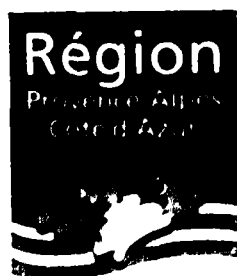




Ministère de l'Industrie,
des Postes et Télécommunications
et du Commerce extérieur



document public

**conception assistée par ordinateur
modélisation tridimensionnelle du sous-sol
logiciel GEOSURF pour WINDOWS**

*programme technologie-recherche
délibération n° 92/00395*

A. Bourrousse
avec la collaboration de K. Khazal et C. Martin

**septembre 1994
Rapport du BRGM R 38098**

numéro de référence P 9304232026



Etude réalisée dans le cadre des
actions de Service public du BRGM

1993 - H - 104

**BRGM
SERVICE GEOLOGIQUE NATIONAL
Département utilisation et protection
de l'espace géologique
Groupe Risques naturels et Géoprospective
BP 167 - 13276 MARSEILLE CEDEX 09 - FRANCE - Tél.: (33) 91 17 74 74**

RESUME

RAPPELS DU PROJET

Le programme de modélisation tridimensionnelle du sous-sol lancé en 1988 dans le cadre d'une convention de recherche entre le Conseil Régional PACA et le BRGM, et poursuivi en 1992 grâce à la convention de recherche n° 92/00395, entre le Conseil Régional PACA et le BRGM (MSP fiche n° 93H104), a été rendu possible grâce à l'évolution des techniques et des matériels informatiques.

Ce type de modélisation présente une série d'avantages au niveau de la recherche fondamentale et des applications pratiques qui en découlent :

- *Au niveau fondamental :*

La modélisation permet de représenter de manière complète, les différentes hypothèses du chercheur et d'en vérifier la cohérence;

- *Au niveau appliqué :*

1. *Prospection* : le modèle permet la recherche systématique des configurations de terrain propices;
2. *Modélisation des gisements* : optimisation de la gestion de l'extraction grâce à l'estimation des différents volumes de terrains qui seront extraits;
3. *Implantation d'ouvrages souterrains* : le modèle permet d'évaluer la nature et l'extension des terrains rencontrés par l'ouvrage. Il permet donc, par exemple, d'optimiser le choix d'un tracé pour un tunnel en fonction de critères techniques et économiques ;
4. *Evaluation des aléas naturels (mouvements de terrain, microzonage sismique)* : la structure du sous-sol est un des facteurs déterminants dans l'étude des mouvements de terrain ou dans l'évaluation des effets de site, et seule l'utilisation d'un modèle 3D permet d'intégrer de manière complète la géométrie des différentes couches.

RESULTATS OBTENUS

La réalisation du programme de recherche a abouti à la création du logiciel GEOSURF qui permet de gérer, de manipuler et d'exploiter des modèles numériques de terrain.

GEOSURF a pour objet la modélisation tridimensionnelle du sous-sol. Il manipule la topographie du site mais aussi les différentes formations sous-jacentes. Ce logiciel est notamment destiné à la modélisation de carrières et de remblais (décharges), sur des zones allant de quelques centaines de mètres carrés à plusieurs kilomètres carrés. Il permet de simuler des scénarios d'exploitation en donnant le contour d'exploitation, la pente des talus et la cote finale de l'ouvrage.

Conception assistée par ordinateur - Modélisation tridimensionnelle du sous-sol
Logiciel GEOSURF pour WINDOWS

GEOSURF permet en outre :

- les calculs de cubatures,
- la génération de coupes/profils,
- le dessin de plans topographiques,
- le dessin de blocs diagrammes 3D orientables,
- la visualisation et l'appréciation de l'impact visuel des ouvrages modélisés,
- l'exportation des résultats vers les plus grands standards de l'infographie informatique.

GEOSURF fonctionne sur micro-ordinateur avec un environnement WINDOWS. Il est destiné à des utilisateurs qui ne sont pas informaticiens. Il a été conçu pour être facilement utilisable par tous.

TABLE DES MATIERES

RESUME

1. INFORMATIONS GENERALES.....	1
1.1. Rappels du projet	1
1.2. Objectifs du programme de recherche 1992.....	1
1.3. Configuration matérielle et logicielle nécessaire	2
1.4. Organisation du manuel et de ses sections.....	2
1.5. Démarrage du programme	2
2. LES AMELIORATIONS INFORMATIQUES DU LOGICIEL	3
2.1. Les améliorations des capacités de traitement.....	3
2.2. Choix du langage de programmation.....	3
2.3. Communication avec les autres logiciels du marché.	4
3. RAPPELS SUR LE MODELE GEOLOGIQUE 3D	6
3.1. Le modèle numérique topographique	6
3.2. Le modèle numérique géologique	6
3.3. Intégration des données brutes.....	7
3.3.1. Données provenant d'un fichier MNT IGN	7
3.3.2. Données provenant d'un fichier DXF AUTOCAD	7
3.3.3. A Partir d'une digitalisation.....	7
3.3.3.1. Digitalisation avec GEOSURF.....	7
3.3.3.2. Digitalisation avec MICROSTATION.....	7
3.4. Acquisition des données géologiques	8
3.4.1. Les profils	8
3.4.2. Habillage des profils.....	8
3.4.3. Transcription sur une carte topographique.....	9

3.5. Construction de la grille: Interpolation des données	9
3.5.1. Méthodes d'interpolation disponibles	9
3.5.2. Inverse des distances	10
3.5.3. Krigeage.....	10
3.5.4. Courbure minimale.....	11
3.5.5. Avantages, inconvénients des méthodes	11
3.6. Vérification qualitative du MNT géologique	11
4. ENSEMBLE DES MODELISATIONS DISPONIBLES	13
4.1. Digitalisation du contour de modélisation	14
4.2. Les différents types d'ouvrages modélisables.....	14
4.2.1. Les fosses.....	15
4.2.1.1. <i>Le contour digitalisé est la limite supérieure d'une fosse</i>	15
4.2.1.2. <i>Le contour digitalisé est la limite inférieure d'une fosse</i>	17
4.2.2. Les remblais	17
4.2.2.1. <i>Le contour digitalisé est la limite supérieure d'un remblai</i>	17
4.2.2.2. <i>Le contour digitalisé est la limite inférieure d'un remblai</i>	18
4.3. Outils complémentaires de modélisation.....	18
4.3.1. Modification partielle d'une grille.....	18
4.3.2. Carte d'affleurement	19
4.3.3. Calculs des cubatures déplacées.....	19
4.3.4. Carte de visualisation (impact visuel).....	19
4.3.4.1. <i>Principe</i>	19
4.3.4.2. <i>Visualisation</i>	20
4.3.5. Densité de visualisation	20
4.3.6. Superposer plusieurs modélisations sur une même grille	20
4.3.7. Affectation d'une cote Z à l'ensemble d'une zone donnée d'une grille.....	21
4.3.8. Modélisation des gradins.....	21
4.3.9. Remonter la cote du fond d'une fosse.....	22
4.3.10. Réduire le nombre de mailles	22
5. VISUALISATION GRAPHIQUE DES RESULTATS (SURFER).....	24
5.1. Contexte d'utilisation.....	24
5.2. Vues en deux dimensions (2D)	24
5.3. Vues en trois dimensions (3D)	25
6. CONCLUSIONS SUR LES RESULTATS OBTENUS.....	29

TABLES DES FIGURES

Figure 1 - Modélisation d'une carrière. Ombrage de type PHONG	5
Figure 2 - Modélisation d'une carrière de roches massives. Vue globale et détails du maillage ...	16
Figure 3 - Modélisation d'une carrière. Différentes étapes de creusement	23
Figure 4 - Sorties graphiques en 2D réalisées avec GEOSURF	26
Figure 5 - Sorties graphiques en 3D réalisées avec GEOSURF	27
Figure 6 - Combinaison 2D et 3D réalisée avec SURFER	28

TABLE DES ANNEXES

Annexe 1 - Les modules de GEOSURF dans leur environnement windows	
Annexe 2 - Comparaison du codage VISUAL BASIC avec le BASIC PDS 7	
Annexe 3 - Procédure de saisie en 3D sous microstation	
Annexe 4 - GEODISC	

GEOSURF pour WINDOWS

VERSION 2.0w - Septembre 1994 (c) BRGM

**Microsoft, MS-DOS, Windows, sont des marques déposées de Microsoft Corporation
Intergraph, Microstation, sont des marques déposées d'Intergraph Corporation / Bentley Systems
Surfer, MapViewer sont des marques déposées de Golden Software Inc.**

1. INFORMATIONS GENERALES

1.1. RAPPELS DU PROJET

Les technologies informatiques qu'elles soient matérielles ou logicielles, permettent la modélisation en trois dimensions (3D) des différents terrains constituant le sous-sol d'une région. Une première étude conduite au BRGM en 1988 avait montré la faisabilité de tels projets. La région PACA ayant par la suite attribué (1988-1989) des crédits de recherche au BRGM, a permis l'élaboration de l'outil informatique GEOSURF. Ce logiciel permet notamment de saisir les horizons géologiques, de contrôler les hypothèses du chercheur, de réaliser des cartes topographiques 2D, des vues 3D largement paramétrables, des coupes (profils)... L'intérêt majeur de GEOSURF consiste en la modélisation de structures de type carrières et remblais, avec le calcul des volumes correspondant aux différentes classes de terrains rencontrés.

Le logiciel GEOSURF a été primé (cf. annexe 4) au concours GEODISC 91/92 de l'IGN (Institut Géographique National). Après avoir été évalué par un comité de sélection, il a obtenu un prix accompagné du "LABEL IGN".

1.2. OBJECTIFS DU PROGRAMME DE RECHERCHE 1992

GEOSURF a pour objet la modélisation tridimensionnelle du sous-sol. Il manipule la topographie du site mais aussi les différentes formations géologiques sous-jacentes. Ce logiciel est destiné à la modélisation d'objets tels que des carrières ou des remblais, sur des zones allant de quelques centaines de mètres à plusieurs kilomètres carrés. Pour les applications de type carrière ou décharge, il permet de simuler des scénarios d'exploitation en donnant le contour de l'exploitation, la pente des talus et la cote finale (Z) de l'ouvrage.

GEOSURF permet en outre :

- les calculs de cubatures (totales ou par classes de terrain),
- la génération de profils intégrant les ouvrages modélisés avec leur géologie,
- le dessin de plans topographiques 2D,
- le dessin de blocs diagrammes orientables 3D,
- la visualisation et l'appréciation de l'impact visuel des ouvrages modélisés.

GEOSURF dans sa version initiale fonctionne sur micro-ordinateur dans un environnement MS-DOS. Cet environnement présente un certain nombre de contraintes dont :

- une ergonomie d'utilisation peu développée (système non graphique),
- une taille limite des données mise en mémoire (barrière des 640 Ko du DOS).

Le projet d'amélioration du programme proposé en 1992, a porté essentiellement sur ces deux points. Pour cela, il a été décidé :

- de porter le programme sous l'environnement WINDOWS (Microsoft),
- d'utiliser un langage de programmation adapté à cet environnement.

Le choix du langage s'est porté sur le VISUAL BASIC (de Microsoft également) afin de garantir une parfaite synergie entre le langage de programmation et l'environnement système du produit.

1.3. CONFIGURATION MATERIELLE ET LOGICIELLE NECESSAIRE

Le programme GEOSURF pour WINDOWS fonctionne sur tout micro-ordinateur possédant au moins un microprocesseur i80386. L'adjonction d'un coprocesseur mathématique n'est pas obligatoire mais vivement recommandée. Une carte graphique VGA ou compatible est nécessaire ainsi qu'une tablette à numériser pour l'acquisition des données. Le système d'exploitation de la machine doit être MS-DOS (version 5 ou 6) et l'environnement WINDOWS 3.1x doit être installé. La mémoire étendue est obligatoire avec un minimum de 4 Mo. Toute la mémoire vive (RAM) disponible du micro-ordinateur sera exploitée.

Pour un fonctionnement optimal, GEOSURF doit être accompagné des logiciels SURFER pour Windows et MICROSTATION d'Intergraph.

1.4. ORGANISATION DU MANUEL ET DE SES SECTIONS

Cette notice est scindée en 6 chapitres qui décrivent :

- les généralités d'utilisation,
- les améliorations informatiques du logiciel,
- des rappels sur la construction d'un modèle géologique 3D,
- l'ensemble des modélisations disponibles,
- l'utilisation du logiciel SURFER avec la visualisation graphique des résultats,
- conclusion-annexes.

1.5. DEMARRAGE DU PROGRAMME

Pour utiliser le logiciel GEOSURF, il convient tout d'abord de l'installer grâce à la procédure spécifique (INSTALL) présente sur la disquette d'installation. Au lancement de Windows, un groupe de programme apparaît ("GEOSURF pour Windows") avec la liste des différents modules disponibles. Chaque icône (voir planche 1 de l'annexe 1) correspond à un module de GEOSURF et pour en activer un il suffit d'un "double-clic" sur l'icône. Les programmes répondent aux normes Windows et sont donc facilement utilisables par tous (interface graphique manipulée à l'aide de la souris).

