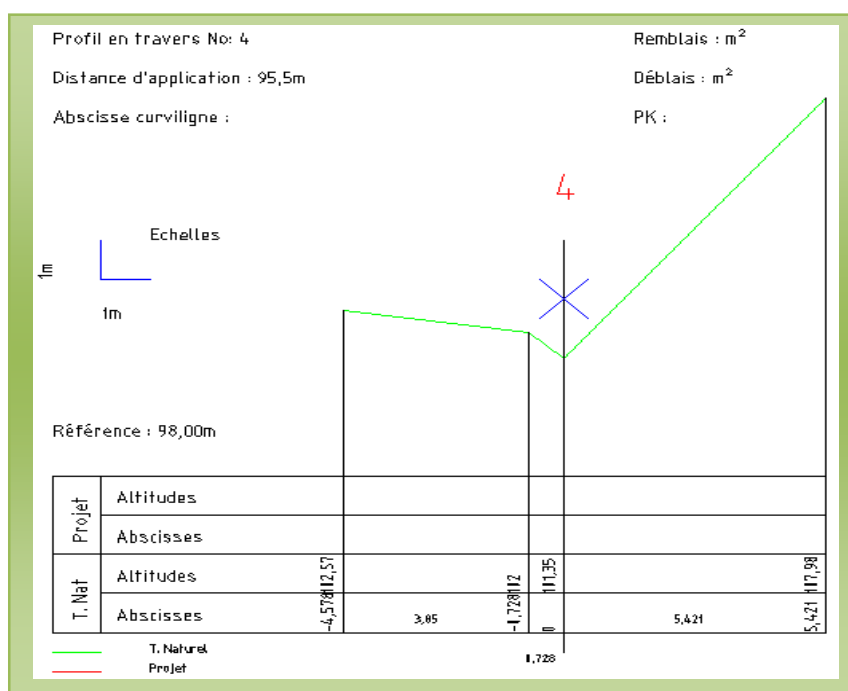
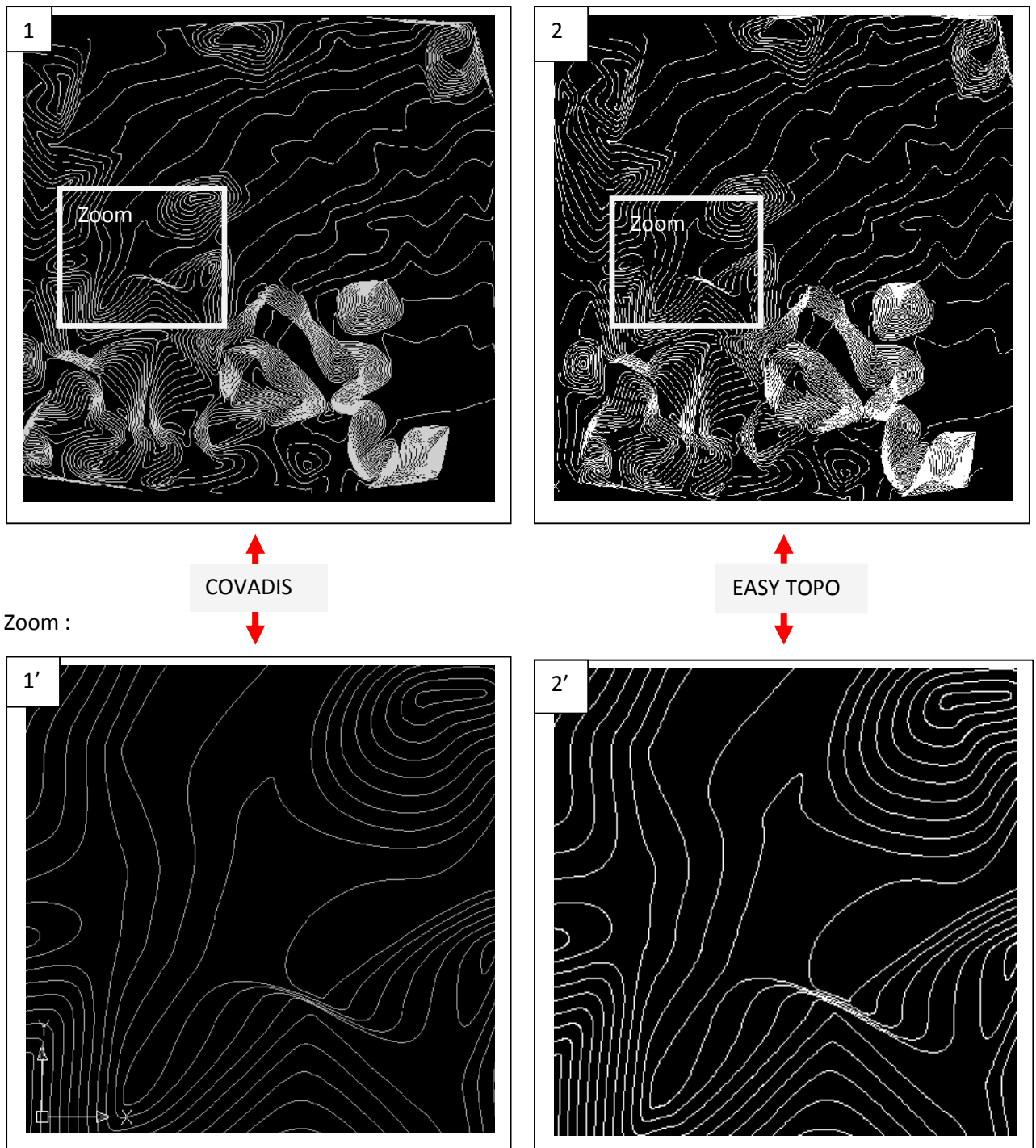