2. LES AMELIORATIONS INFORMATIQUES DU LOGICIEL

2.1. LES AMELIORATIONS DES CAPACITES DE TRAITEMENT.

Dans sa version précédente, GEOSURF était limité par les contraintes du système d'exploitation MS-DOS (barrière des 640 ko de mémoire vive). La faible quantité de mémoire adressable par le programme n'autorisait pas la manipulation de données importantes. Une grille de données était limitée à environ 200 mailles par 200 mailles soit un total de 40 000 mailles par interface. Cette limite correspondait à des tailles de mailles d'environ 5 m sur 5 m pour la plupart des cas traités; la précision sur les calculs de volumes s'en trouvait d'autant affectée avec une précision assez médiocre lors des calculs sur les digues de décharge par exemple.

La version réalisée dans le cadre du projet de recherche 1992 a eu pour but de s'affranchir de façon définitive de la limite des 640 ko du DOS. La reprogrammation en Visual Basic de tous les modules de GEOSURF a permis d'atteindre cet objectif. Le produit obtenu permet d'intégrer des grilles de données de taille quasi illimitée avec pour seule limite la quantité de mémoire installée sur le micro-ordinateur par le constructeur, soit généralement 8 Méga octets.

Les possibilités offertes par le logiciel sont de ce fait considérables. Le nombre de mailles pouvant être choisi sans aucune contrainte extérieure, les calculs sont de ce fait très précis avec des mailles pouvant atteindre 50 cm de côté. La prise en compte de structures géologiques plus complexes est facilitée et tous les calculs qui nécessitaient une résolution plus fine sont à présent accessibles et performants (modélisations de gradins d'exploitation, de rampes d'accès, de fossés d'écoulements, de digues...)

2.2. CHOIX DU LANGAGE DE PROGRAMMATION.

Lors de la phase de codage du programme, nous avons dû choisir un langage qui nous permettrait d'atteindre les objectifs fixés dans le cahier des charges. Compte tenu de l'environnement Windows sous lequel nous voulions migrer, peu de langages était présents sur le marché. Le choix d'un langage orienté objet était inévitable et notre choix pouvait se porter soit vers le Visual Basic de Microsoft soit vers le langage C de Borland.

La précédente version de GEOSURF ayant été programmée en BASIC 7 de Microsoft une migration vers le Visual Basic était plus naturelle; le recodage des modules fut donc décidé.

Au niveau du recodage, les entrées/sorties et l'interface ont dû être complètement reprogrammées. Les algorithmes de calculs n'ont pas beaucoup variés mais ont tout de même été repris.

Par rapport au Basic 7, le Visual Basic présente une conception très différente puisqu'il n'est plus un langage linéaire mais un langage orienté objet. Ces objets sont reliés à des procédures qui réagissent en fonction des actions de l'utilisateur (manipulation de "boutons" à l'aide de la souris). Un programme peut enchaîner jusqu'à 230 écrans par projet avec un maximum de 5200 procédures et modules.

L'annexe 2 montre le code d'un même programme écrit en Basic 7 puis en Visual Basic. Les différences sont énormes et les seules similitudes se retrouvent au niveau des algorithmes de codage.

Le choix d'une solution complète Microsoft (Visual Basic et Windows) nous assure une pérennité des programmes avec l'évolution de Windows, fortement prévisible dans les prochaines années.

2.3. COMMUNICATION AVEC LES AUTRES LOGICIELS DU MARCHE.

La communication de nos programmes avec les grands standards du marché était devenue une nécessité. Aussi une passerelle a été prévue avec les outils de Conception Assistée par Ordinateur (CAO). Un module de communication de GEOSURF vers MICROSTATION a été développé; ce programme (GRD2US4) effectue un transfert des Modèles Numériques de Terrain (MNT) vers le logiciel de CAO. A partir de là, tous les transferts vers les logiciels acceptant la norme DXF (AUTOCAD) sont possibles.

Une des toutes premières applications qui a pu être faite consiste en la réalisation d'image de synthèse (création d'image) pour rendre plus réaliste un paysage modélisé par GEOSURF. La figure 1 montre un ombrage de type Phong; dans ce modèle ombré, la couleur de chaque pixel est recalculée. L'ombrage Phong permet de produire des images de haute qualité quand la vitesse n'est pas critique (performances de la machine) et quand on désire obtenir des effets d'éclairage très exacts. Si les sources lumineuses sont proches des objets rendus, tel qu'un projecteur braqué sur une maison, l'ombrage Phong est la seule technique qui permet de visualiser l'emplacement exact du rayon de lumière.

La communication du logiciel GEOSURF ne se limite pas à des échanges avec les logiciels de CAO. En effet, le fait que le programme évolue sous un environnement Windows, autorise la communication directe via le presse-papiers (buffer) avec tout logiciel de ce même environnement (Word, Excel, Corel Draw...).

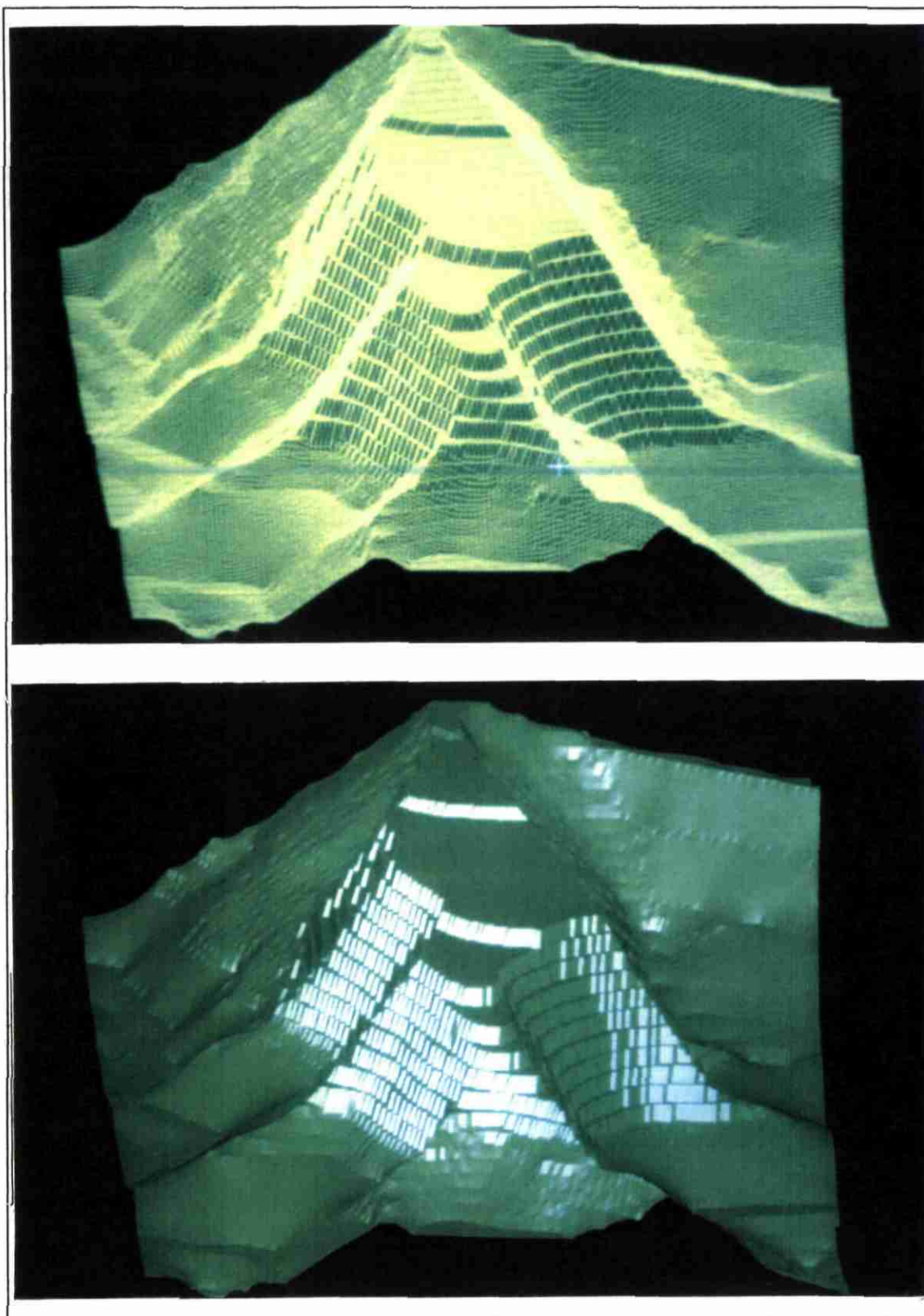


Figure 1 - Modélisation d'une carrière
Ombfrage de type PHONG
(Photographies d'écran)

3. RAPPELS SUR LE MODELE GEOLOGIQUE 3D

DEFINITION: Un modèle géologique possède une structure définie en tout point. Il peut ainsi satisfaire à la modélisation de la plupart des structures géologiques rencontrées.

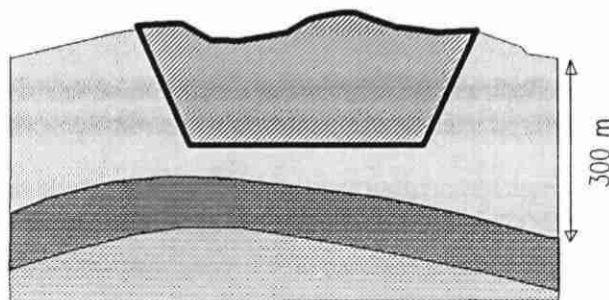
3.1. LE MODELE NUMERIQUE TOPOGRAPHIQUE

Le modèle numérique de terrain (MNT) topographique représente la surface altimétrique du sol. Chaque point est identifié par ses coordonnées X et Y exprimées en kilomètres et par son Z exprimé en mètres.

Il correspond à une matrice de points cotés en XYZ. Cette matrice est souvent carrée (mais pas obligatoirement) et possède un nombre important de points identifiés de façon numérique (mailles).

3.2. LE MODELE NUMERIQUE GEOLOGIQUE

Le cas le plus simple est constitué par la seule surface topographique. Il correspond au cas où la première couche géologique affleurante est suffisamment épaisse pour supporter toutes les modélisations.



Dans ce cas, il est inutile de générer les autres interfaces car ils n'interviendront pas dans le calcul des cubatures.

Pour des cas plus complexes, composés de plusieurs couches géologiques, seules les interfaces séparant deux formations sont représentées.

Ces différentes interfaces sont entièrement indépendantes les unes des autres et leur création aboutit à autant de matrices que d'interfaces géologiques. Chaque matrice d'interface présente la même structure que celle qui modélise la topographie et chaque point d'interface est représenté par un triplet (X,Y,Z).

On imagine facilement la quantité de données manipulées par un modèle; ainsi pour un modèle comportant cinq couches différentes, quatre interfaces géologiques seront nécessaires plus le niveau topographique supérieur soit 40 000 X 5 ce qui équivaut à 200 000 points cotés soit environ 800 Ko (une grille de 200 x 200 a une taille d'à peu près 160Ko).

L'importance des mémoires de masse des machines utilisées et l'optimisation des traitements sont primordiaux pour obtenir des temps de réponse corrects.

3.3. INTEGRATION DES DONNEES BRUTES

Que ce soit pour la fabrication de la matrice (ou grille) "surface topographique" ou pour les matrices constituant les interfaces géologiques, le principe est le même. On part d'un nombre N de points, dits de contrôle, qui sont ensuite interpolés pour nourrir les mailles du modèle.

3.3.1. Données provenant d'un fichier MNT IGN

Ces fichiers de données sont disponibles auprès de l'IGN. Ils sont déjà maillés et sont livrés sous forme ASCII. Le module dédié à cette intégration permet de prendre en compte la totalité du fichier ou bien d'en extraire une partie qui sera par la suite interpolée et mise au format GRD de GEOSURF. Le format du fichier généré autorise une interpolation avec SURFER ou directement avec GEOSURF.

NOTA: Le format GRD est le format des grilles utilisées par GEOSURF et SURFER.

3.3.2. Données provenant d'un fichier DXF AUTOCAD

Les fichiers Autocad DXF sont des fichiers ASCII. Ce format très largement répandu est généralement fourni lors des relevés effectués par les géomètres ou suite à une photo-restitution d'un site. Le module qui en effectue le décodage extrait les points (X,Y,Z) qui constituent les courbes de niveaux et les points cotés. Une interpolation de l'ensemble de ces points fournit la grille GRD.

3.3.3. A Partir d'une digitalisation

Très souvent, l'utilisateur sera amené à digitaliser lui-même ses données pour restituer la surface topographique ou les interfaces géologiques.

3.3.3.1. Digitalisation avec GEOSURF

LINEAS est un module de GEOSURF qui permet de digitaliser les données, le document de base étant une carte topographique. Il est ainsi possible de saisir des courbes avec identifiant (Z) ou des points cotés. Les données ainsi saisies seront stockées dans un fichier de type ASCII.

3.3.3.2. Digitalisation avec MICROSTATION

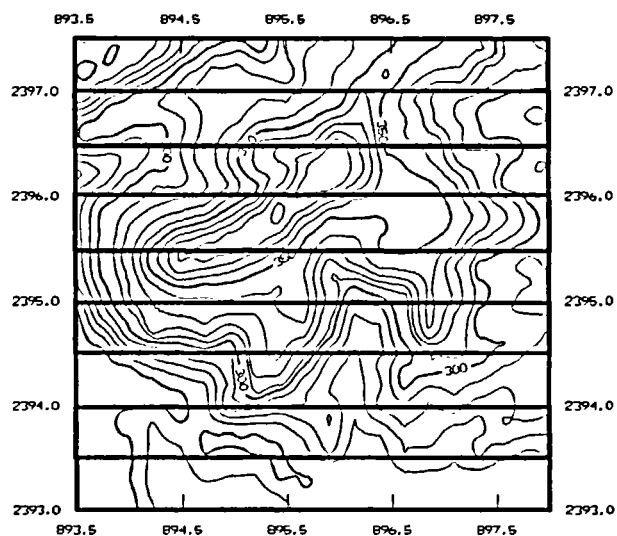
MICROSTATION permet d'effectuer la digitalisation en 3D des courbes de niveau. Une procédure de numérisation a été écrite afin d'en faciliter la mise en oeuvre; elle est fournie en annexe 3. Les données sont stockées dans un fichier binaire décodé lors de l'interpolation du MNT par la procédure DGNGRILL.

3.4. ACQUISITION DES DONNEES GEOLOGIQUES

La méthode présentée permet de reconstituer cartographiquement les courbes de niveau du toit de chaque couche, à partir d'une carte géologique. Ces couches seront ensuite digitalisées avec LINEAS puis transformées en grille GEOSURF.

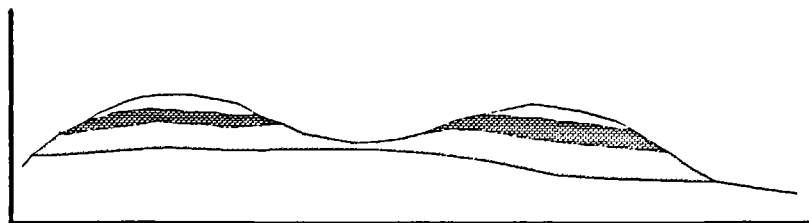
3.4.1. Les profils

La méthode est basée sur la réalisation d'une série de profils (ou coupes) qui vont recouper l'ensemble de la zone considérée. Généralement une dizaine de profils suffisent pour obtenir une information fiable. Ces profils peuvent être générés automatiquement par GEOSURF.

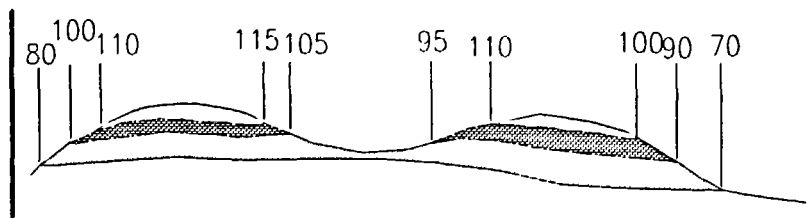


3.4.2. Habillage des profils

L'opérateur habillera les différents profils en y reportant les couches géologiques:

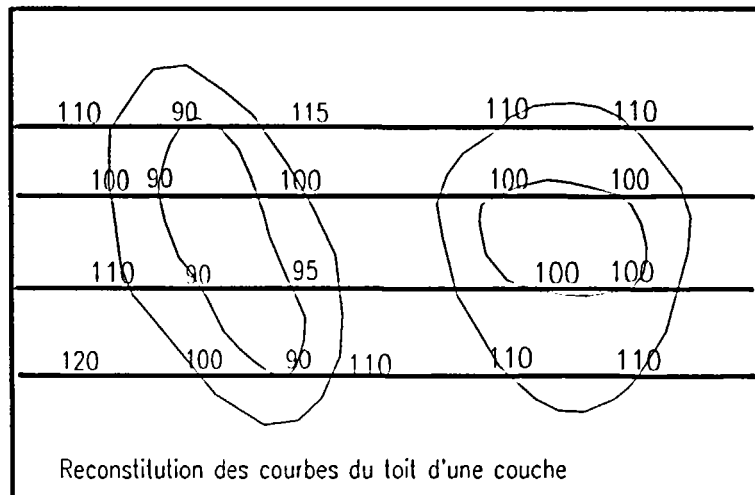


Pour chaque limite de couche, il repérera un maximum de points d'iso-altitude.



3.4.3. Transcription sur une carte topographique

A l'emplacement où ont été réalisés les profils, on reporte les points connus, puis on trace les courbes reliant les points d'une même altitude. On obtient alors, une carte classique que l'on digitalisera avec LINEAS et que l'on interpolera pour aboutir à une grille GRD.



3.5. CONSTRUCTION DE LA GRILLE: INTERPOLATION DES DONNEES

L'interpolation est un procédé qui à partir d'un semis de points répartis de façon discontinue réalise une série maillée continue. Le calcul d'une valeur se fait à partir des mesures disponibles en des points voisins.

Plus le nombre de points de contrôle est grand, plus la qualité du MNT interpolé sera meilleure. Il est donc essentiel d'avoir de bonnes données de base.

3.5.1. Méthodes d'interpolation disponibles

Pour interpoler les données, deux voies sont disponibles; trois algorithmes d'interpolation sont proposés:

3.5.2. Inverse des distances

Cette méthode est assortie d'une puissance de pondération qui par défaut est $j=2$.

L'équation utilisée est :

$$Z = \frac{\sum_{i=1}^n Z_i / (d_i)^j}{\sum_{i=1}^n 1 / (d_i)^j}$$

avec:

- Z_i est un point voisin
- d la distance
- n le nombre de points utilisés (lié à la méthode de recherche des points)
- j le poids de la pondération

Les points qui sont utilisés par cette méthode sont pondérés afin que l'influence d'un point éloigné soit plus faible que celle d'un point plus proche donc plus représentatif.

Plus la puissance de pondération est grande, plus rapide est la baisse d'influence des points utilisés lors du calcul. Le poids de cette pondération peut varier de 1 à 10 .

3.5.3. Krigeage

Utilisation de la théorie de la variation régionale. Cette méthode donne de très bons résultats pour l'interpolation de données altimétriques. Elle est basée sur les constatations suivantes:

- la variabilité d'un échantillon d'altitudes distribuées dans l'espace et représentant des structures d'ordre géologiques n'est pas complètement aléatoire ni complètement dissymétrique. Lors du calcul d'un modèle continu,

cela signifie que la valeur Z pour un point inconnu, peut être estimée à partir des points voisins, et que le point inconnu est nécessairement entièrement dépendant des valeurs des autres points connus.

- la variabilité des altitudes Z dans un échantillon est fonction de deux facteurs: distance et direction. De manière générale des points proches les uns des autres montrent une variabilité plus faible que des points éloignés les uns des autres.

Le processus de calcul commence par estimer un variogramme qui représente la variabilité des altitudes en fonction de la distance entre les points de données. La deuxième phase réalise l'interpolation, en utilisant les coefficients de l'équation représentant le variogramme.

3.5.4. Courbure minimale

La première étape est réalisée par le calcul des mailles qui contiennent un point de donnée originel. Les valeurs initiales seront toujours respectées. Ensuite, la méthode applique un calcul itératif sur la surface. L'objectif de la méthode est d'obtenir un lissage entre les points connus.

Chaque segment est recalculé jusqu'à ce que la valeur des changements successifs soit inférieure à la valeur absolue de l'erreur maximale, ou que le nombre limite d'itérations soit atteint.

La valeur maximale de l'erreur ainsi que le nombre d'itérations sont fixés par l'utilisateur. Des valeurs raisonnables sont de l'ordre de 1% à 0,1% du relief total de l'échantillon. Un nombre très petit donnera une surface très lisse mais demandera un nombre d'itérations important.

3.5.5. Avantages, inconvénients des méthodes

Ces trois méthodes présentent bien entendu des avantages et des inconvénients. L'inverse des distances est la méthode la plus rapide, mais le krigeage et la courbure minimale donnent des résultats plus fiables dans la plupart des cas. Cependant lorsqu'il existe des zones avec peu de données utilisables, ces deux méthodes peuvent donner des résultats imprévisibles contrairement à la méthode de l'inverse des distances.

Le choix de la méthode d'interpolation est donc intimement lié au type de semis de points dont on dispose au départ et à l'importance que l'on souhaite donner à la restitution des données initiales après interpolation.

3.6. VERIFICATION QUALITATIVE DU MNT GEOLOGIQUE

Comme pour tout modèle, il convient de s'assurer, lorsque celui-ci est terminé, qu'il reflète bien la réalité. Aussi, le module CARTEGEO permet de recréer la carte d'affleurements des formations, en combinant les différentes grilles qui composent le modèle numérique géologique. Le résultat de ce module donne une carte topographique (grille GRD compatible avec SURFER) où les affleurements sont codés de 1 à N. Il suffit alors de comparer visuellement avec la carte géologique établie puis d'effectuer les corrections qui peuvent être nécessaires. Le fichier PARAMGEO qui décrit l'ordre de superposition des interfaces géologiques doit être présent dans le répertoire courant. Son format est le suivant :

PARAMGEO

- | |
|--|
| <ul style="list-style-type: none">- Nombre n d'interfaces géologiques- N lignes désignant les noms des interfaces (fichiers ".GRD") du plus récent vers le plus ancien.- N+1 intitulés décrivant les formations rencontrées. |
|--|

Exemple: MNT géologique

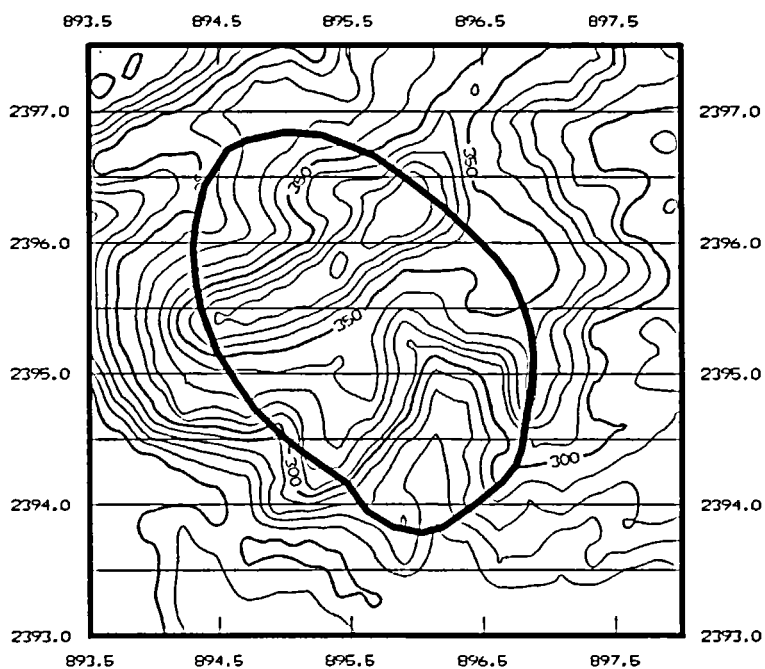
PARAMGEO

4
L13-T10
T10-T9
T9-T8B
T8B-T78A
Sinémurien - Hettangien (calcaires à *Gryphea arcuata*)
Rhétien (marnes de Levallois)
Keuper supérieur (Marnes bariolées)
Keuper moyen (Dolomie moellon)
Keuper inférieur (Marnes irisées gypsifères)

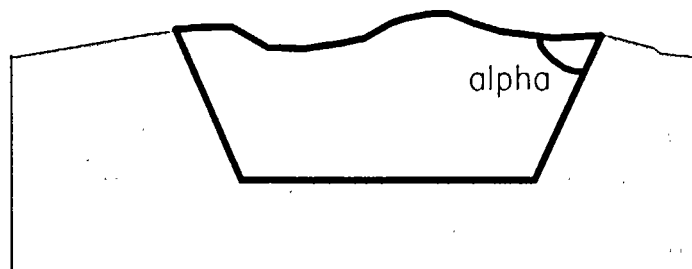
4. ENSEMBLE DES MODELISATIONS DISPONIBLES

Les simulations qui peuvent concerner soit des fosses (trous) soit des remblais ont besoin de trois paramètres pour fonctionner :

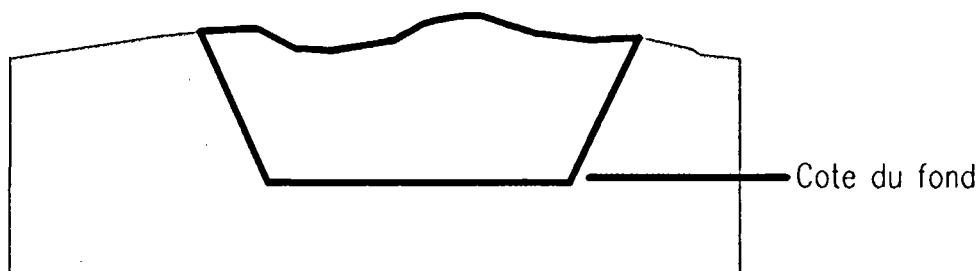
- le contour limite de l'ouvrage,



- l'angle de pente des talus, exprimé en degrés et par rapport à l'horizontale.