Figure 16 : Habillage du profil en travers

3) Création de courbes de niveaux :



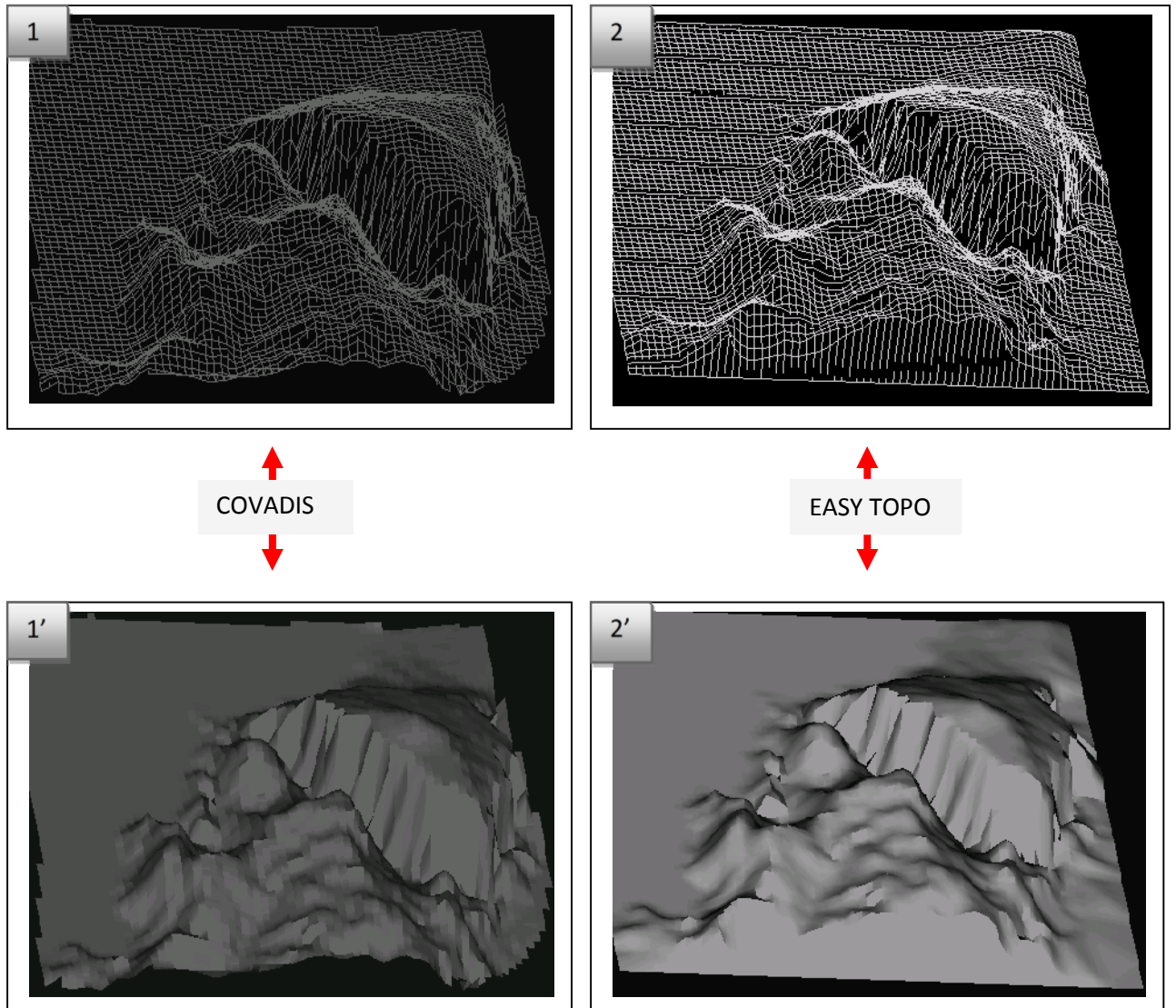
Les figures 1 et 1' sont les représentations des courbes de niveaux effectués par COVADIS.

Les figures 2 et 2' sont les représentations des courbes de niveaux effectués par EasyTopo.

Nous voyons une différence mais compte tenue de d'échelle de la carte (1 : 100000 par exemple), l'erreur est tolérable.

4) Calcul et rendu de maillage 3D :

Maillage régulier en pixel de dimensions 10m x 10m.



Rendu 3D du maillage avec lissage des surfaces et ombrages.

Figures 1 et 1' par COVADIS ; Figures 2 et 2' par EasyTopo.

PARTIE IV - INTEGRATION DU LOGICIEL DANS LE SYSTEME D'EXPLOITATION « WINDOWS »

I - PRESENTATION DU LOGICIEL :

En cohérence avec le logiciel de version 1.0 « Easy Topo », nous avons spécifié cette seconde version 2.0 comme suit :

Nom : proJET Easy Topo

Taille : 40 Mo

Version de Microsoft Excel : 1997-2003, 2007

Version d'AutoCAD : 2002 - 2008

Version du fichier DXF : R14 (possibilité d'être importé par ArcViews)

R 09 (ancienne version)

Systèmes d'exploitation compatibles : Microsoft Windows

Matériels requis : Micro-ordinateur (supportant AutoCAD 2008), Traceur
RAM 256 Mo ou plus
Processeur 1,6 GHz ou plus
Mémoire Graphique 256Mo ou plus (pour le relief 3D)

Packages(*) requis :

Microsoft Net. Framework 3.0 ou plus

Microsoft DirectX 7 ou plus

**Packages : ensemble de programmes informatiques installés, et exigés par un logiciel pour permettre son exécution dans le système d'exploitation et à optimiser leur fonctionnement.*

II - INSTALLATION ET DESINSTALLATION :

1) Installation :

Notre logiciel est composé de paquet de fichiers prêt à s'installer automatiquement en cliquant deux fois sur « SETUP.EXE ». Pour l'obtention de ce produit final, on a employé l'outil « Package and Deployment Wizard » de Microsoft Visual Basic 06.