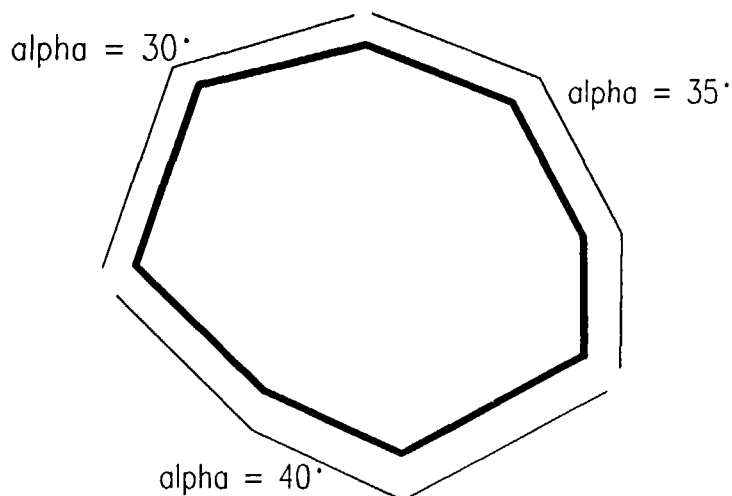
- la cote Z du fond (fosses) ou du toit (remblai) de la simulation.



4.1. DIGITALISATION DU CONTOUR DE MODELISATION

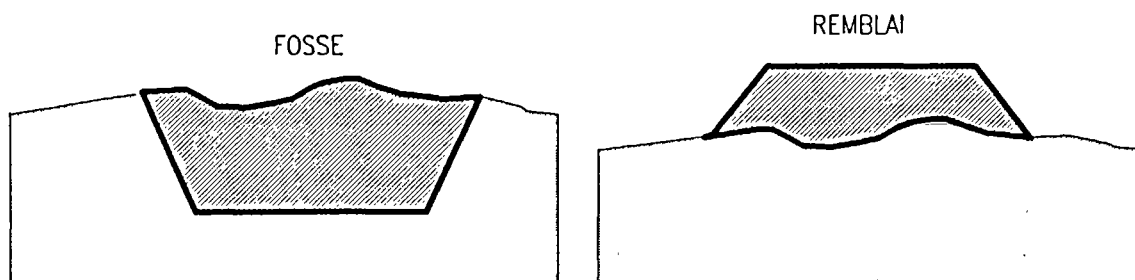
MODULE : *PERIFOS*

L'objectif de PERIFOS est de produire un fichier ASCII qui contient les coordonnées (en km) d'un polygone formant le périmètre de l'ouvrage modélisé. Le programme permet aussi d'affecter à chaque segment digitalisé, un angle qui servira de pente pour la modélisation.



4.2. LES DIFFERENTS TYPES D'OUVRAGES MODELISABLES

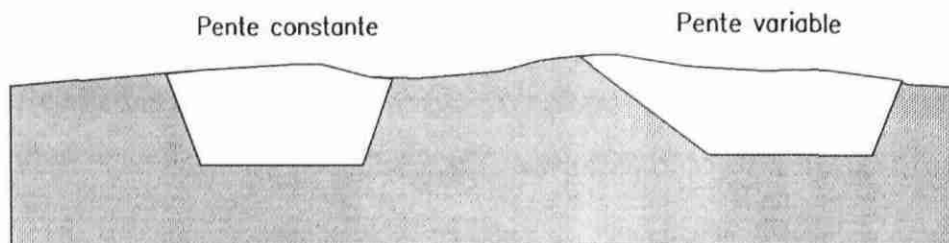
Comme il a été indiqué précédemment, GEOSURF est capable de modéliser des fosses et des remblais :



Les modélisations peuvent partir du bas vers le haut de l'ouvrage, ou du haut vers le bas.



Les pentes peuvent être constantes ou variables sur l'ensemble du contour.



Le résultat des simulations est constitué par une nouvelle surface topographique (fichier GRD type GEOSURF).

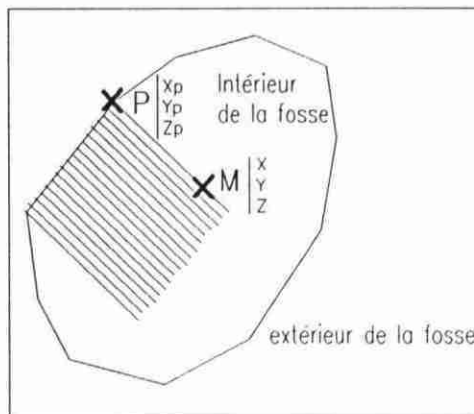
4.2.1. Les fosses

Un exemple de modélisation est donné en figure 2.

4.2.1.1. Le contour digitalisé est la limite supérieure d'une fosse

MODULE : *FOSSEHBK* si pente constante (voir annexe 1 planche 2)

MODULE : *FOSSEHBV* si pente variable (voir annexe 1 planche 3)



Pour chaque segment du contour, et ceci tant qu'on est à l'intérieur du polygone, on balaye la perpendiculaire issue du point courant P(xp,yp,zp). On attribue à un point M(x,y) de cette droite, la valeur d'altitude :

$$Z = Z_P - \text{PENTE} * ((X_P - X)^2 + (Y_P - Y)^2)^{1/2}$$

à condition que celle-ci soit supérieure à celle précédemment calculée en ce point.

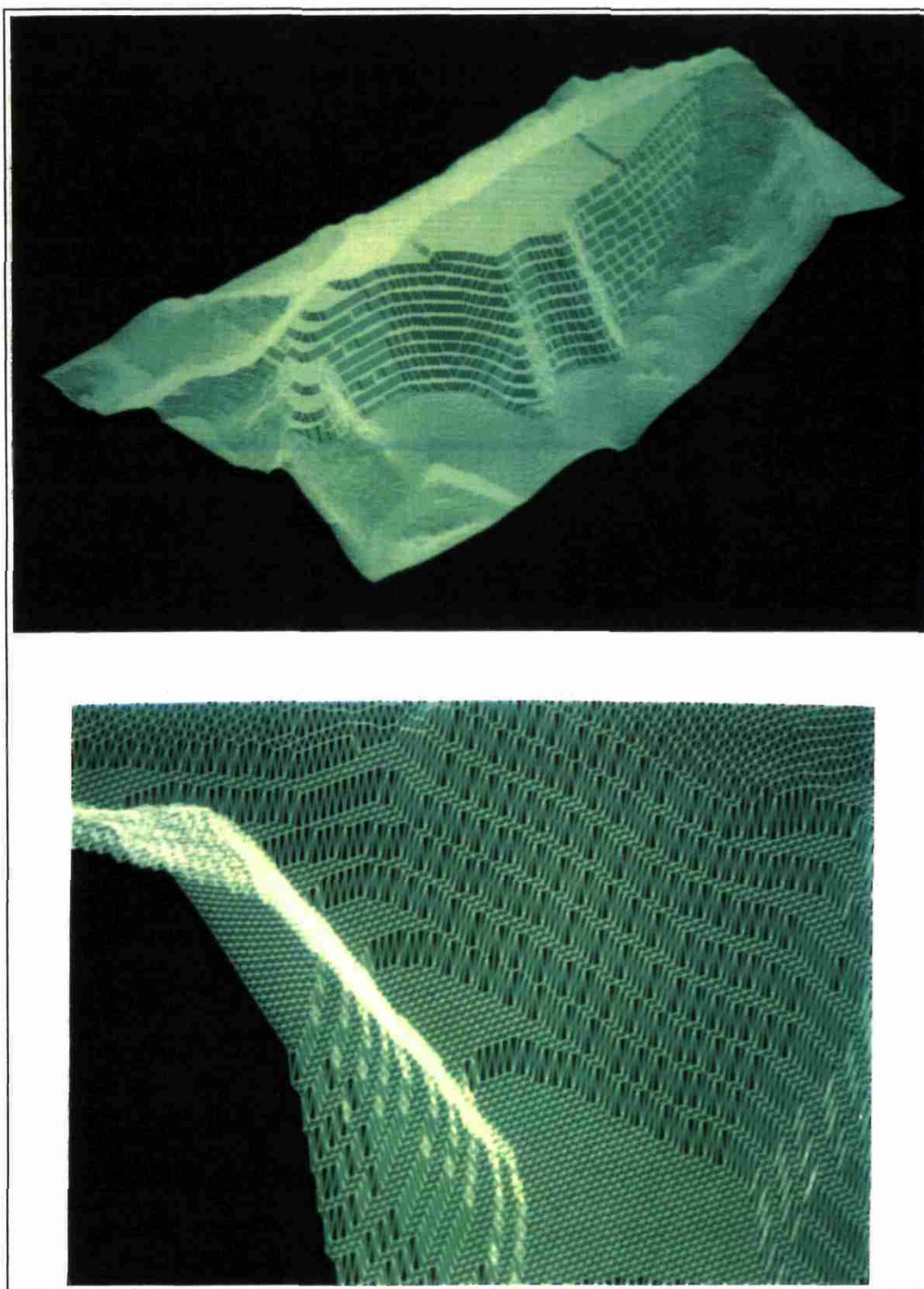


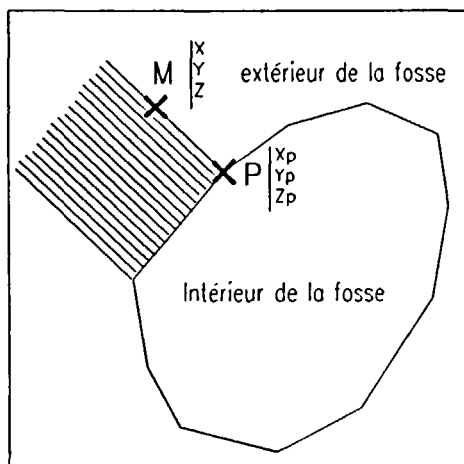
Figure 2 - Modélisation d'une carrière de roches massives
Vue globale et détails du maillage
(Photographies d'écran)

4.2.1.2. Le contour digitalisé est la limite inférieure d'une fosse

MODULE : *FOSSEBHK* si pente constante (voir annexe 1 planche 4)

MODULE : *FOSSEBHV* si pente variable (voir annexe 1 planche 5)

Les calculs se font sur la perpendiculaire du segment courant, coté extérieur.



Pour tout point M de cette droite, son altitude Z est égale à:

$$Z = Z_{\text{FOND}} + Z_P - \text{PENTE} * ((X_P - X)^2 + (Y_P - Y)^2)^{1/2}$$

où zfond désigne la cote du fond de la fosse.

De deux choses l'une :

- ou bien le point M(x,y) n'a pas encore été calculé (avec un segment précédent), auquel cas on lui affecte la valeur Z, en tenant compte du fait que si celle-ci est inférieure à la cote du fond la dernière valeur est retenue.
- ou bien, lors de précédents parcours, l'altitude du point M(x,y) a été calculée ; dans ce cas la nouvelle valeur Z ne lui est affectée que si elle est supérieure à celle précédemment calculée.

4.2.2. Les remblais

4.2.2.1. Le contour digitalisé est la limite supérieure d'un remblai

MODULE : *REMBHBK* si pente constante (voir annexe 1 planche 6)

MODULE : *REMBHBV* si pente variable (voir annexe 1 planche 7)

Le principe de calcul reste le même, à ceci près que : la nouvelle valeur d'altitude est calculée suivant la formule

$$Z = Z_{TOIT} - ZP + PENTE * ((X_P - X)^2 + (Y_P - Y)^2)^{1/2}$$

où z_{toit} désigne la cote supérieure du remblai.

Cette valeur n'est attribuée au point courant que si elle est inférieure à l'ancienne altitude du point.

4.2.2.2. Le contour digitalisé est la limite inférieure d'un remblai

MODULE : *REMBBHK* si pente constante (voir annexe 1 planche 8)

MODULE : *REMBBHV* si pente variable (voir annexe 1 planche 9)

Le principe de progression le long du contour est le même que précédemment; on attribuera à un point $M(x,y)$ de la perpendiculaire issue d'un point courant $P(x_p, y_p, z_p)$ la valeur :

$$Z = ZP + PENTE * ((X_P - X)^2 + (Y_P - Y)^2)^{1/2}$$

pourvu que celle-ci soit inférieure à l'ancienne cote en ce point.