L'installation se fait comme suit :

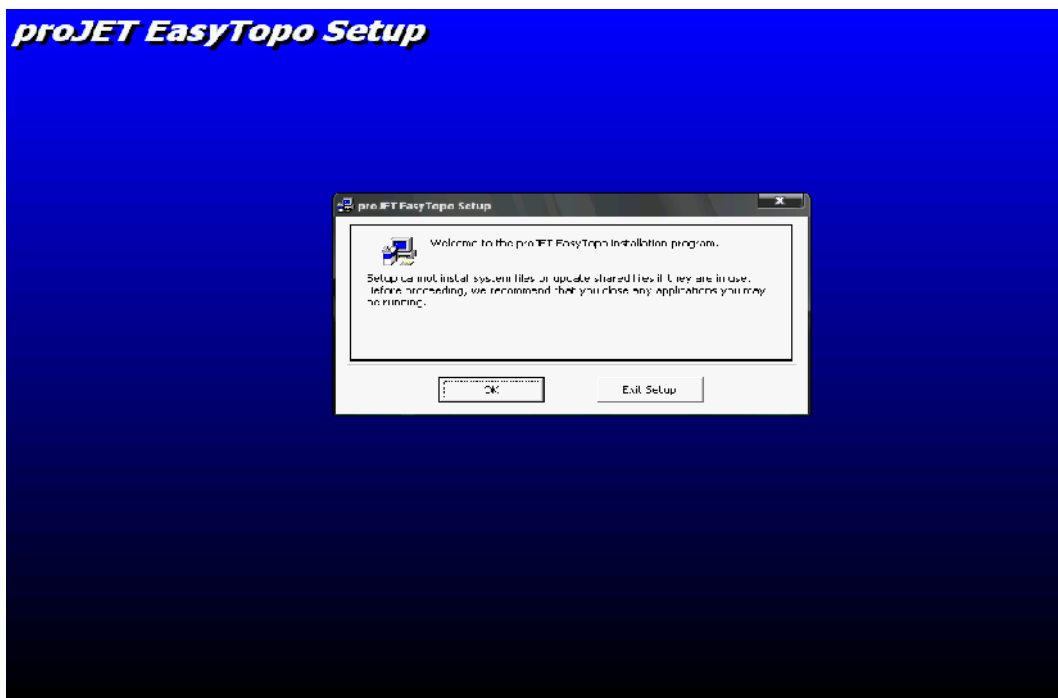
- ✓ Entrer dans **Panneau de configuration** → **Ajouter ou Supprimer des programmes**

- ✓ Vérifier si **Net Framework 3.0** est déjà installé

- ✓ Lancer **Setup.exe**



- ✓ Pour réussir l'installation, il faut suivre les instructions de chaque fenêtre de la procédure d'installation.

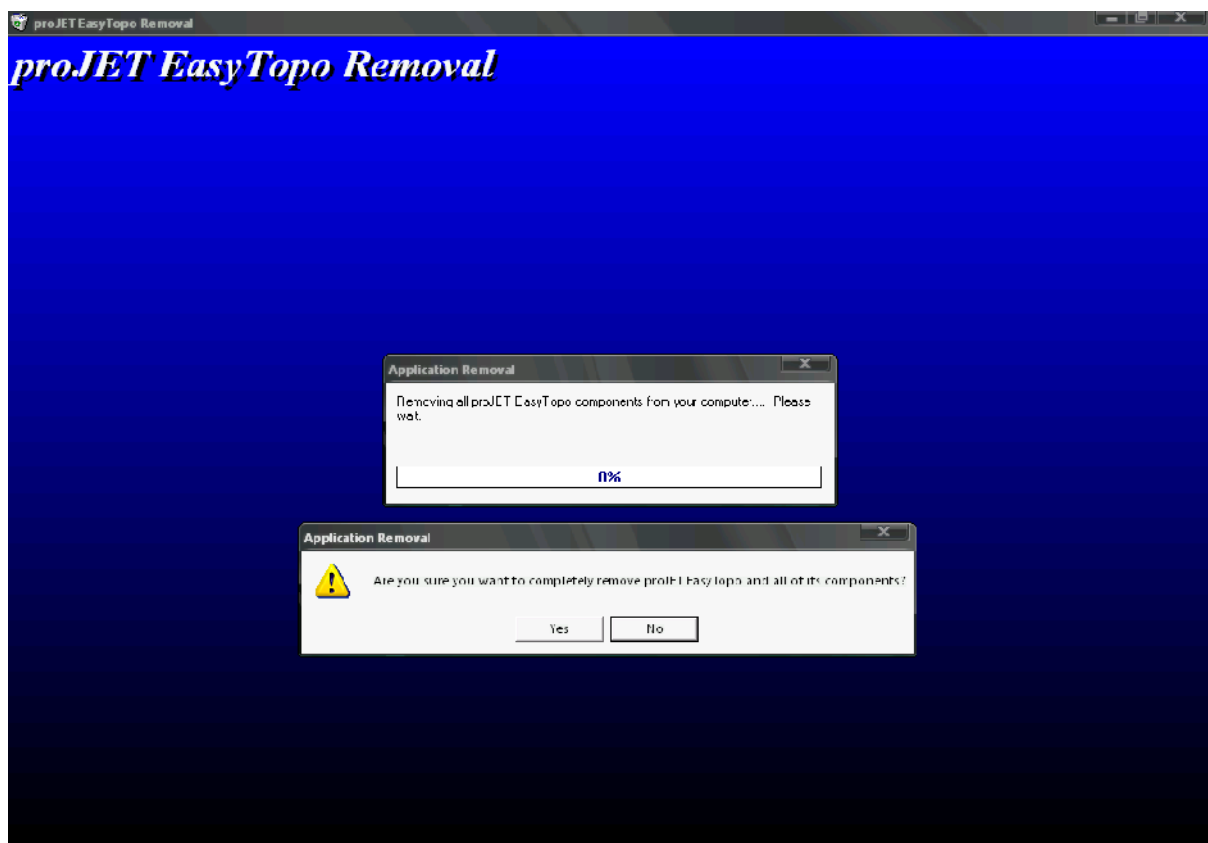


L'installation ainsi terminée, nous pouvons démarrer le programme en cliquant sur le bouton Démarrer → Tous les programmes → proJET EasyTopo.

2) Désinstallation :

Pour pouvoir désinstaller le logiciel proJET Easy Topo :

- ✓ Entrer dans **Panneau de configuration** → **Ajouter ou Supprimer des programmes**
- ✓ Sélectionner le logiciel proJET Easy Topo.
- ✓ Confirmer la suppression.



III - MODE D'EMPLOI

Avant les traitements, le logiciel demande de créer un nouveau projet ou d'ouvrir un projet existant. Tous les fichiers de sauvegardes, les fichiers de traitements, et les fichiers terrains se trouveront dans le répertoire du projet.

Notre programme est facile à manipuler. Vérifier si AutoCAD 2008 est installé. Mais si ce n'est pas le cas, notre logiciel offre la possibilité de visualiser les dessins avec sa propre fenêtre de visualisation.

En l'absence de Traceur pour l'impression des dessins, nous pouvons exporter les visualisations en format BMP, ou les sauvegarder sous fichiers numériques.

Pour éviter des petites erreurs, il faut conserver les extensions de fichiers reconnues par le logiciel.

Le tableau suivant résume les étapes à effectuer avec les fichiers pris en charge par notre logiciel :

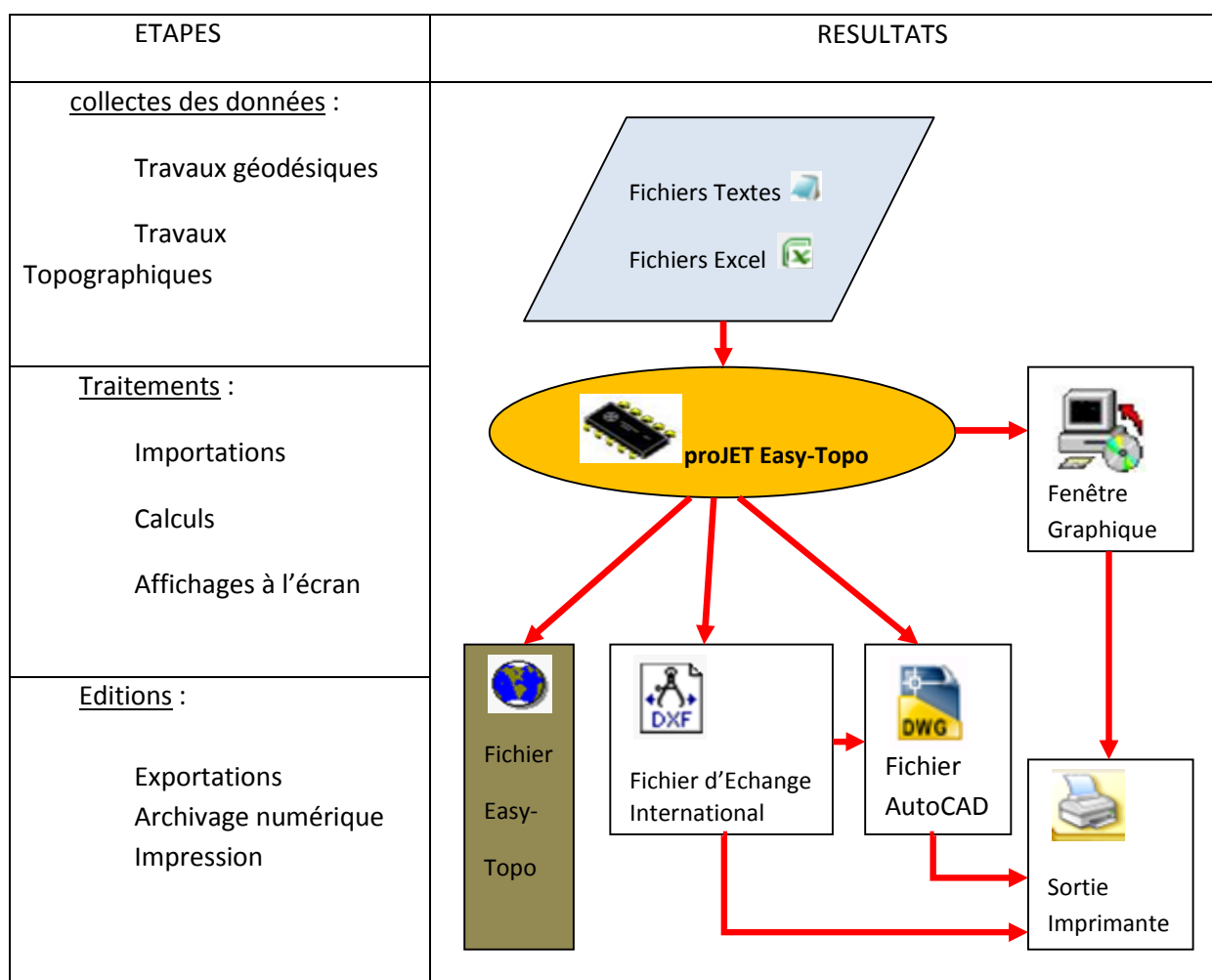


Tableau 2 : Mode d'emploi du Logiciel

1) APERÇUES DU LOGICIEL :