4.3. OUTILS COMPLEMENTAIRES DE MODELISATION

4.3.1. Modification partielle d'une grille

MODULE : *BLANKED* (voir annexe 1 planche 10)

Compte tenu du nombre de mailles mises en oeuvre, il a fallu mettre au point des outils qui permettent de modifier globalement la valeur Z d'une grille pour une zone donnée.

Ces modifications peuvent se faire maille par maille dans le logiciel SURFER. Cette option sera à privilégier lorsque le nombre de mailles à modifier est inférieur à 20, ou lorsque ces mailles sont éparpillées dans le MNT.

Pour effectuer le remplacement d'une partie d'une grille par la même partie d'une autre grille, il convient de procéder de la façon suivante :

a) Générer le polygone qui entoure la zone à modifier.

b) Sous éditeur de texte:

- ajouter le chiffre 0 aux côtés du premier chiffre du fichier qui indique le nombre de points constituant le polygone.
- effacer la notion de pente du contour pour ne conserver que les coordonnées X et Y .

c) Utiliser le module GRID de SURFER option Modify-Blank:

- en entrée on donnera le nom du fichier grille à modifier, et le nom du fichier contour (modifié sous éditeur de texte).
- en sortie, on obtient une nouvelle grille ou la zone du polygone est masquée (BLANKED AREA).

d) Lancer le module BLANKED avec en paramètre

- le nom de la grille d'où proviennent les nouvelles valeurs,
- le nom de la grille masquée,
- le nom de la nouvelle grille résultat.

4.3.2. Carte d'affleurement

MODULE : *CARTEGEO* (voir annexe I planche 11)

Permet de reconstituer la carte d'affleurement depuis les données numériques qui constituent le MNT. Très utile pour vérifier le calage du modèle.

4.3.3. Calculs des cubatures déplacées

MODULE : *CUBEGEOL* (voir annexe I planche 12)

La composition du MNT géologique est décrite par le fichier *PARAMGEO* qui donne au module *CUBEGEOL* les caractéristiques de superposition des couches.

CUBEGEOL a besoin de connaître comme autres paramètres le nom de la grille contenant l'état initial topographique et le nom de la modélisation qui va être évaluée.

Les cubatures totales ou par classe de terrain sont alors calculées en m³.

NOTA: Une modélisation étant un calcul mathématique, l'utilisateur saura rester critique devant ses résultats et ne donnera jamais des valeurs de volumes sans les "ARRONDIR" auparavant, compte tenu de la précision de son MNT.

4.3.4. Carte de visualisation (impact visuel)

MODULE : *RADAR* (voir annexe I planche 13)

4.3.4.1. Principe

Pour générer une carte de visualisation par rapport à un point de vue, et à une cote donnée, on considère qu'un observateur placé à un point coté P(x,y,z), balaye un angle de 360° autour de ce point. Le pas d'incrément de l'angle est fixé par l'utilisateur. A chaque fois qu'un point du MNT est à l'intersection de la droite qui part du point de visualisation, un symbole est affecté à la maille, ce qui permet de visualiser ces zones de façon graphique.

L'impact d'une zone donnée d'un MNT est inversement proportionnel à la distance qui la sépare du point de vue. Ainsi sur deux points ayant les mêmes caractéristiques (mais présentant un éloignement différent par rapport au point de vue) celui qui sera le plus proche aura l'impact le plus important.

Les paramètres à fournir à ce module sont donc:

- le nom de la grille sur laquelle est fait le test d'impact,
- le nom du fichier qui réceptionne les symboles (avec leur coordonnées),
- les coordonnées X, Y, Z du point de vue,
- la hauteur H de l'observateur (2m environ, si l'observateur est positionné au niveau du sol),
- le pas d'incrément de l'angle de balayage (ex: 0.5°),
- le pas de progression dx , sur la trajectoire de balayage (ex: 0.01km),
- la longueur de la "zone proche" qui représente la zone à ne pas trop noircir, afin que le point de vue reste visible (ex: 0.1km).

4.3.4.2. Visualisation

La visualisation du résultat se fait sur une carte topographique générée avec le logiciel SURFER . Le fichier des symboles généré par le module RADAR est superposé à la carte 2D (SURFER - option POST). Une petite étoile convient généralement pour donner une bonne visualisation du phénomène.

4.3.5. Densité de visualisation

MODULE : *DENSITE* (voir annexe 1 planche 14)

A partir d'un fichier de points issus de RADAR, le module DENSITE génère un fichier grille (type GEOSURF) qui contient dans chaque maille, le nombre total des points interceptés. DENSITE offre la possibilité de faire un lissage avec les mailles contiguës. Avec SURFER , il est alors très simple de tracer une carte d'iso-densité.

Les paramètres nécessaires au fonctionnement de ce module sont :

- le fichier contenant le positionnement des symboles,
- le nom de la grille qui va être tracée,
- le nom de la grille qui va servir de référence (nombre de lignes, de colonnes,...)
- la valeur de l'indice de lissage (nombre de mailles prises de part et d'autre pour calculer le lissage de la surface); si l'on ne souhaite pas de lissage, mettre 0.

4.3.6. Superposer plusieurs modélisations sur une même grille

MODULE : *INCRUSTE* (voir annexe 1 planche 15)

Lorsque des modélisations ont été faites sur plusieurs zones d'une même grille, il peut être intéressant de les visualiser sur un même MNT. INCRUSTE réalise cette opération. Les paramètres suivants doivent lui être fournis :

- la grille contenant la topographie initiale,
- la grille contenant la modélisation à exporter,

- le fichier récepteur qui contiendra les modélisations importées.

NOTA: *Le fichier récepteur doit exister dans le répertoire courant. Si vous devez le créer, vous pourrez prendre comme copie une des grilles contenant une des modélisations à importer.*

4.3.7. Affectation d'une cote Z à l'ensemble d'une zone donnée d'une grille

MODULE : *PUIS38* (voir annexe 1 planche 16)

Le principe est strictement le même que précédemment, si ce n'est que la zone masquée est affectée par une même cote à valeur constante.

Une fois le passage dans GRID effectué, il suffit de lancer le module PUIS38 avec en paramètres :

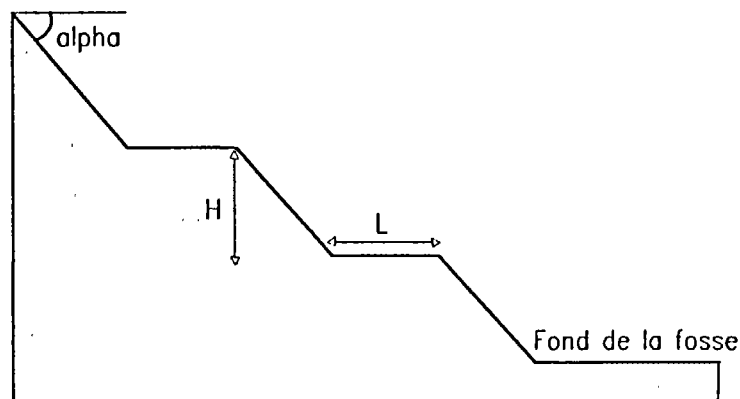
- le nom de la grille masquée,
- le nom de la nouvelle grille résultat,
- la cote à attribuer aux zones masquées.

NOTA: *Les valeurs masquées prennent la valeur 1,70141E+38*

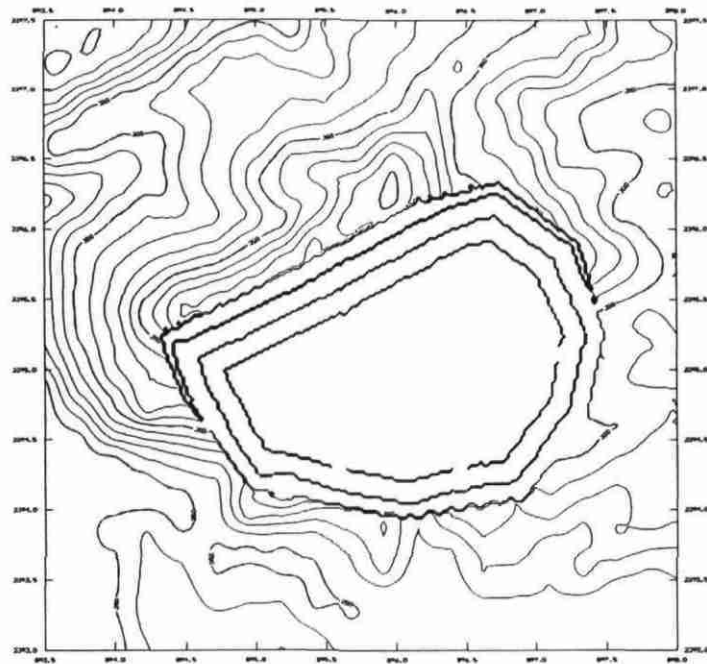
4.3.8. Modélisation des gradins

MODULE : *GRADIN* (voir annexe 1 planche 17)

Ce module permet de "sculpter" des gradins sur les parois d'une fosse. Il doit connaître l'angle α , la hauteur H des gradins, leur largeur L , et la cote du fond de la fosse.



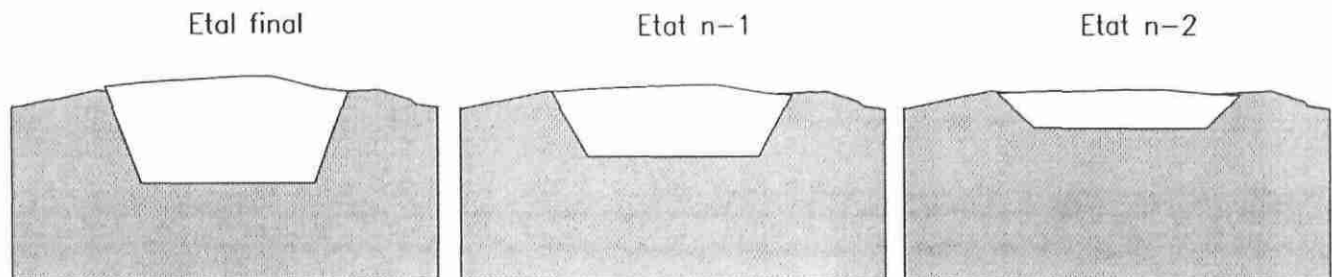
Le résultat est une nouvelle grille type GEOSURF où pourront être réalisés des coupes, des calculs de volumes et des sorties graphiques sous forme de cartes ou de blocs diagrammes.



4.3.9. Remonter la cote du fond d'une fosse

MODULE : *RELEVE* (voir annexe 1 planche 18)

Ce module permet de décomposer les différentes étapes de la vie d'une carrière; il part de l'état final (cote n) et permet de générer les étapes n-x à des cotes intermédiaires.



A chaque étape, les volumes pourront être calculés et les plans (ou blocs diagrammes) dessinés.

La figure 3 montre un exemple de ces différentes étapes sur un projet étudié pour un industriel de la région.

4.3.10. Réduire le nombre de mailles

MODULE : *REDUIGRD* (voir annexe 1 planche 19)

Ce module permet de réduire le nombre de lignes et de colonnes d'un MNT. Pour cela on utilise un coefficient de réduction: un facteur de 2 réduira de moitié le nombre de mailles du MNT