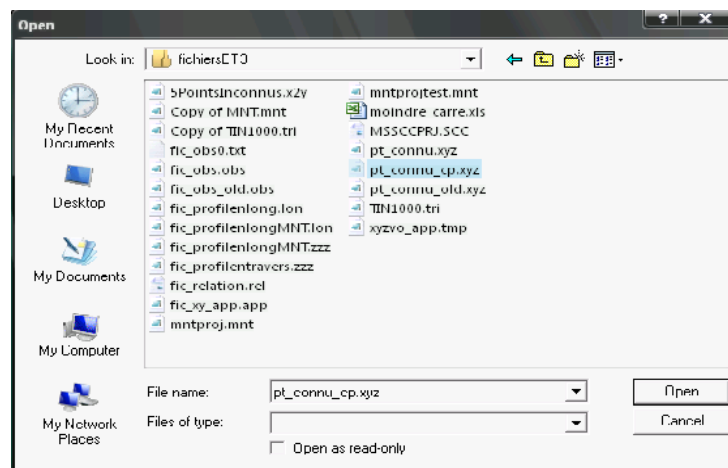
a) Icône et identité :



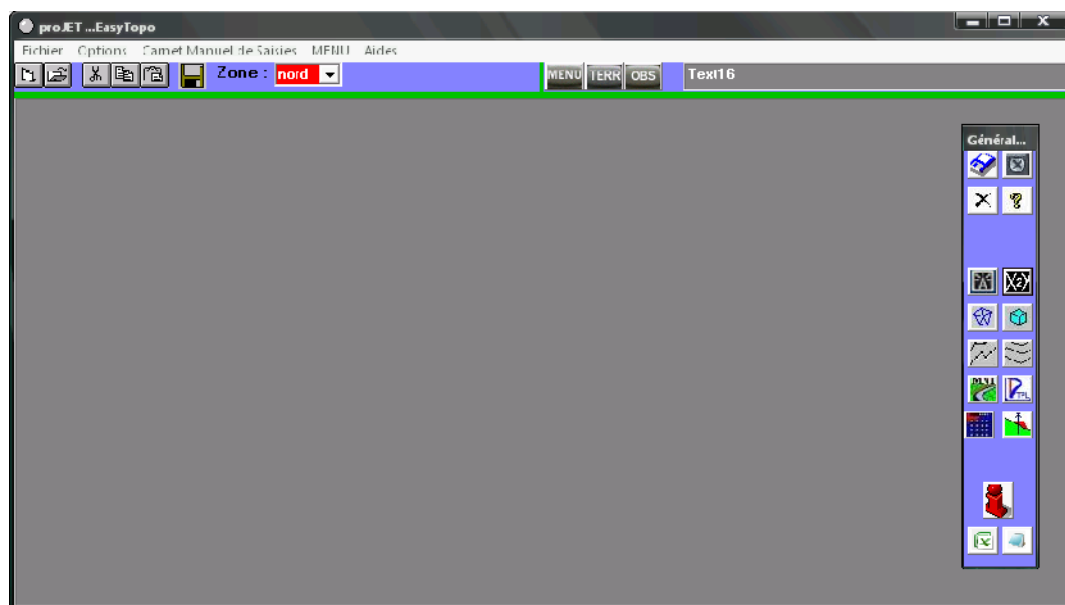
b) Fenêtre d'accueil :



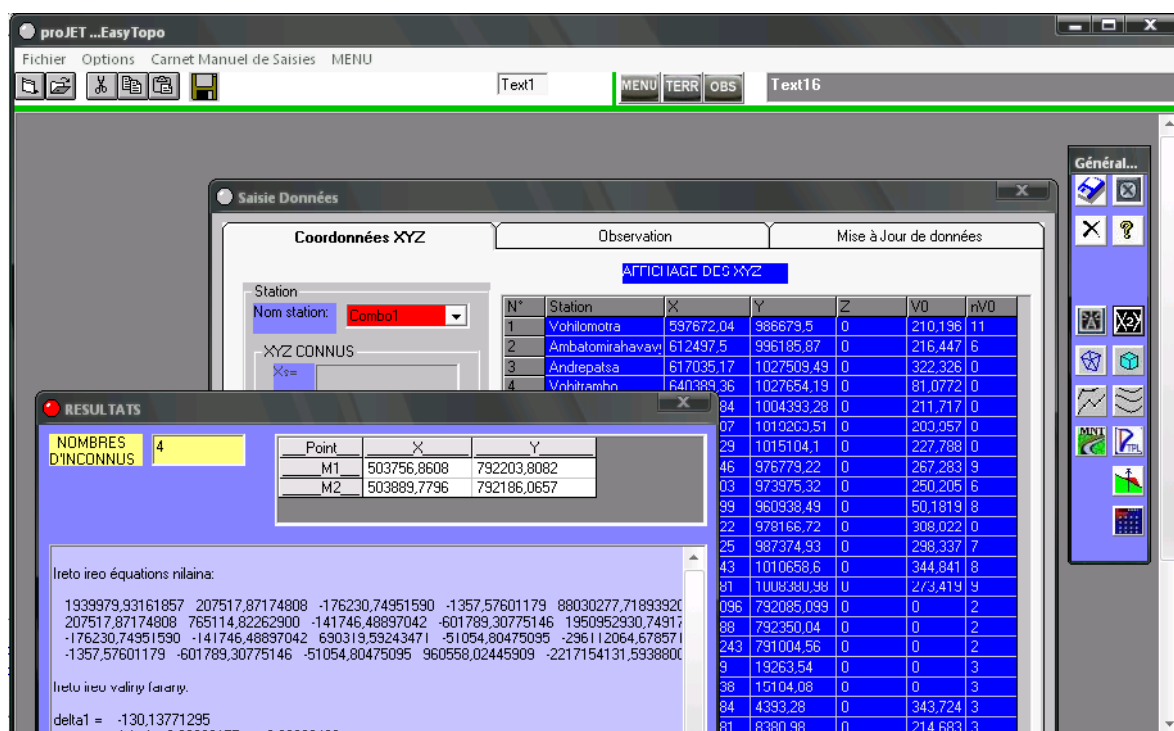
c) Fenêtre d'ouverture ou de sauvegarde de fichiers