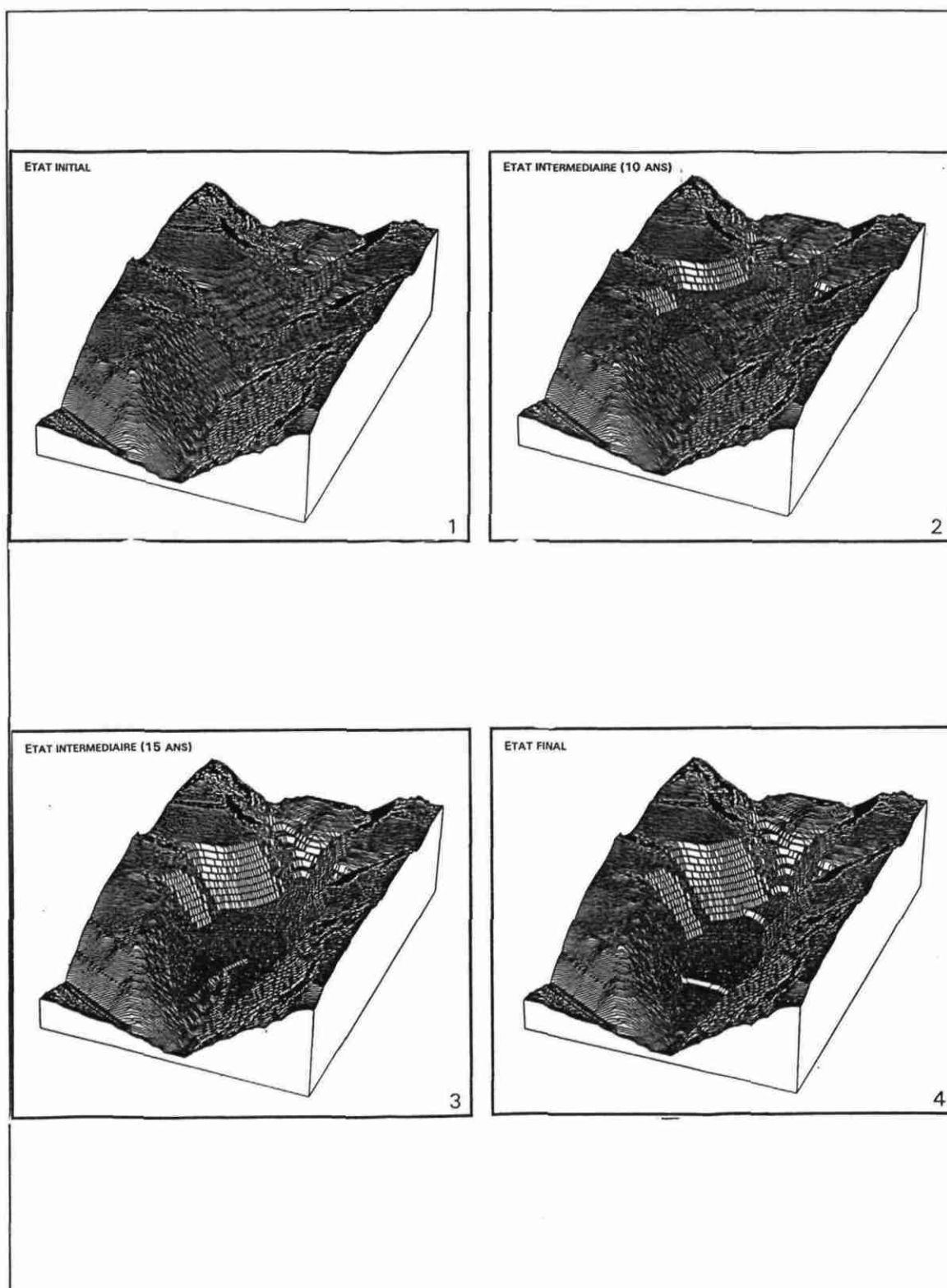


Figure 3 - Modélisation d'une carrière
Différentes étapes de creusement

5. VISUALISATION GRAPHIQUE DES RESULTATS (SURFER)

5.1. CONTEXTE D'UTILISATION

La version Windows de GEOSURF utilise le logiciel SURFER (Golden Software Inc.) pour effectuer les sorties graphiques en deux et trois dimensions. Ce logiciel était utilisé dans sa version DOS avec la première version de GEOSURF; en mars 1994 la version Windows a été diffusée et nous a donc permis de faire évoluer notre propre produit.

Tout comme GEOSURF, SURFER pour Windows s'affranchit de la limite des 640 ko du DOS et permet donc de traiter des grilles (GRD) de taille quasi illimitée (fonction de la taille de la mémoire installée sur le micro ordinateur).

L'ergonomie du logiciel s'est améliorée de façon très significative avec une interface graphique adaptée. Les performances (rapidité d'exécution) ont cependant légèrement diminué du fait de la gestion du mode graphique sous Windows; cette remarque s'applique également au logiciel GEOSURF.

5.2. VUES EN DEUX DIMENSIONS (2D)

Pour obtenir de façon exhaustive toutes les informations relatives à la manipulation de ces options, l'utilisateur se rapportera à la notice de SURFER.

L'exécution de ces cartes est rapide, et de très bonne qualité graphique. Le programme génère des cartes topographiques (avec des courbes de niveaux); il permet notamment de figurer:

- des cotes orientées sur les courbes, avec choix de la couleur, de la police des caractères, de la taille, du format et de la précision.
- des lignes en trait continu, en tirets, épaisses ou fines et avec des hachures indiquant la pente.
- des courbes de couleurs différentes ou des zones de couleur.
- des contours lissés ou normaux (bruts).
- des intervalles de contours réguliers ou irréguliers.
- une mise à l'échelle indépendante pour X et pour Y
- des positionnements, depuis plusieurs fichiers, de symboles fixes ou proportionnels avec des étiquettes numériques ou de type texte. Rotation individuelle des symboles.
- des bordures de carte avec des graduations optionnelles définies par l'utilisateur, incluant des annotations numériques et des titres pour les axes.
- des blocs de texte à n'importe quel angle, couleur, police de caractère et taille.
- une édition-modification des données de la grille, déplacement, mise à l'échelle et rotation du texte, le tout à l'écran.
- des zones masquées.
- un habillage complet issu d'une digitalisation externe.

Le résultat final est stocké dans un fichier macro (.SRF) qui satisfait aux normes graphiques de Golden Software. Il s'imprimera avec le gestionnaire d'impression de Windows.

La figure 4 montre quelques possibilités de graphiques 2D réalisés avec SURFER.

5.3. VUES EN TROIS DIMENSIONS (3D)

Pour obtenir de façon exhaustive toutes les informations relatives à la manipulation de ces options, l'utilisateur se rapportera à la notice de SURFER.

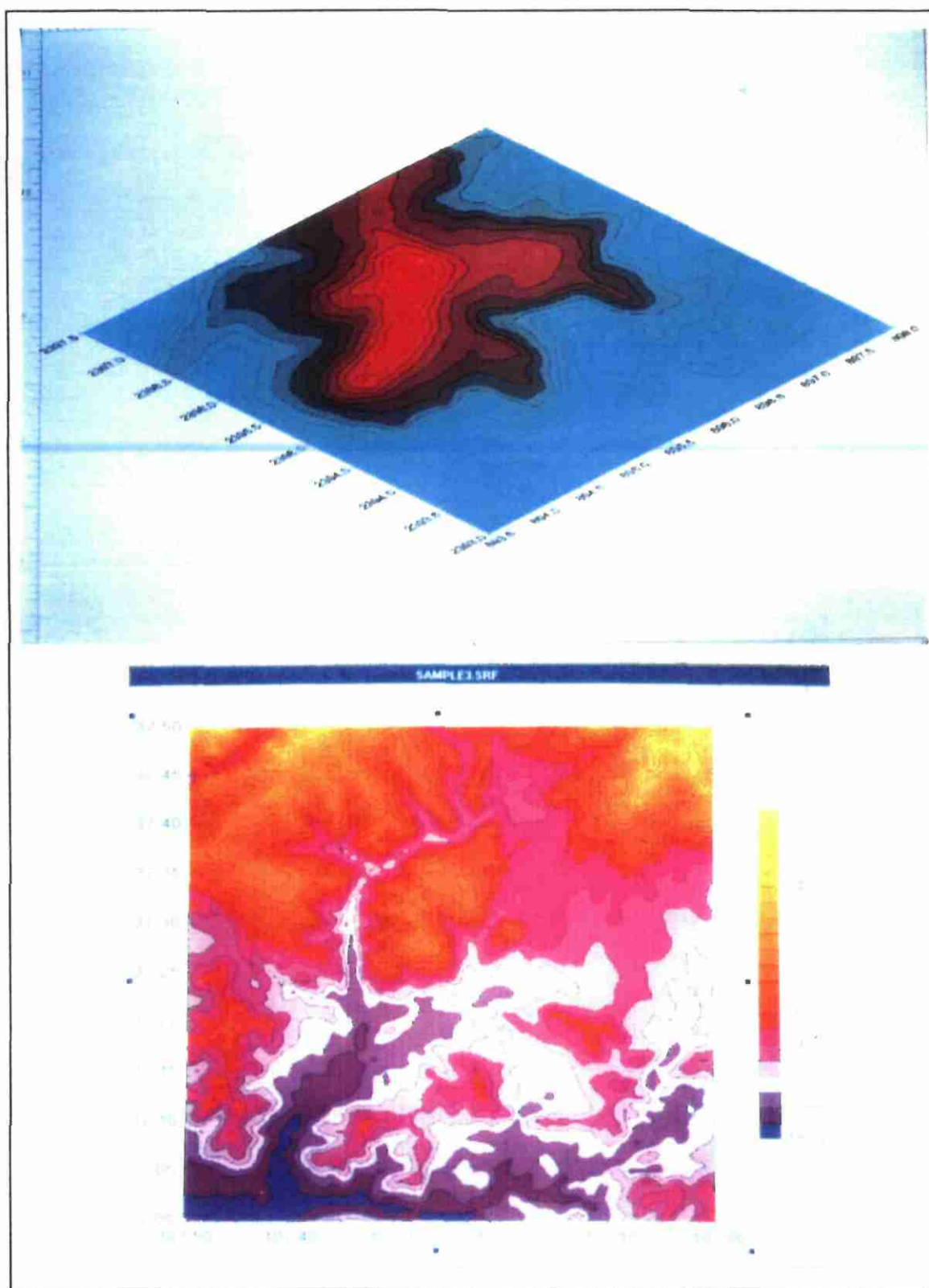
L'exécution de ces blocs diagrammes est simple. Elle donne des vues 3D très pertinentes que l'on peut combiner avec des vues en deux dimensions. Différents éléments peuvent être figurés :

- des surfaces de plusieurs couleurs.
- des projections orthographiques ou en perspective.
- gestion des surfaces cachées, rotation et basculement sous n'importe quel angle.
- plages de couleurs à intervalles altimétriques réguliers ou irréguliers.
- des positionnements depuis plusieurs fichiers, de symboles fixes ou proportionnels avec des étiquettes numériques ou de type texte. Rotation individuelle des symboles.
- axes 3D avec des graduations optionnelles définies par l'utilisateur, incluant des graduations numériques et des titres.
- une mise à l'échelle indépendante pour X , Y et Z.
- des blocs de texte à n'importe quel angle, couleur, police de caractères et taille.
- déplacement, mise à l'échelle et rotation du texte à l'écran de façon interactive.
- un habillage complet issu d'une digitalisation externe.
- possibilité de lissage des surfaces.

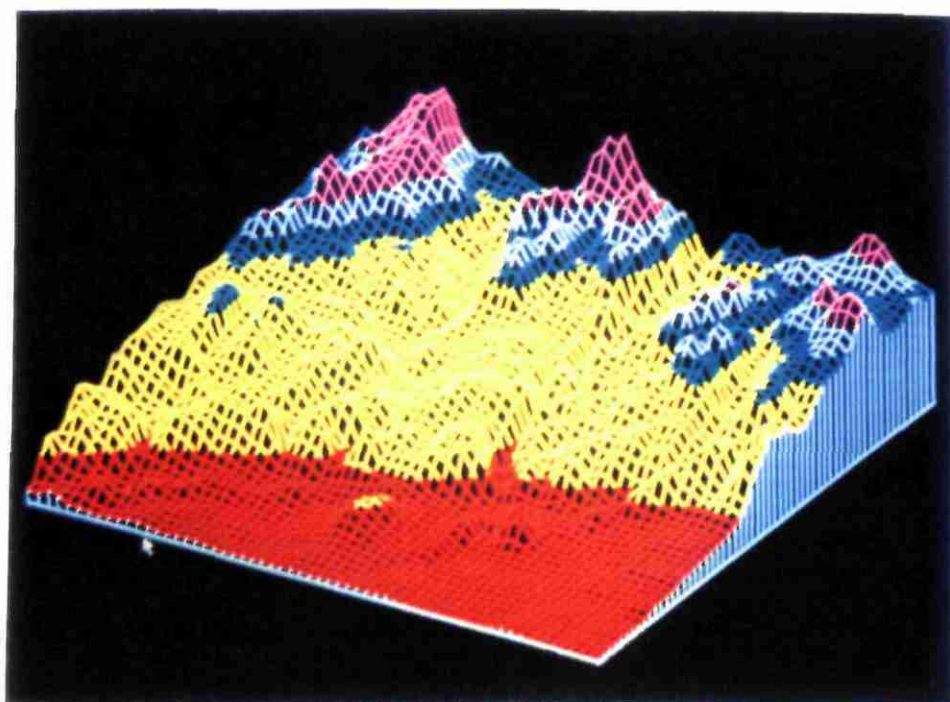
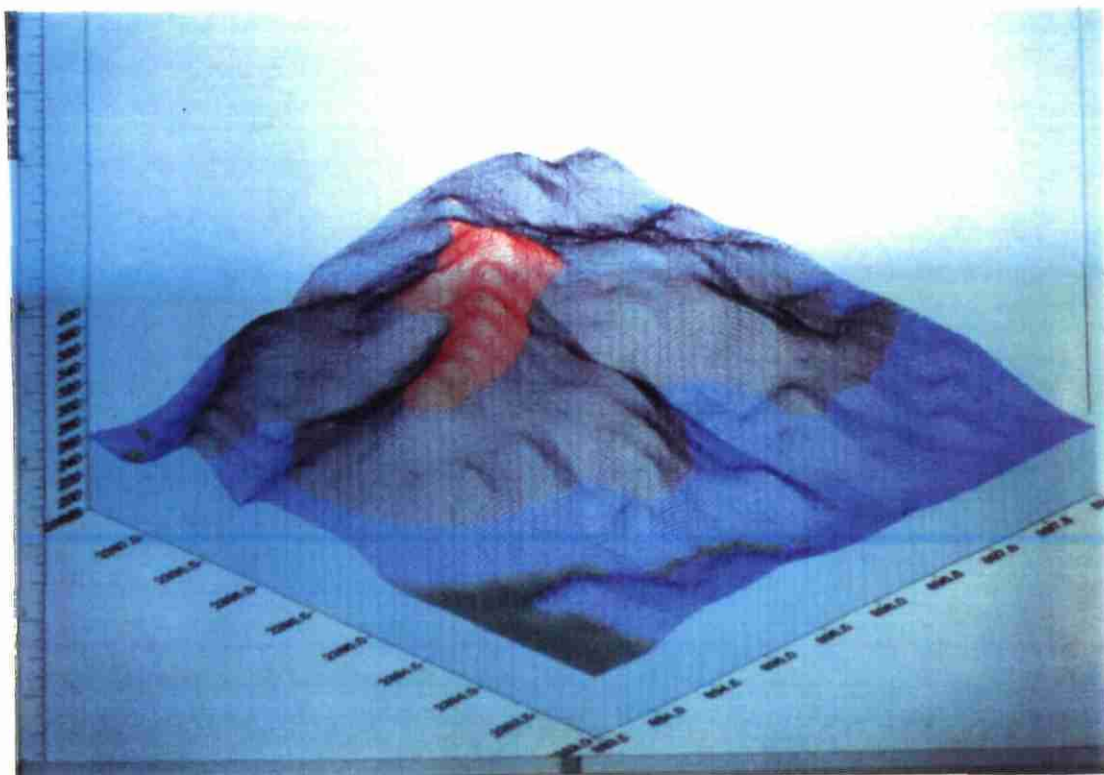
Le résultat final est stocké dans un fichier macro (.SRF) qui satisfait aux normes graphiques de Golden Software. Il s'imprimera avec le gestionnaire d'impression de Windows.

La figure 5 montre quelques possibilités de graphiques 3D réalisés avec SURFER.

La figure 6 montre la possibilité de combiner des vues en 2 et 3 dimensions.



**Figure 4 - Sorties graphiques en 2D réalisées avec GEOSURF
(Photographies d'écran)**



**Figure 5 - Sorties graphiques en 3D réalisées avec GEOSURF
(Photographies d'écran)**

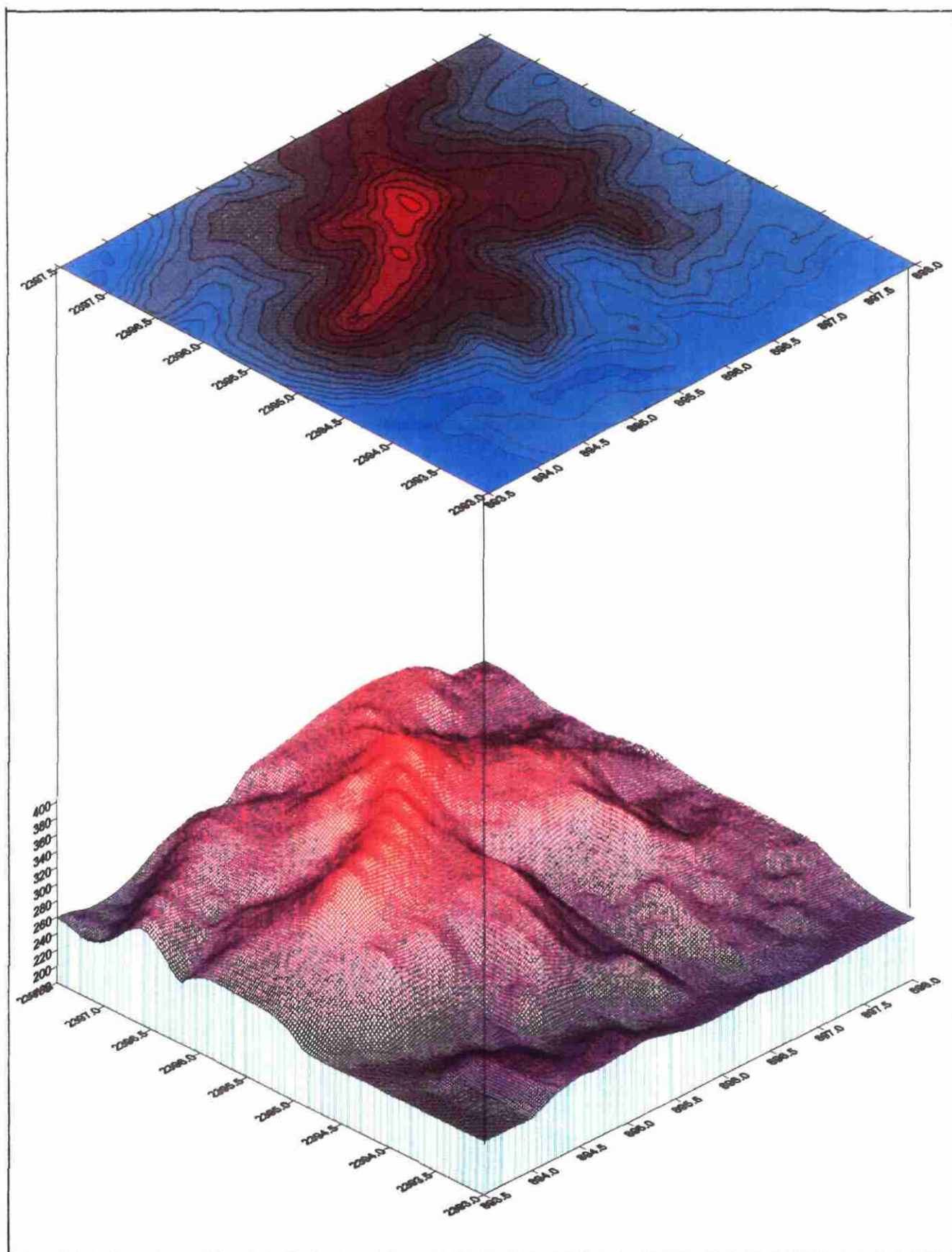


Figure 6 - Combinaison 2D et 3D réalisée avec SURFER

6. CONCLUSIONS SUR LES RESULTATS OBTENUS

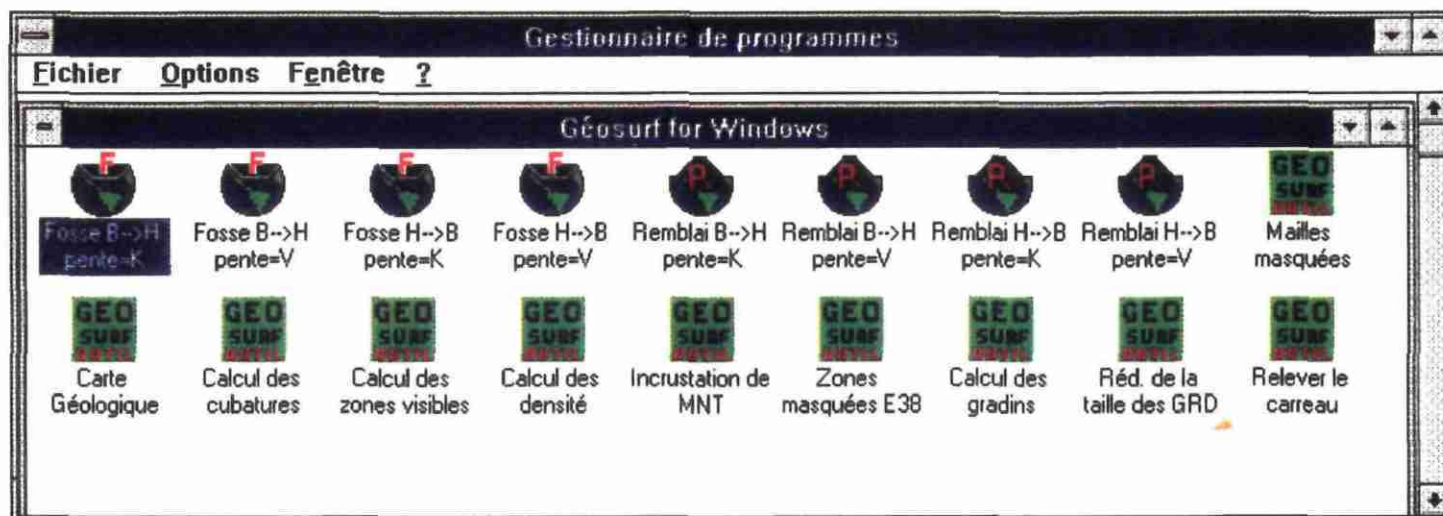
Ce projet de recherche a permis d'atteindre un certain nombre d'objectif :

- améliorer de façon conséquente l'outil existant en le faisant évoluer,
- conforter les investissements effectués depuis 1988,
- explorer les derniers concepts de la programmation par événement de type objet,
- doter la Région d'un outil performant dans un délai relativement court, afin de pouvoir répondre aux préoccupations concernant les problèmes de stockage de déchets et d'ouverture de carrières,
- impliquer les industriels locaux aux travers des différentes études exécutées pour leur compte; leurs remarques et leurs besoins ont guidé les lignes directrices de ce projet.

Ce produit GEOSURF est actuellement en phase d'application sur des cas réels en collaboration avec des industriels de la région. On peut d'ors et déjà affirmer que les résultats obtenus vont au-delà de prévisions les plus optimistes qui avaient pu être faites lors du montage du projet de recherche.

ANNEXE 1

LES MODULES DE GEOSURF DANS LEUR ENVIRONNEMENT WINDOWS



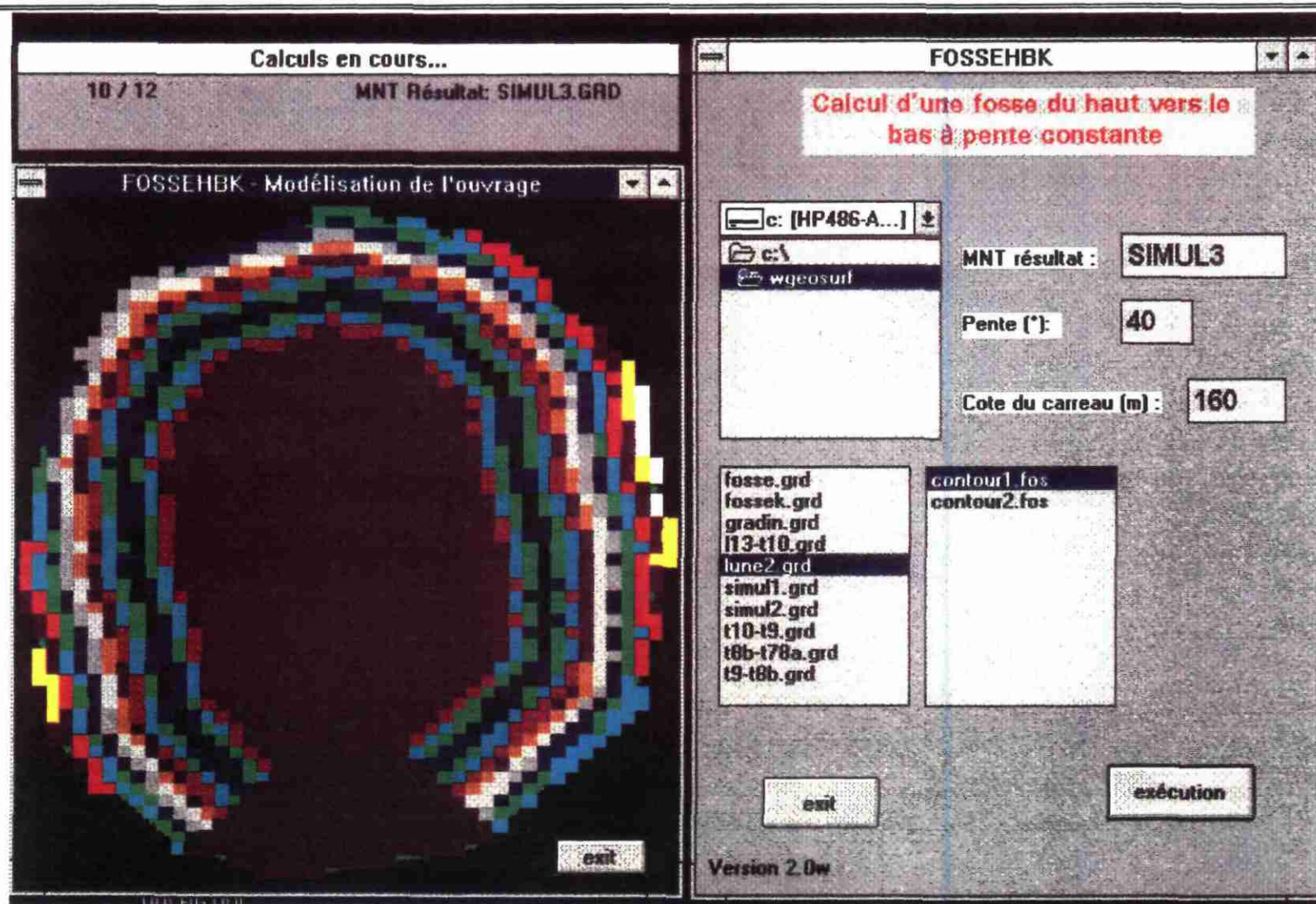
R 38098

CONSEIL REGIONAL PROVENCE-ALPES-COTE D'AZUR
CAO — Modélisation tridimensionnelle du sous-sol — Logiciel Geosurf