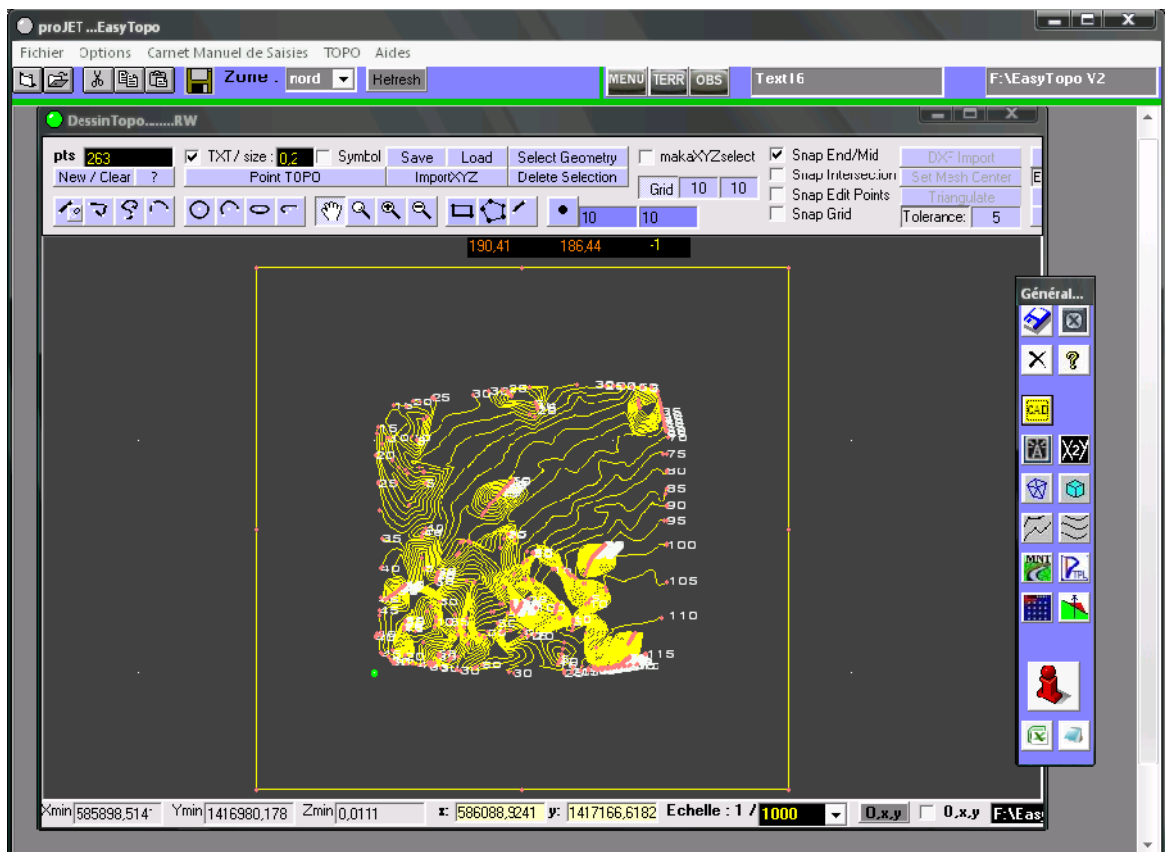
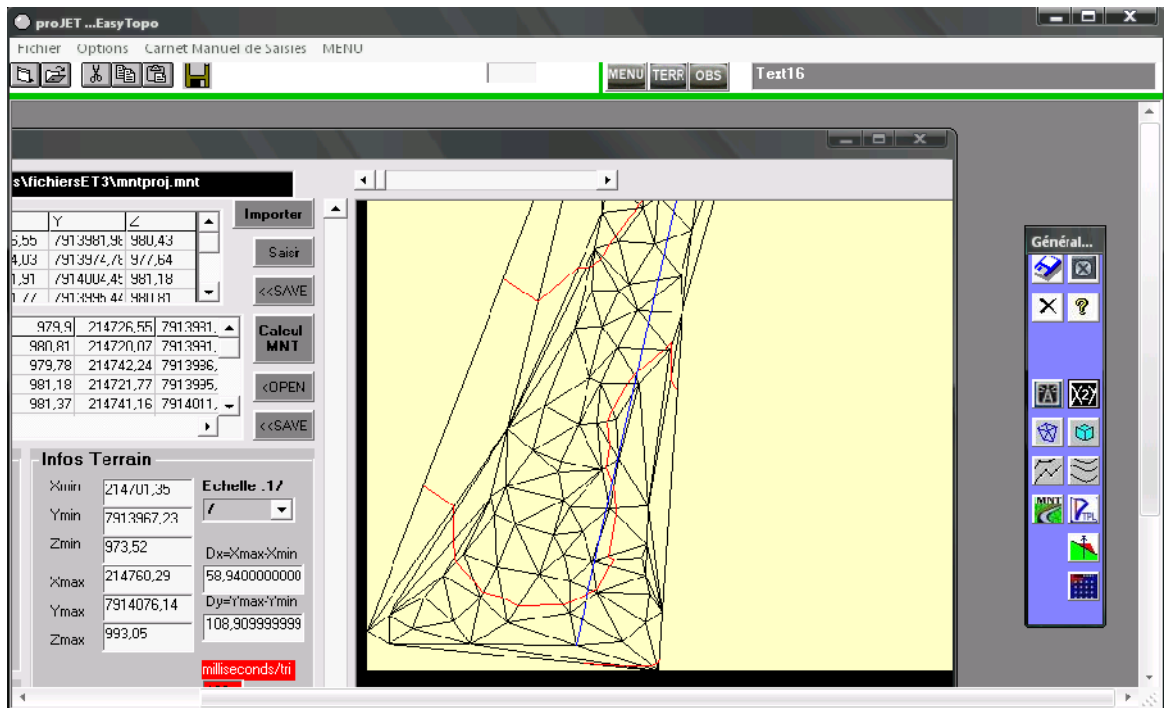
d) Fenêtre principale :



e) Fenêtres filles :



f) Fenêtre de visualisation des plans topographiques :



CONCLUSION

Les deux premières parties nous ont permis de focaliser l'étude sur le besoin d'application informatique, pour la topographie, vu les lourdes tâches de calcul et traitements manuels. Nous avons spécifié quels sont les différents modules de la topographie les plus utilisés dans les travaux : ceux sont la méthode de compensation par moindres carrés, en 2D, l'édition des profils topographiques et la création de MNT, en 3D. Nous avons ajouté l'exploitation du MNT dans le but d'avoir les courbes de niveaux, de tracer des profils, et de visualiser un rendu 3D du relief.

Dans la troisième partie nous avons développé le logiciel.

Le premier avantage de la programmation est la conformité du produit aux besoins de l'utilisateur, car c'est l'utilisateur même qui modélise le logiciel qu'il va employer. Pourtant une programmation s'avère très coûteux en matière de délai de développement du logiciel et en matière de financement du projet. Mais surtout il faut maîtriser les langages informatiques. Un des avantages de ce type de programmation, orientée objet, est la possibilité de compléter ou d'améliorer le programme ultérieurement ; par exemple pour l'année 2012 on va recompiler le logiciel avec une version d'AUTOCAD 2012.

L'étude serait plus complète si nous avions pu programmer tous les produits dérivés du MNT.

La dernière partie consiste à l'emploi du logiciel déjà compilé et installé sous Microsoft Windows. Dans cette partie nous avons expliqué le fonctionnement du logiciel. Mais on doit se contenter de ce qu'il nous offre : il a été créé avec Visual Basic, or Visual basic ne peut pas calculer une matrice d'équations supérieure à 10x10.

Ce mémoire devrait aider, inciter et guider tout secteur utilisateur de la topographie à appliquer l'informatique dans la réalisation de leurs travaux, avec automatisation.

Ce mémoire, en plus de son complément pour le logiciel « EASYTOPO version 1.0 », présente un début d'une base de conception et de modélisation 3D. Ce logiciel peut être complété et mis sur le marché. Cela dépend beaucoup de mes cadets dans le département Information Géographique et Foncière. Je laisse ici un travail qui peut être un sujet de mémoire ultérieur, et un produit purement IGF.

REFERENCES

BIBLIOGRAPHIE :

- [1] : RM Di Scala, « Les bases de l'informatique et de la programmation », Révision du 05 Septembre 2004.
- [2] : Rod Stephens, "Visual Basic 2005 Programmer's Reference", Edition Wiley
- [12]: MAMILALA Tolotra Nantenaina, « Elaboration d'un logiciel de calcul topographique : Easy-Topo Version 01 », Promotion 2009.

DOCUMENTS :

- [3] : NARY HERILALAO IARIVO, «Métrologie – Compensations Planimétriques »
- [4] : RABEHERIMANANA Lyliane, « Cours d'Informatique – Programmation », 3^{ème} Année
- [5] : Aide au Développeurs d'AutoCAD 2002-2008.
- [6] : Jean-Baptiste HENRY, « Cours de Topographie et Topométrie Générale » Université Luis Pasteur Strasbourg, Chapitre II.

WEBOGRAPHIE :

- [7] : [Http: //www.geomatique.georezo.net/](http://www.geomatique.georezo.net/) : "Exploitation du MNT"
- [8] : [Http: //www.developpez.net/](http://www.developpez.net/) : "Algorithmes de Delaunay"
- [9] : [Http: //fr.wikipedia.org/](http://fr.wikipedia.org/)Définitions : "MNT-Courbes de niveaux-Rendu 3D"
- [10] : [Http: //www.cours-genie-civil.com/](http://www.cours-genie-civil.com/) : "LES PROJETS ROUTIERS"
- [11] : [Http://raphaello.univ-fcomte.fr/](http://raphaello.univ-fcomte.fr/) et mathcurves.com/ : « Lissage de lignes polygonales et Lissage de Surfaces »

ANNEXES

ANNEXE 1 : Extrait de codes de notre programme

```

Option Explicit

'Manaboatra Points (Vertices)
Public Type dVertex
    x As Double
    y As Double
    z As Double
End Type

'Manamboatra Triangles, vv# ireo 3 sommets triangle
Public Type dTriangle
    vv0 As Double
    vv1 As Double
    vv2 As Double
End Type

'afaka reglana
Public Const MaxVertices = 30000
Public Const MaxTriangles = 100000 'inty ny farany zaka 2 147 483 647 type long

'ireo point
Public Vertex() As dVertex

'ireo triangle
Public Triangle() As dTriangle

'*****anaty cercle*****

    Dim Eps As Double
    Dim m1 As Double
    Dim m2 As Double
    Dim mx1 As Double
    Dim mx2 As Double
    Dim my1 As Double
    Dim my2 As Double
    '    Dim dx As Double
    '    Dim dy As Double
    Dim rsqr As Double

```

'Déclaration des variables et création des objets



```





Private Sub optCommand_Click(Index As Integer)
Redraw
StartCommand Index
End Sub

'*****OUTIL DESSIN TOPO*****
Sub StartCommand(Index As Integer)
Stage = 0
picDraw.MousePointer = 2
Select Case Index
Case 0: Command = "Line": StageMax = 1
Case 1: Command = "PolyLine": StageMax = 1
Case 2: Command = "Spline": StageMax = 1
Case 3: Command = "3 Point Arc": StageMax = 2
Case 4: Command = "Circle": StageMax = 1
Case 5: Command = "Arc": StageMax = 3
Case 6: Command = "Closed Ellipse": StageMax = 2
Case 7: Command = "Ellipse": StageMax = 4
Case 8: Command = "Box Select": StageMax = 1
Case 9
Command = ""
DeleteSelection Selection(), DrawGeo(0)
Redraw
Erase Selection()
optCommand(8).Value = True
Case 10: Command = "DCenter": StageMax = 0
Case 11
picDraw.MousePointer = 99
Command = "Pan"
StageMax = 0
Case 12: Command = "Zoom": StageMax = 0
Case 13: Command = "Quadri": StageMax = 1
Case 14: Command = "Pointopo": StageMax = 0
Case 15: Command = "Polygon": StageMax = 1
End Select
GetSnap '*****acrochage objet*****
End Sub


```






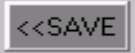
ANNEXE 2 : Aide globale à Easy Topo Version 02 :**Comment faire pour ... ?****Comment importer des fichiers terrains ?**

{ Icône  ou 
Menu « Fichier » > Importer

1. Cliquer sur  ou  ou Menu « Fichier » > Importer
2. Spécifier les options concernant l'importation
 - Séparateurs
 - Décimale (virgule ou point)
 - Entête du fichier
 - Nom du fichier de sortie : (ex : test.mnt ; test.xyz...)
3. Cliquer sur 
4. Spécifier l'Identité de chaque colonne : ex : colonne n°1→X, n°2→Y
5. Terminer en cliquant sur 

Comment calculer le MNT ou autres traitements ?

{ Icône 
Menu « TOPO » > Générateur de MNT > Calcul

1. Cliquer sur  ou Menu « TOPO » > Générateur de MNT > Calcul
2. Cliquer sur  (Choisir le répertoire du fichier)
3. Spécifier les options concernant les traitements :
 - ☐ Tracés seulement
 - Cases à cocher : ☒ Afficher Points
 - Option de choix obligatoire 
 - Echelle du dessin
 - Attributs du Calque MNT : couleur, hauteur de texte,...
4. Cliquer sur  ou CALCULER ou ENTER ou VALIDER / puis 
5. Vous pouvez cliquer sur  pour sauvegarder les résultats sous un autre répertoire.

Nom et prénoms : RASAMOELY Aina Natolotra Jonathan

Adresse : Lot II G 104 P Antsobolo Ambatomaro - Antananarivo 101

Contacts : +26133 05 065 84

E-mail : yonathanstc@yahoo.fr

Titre de mémoire :

**ELABORATION D'UN LOGICIEL DE CALCUL ET DE DESSIN
TOPOGRAPHIQUES**

Nombres de pages : 76

Nombres de tableaux : 02

Nombres de figures : 16

Résumé :

Cet ouvrage de mémoire présente l'élaboration d'un logiciel de calcul et de dessin topographique, un logiciel permettant de personnaliser et d'automatiser le traitement en bureau des différents travaux pratiques entrepris dans le département IGF à citer les calculs topométriques et les dessins topographiques.

Notre étude se focalise sur la programmation de ce logiciel à travers des étapes de modélisation, de conception et de codage dans l'environnement de développement du langage Visual basic.

Le logiciel manipule des fichiers d'observations, des fichiers bases de données de points connus pour donner des résultats et des visualisations par différents algorithmes de calculs. Ainsi, nous avons produit un logiciel éloigné des fautes de calcul tout en réduisant le temps des traitements et des éditions.

Summary :

This work presents the development of a topographical drawing and computation software. It is created for customization and automation in the treatment of the various practical works undertaken in IGF department, such as the topometric calculations and the topographical drawings.

Our study is focused on the programming of this software through stages of modelling, design and coding in the environment development of the Visual BASIC language.

The software handles files of observations, files database of points known to give results and visualizations by various calculation algorithms. Thus, we produced software far away from the analogue calculations error and reducing the time of the treatments and the time of editions.

Encadreurs :

Mr. RABETSIAHINY

Maître de conférences à l'Ecole Supérieure Polytechnique d'Antananarivo.

Mr. RABEMALAZAMANANA

Enseignant à l'Ecole Supérieure Polytechnique d'Antananarivo.

Mots Clés :

Calculs Topographiques, Dessins 3D, Fichiers à importer, diagramme UML, algorithme, organigramme, Visual Basic.