Planche 1

Liste des différents modules sous leur environnement Windows 3.11





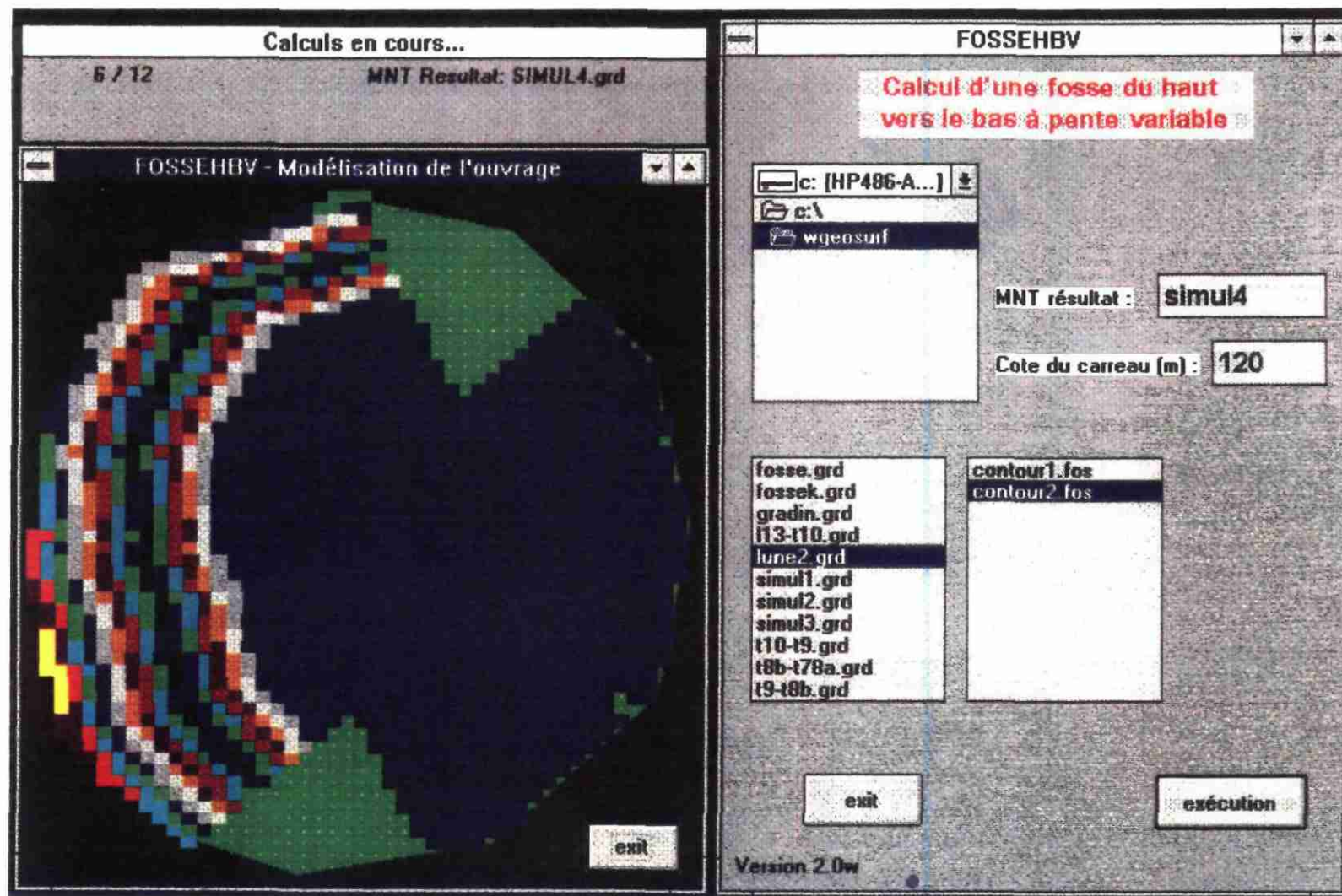
R 38098

CONSEIL REGIONAL PROVENCE-ALPES-COTE D'AZUR
CAO — Modélisation tridimensionnelle du sous-sol — Logiciel Geosurf

Planche 2

Modélisation d'une fosse à pente constante
Le contour supérieur est connu





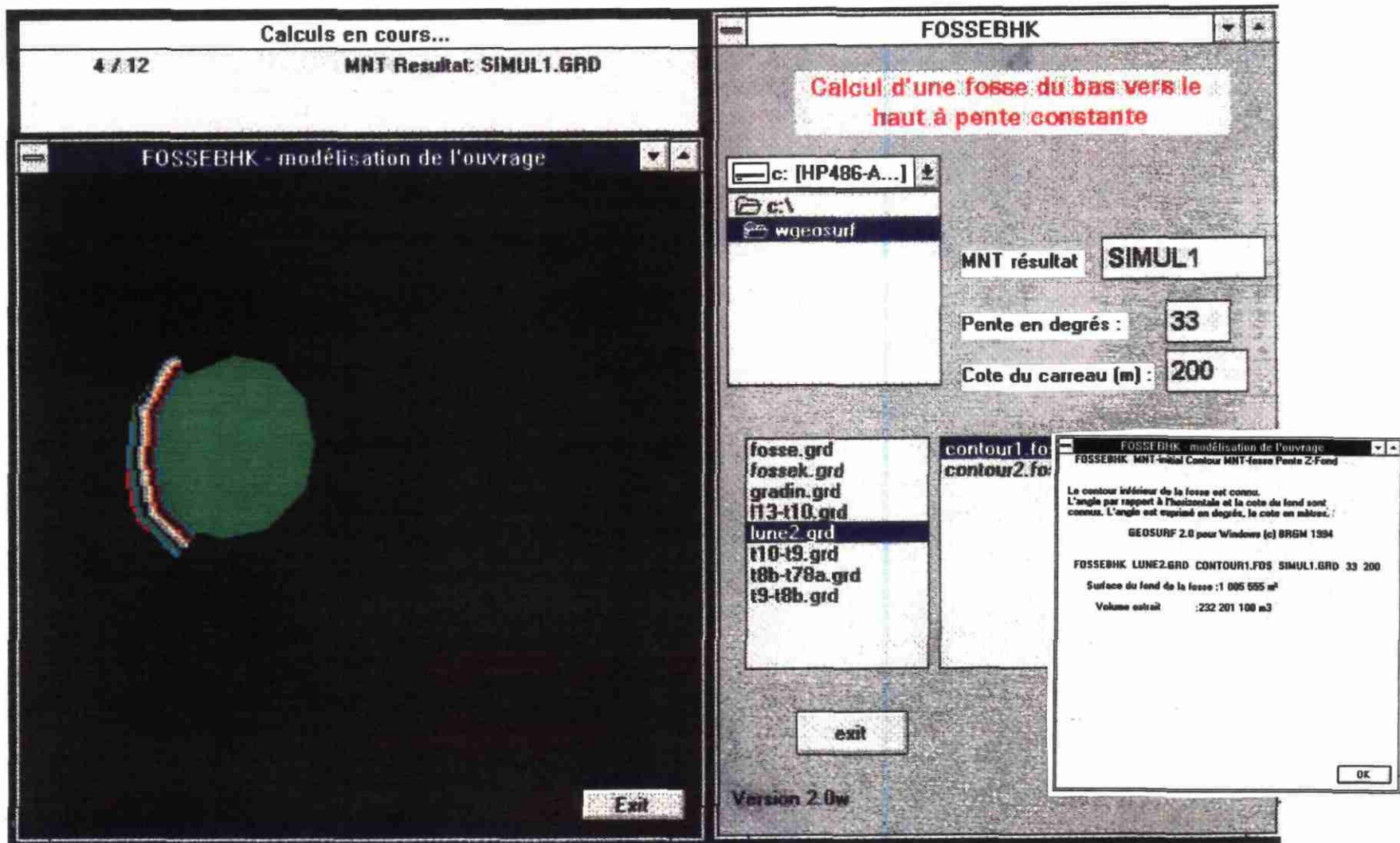
R 38098

CONSEIL REGIONAL PROVENCE-ALPES-COTE D'AZUR
CAO — Modélisation tridimensionnelle du sous-sol — Logiciel Geosurf

Planche 3

Modélisation d'une fosse à pente variable
Le contour supérieur est connu





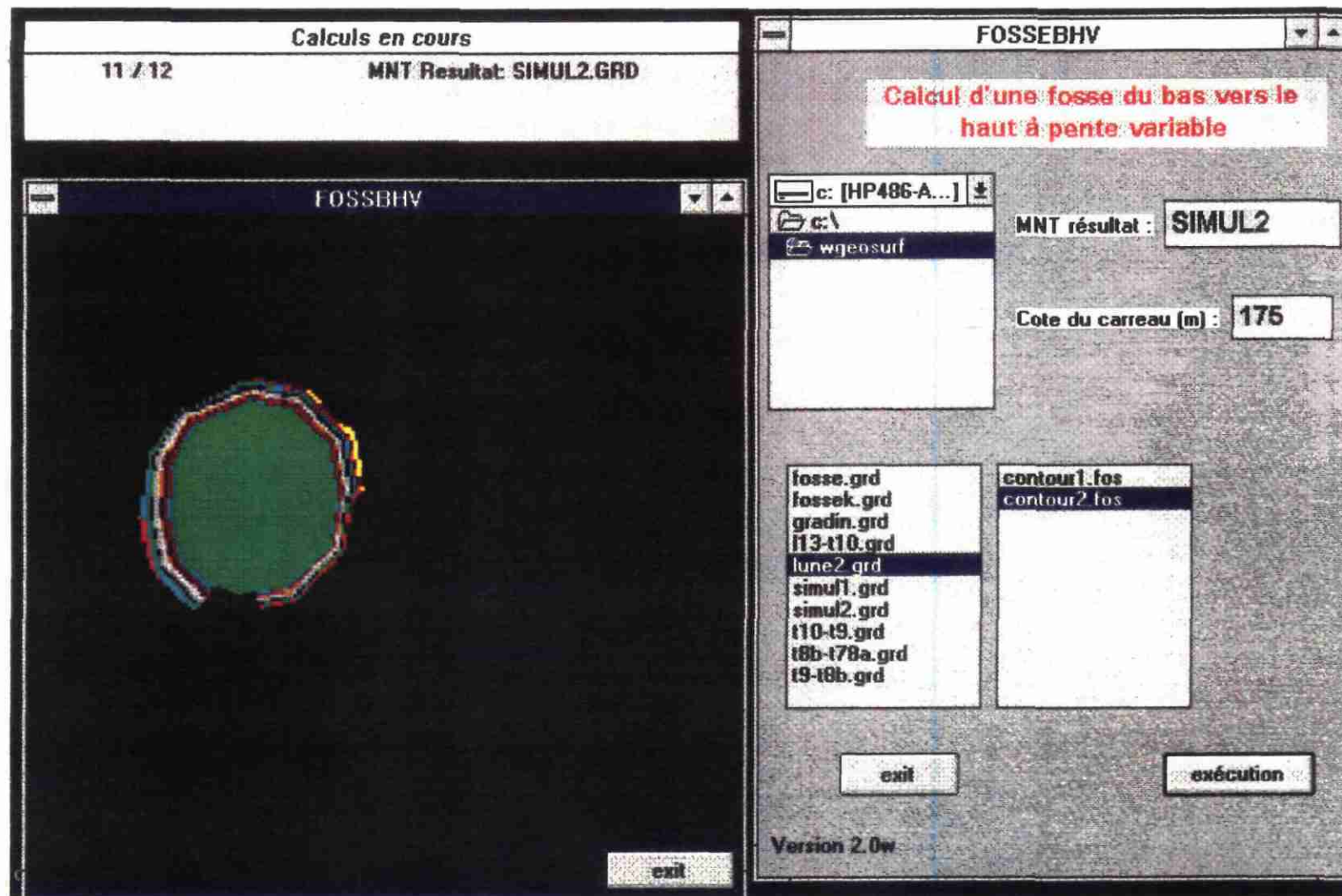
R 38098

CONSEIL REGIONAL PROVENCE-ALPES-COTE D'AZUR
CAO — Modélisation tridimensionnelle du sous-sol — Logiciel Geosurf

Planche 4

Modélisation d'une fosse à pente constante
Le contour inférieur est connu





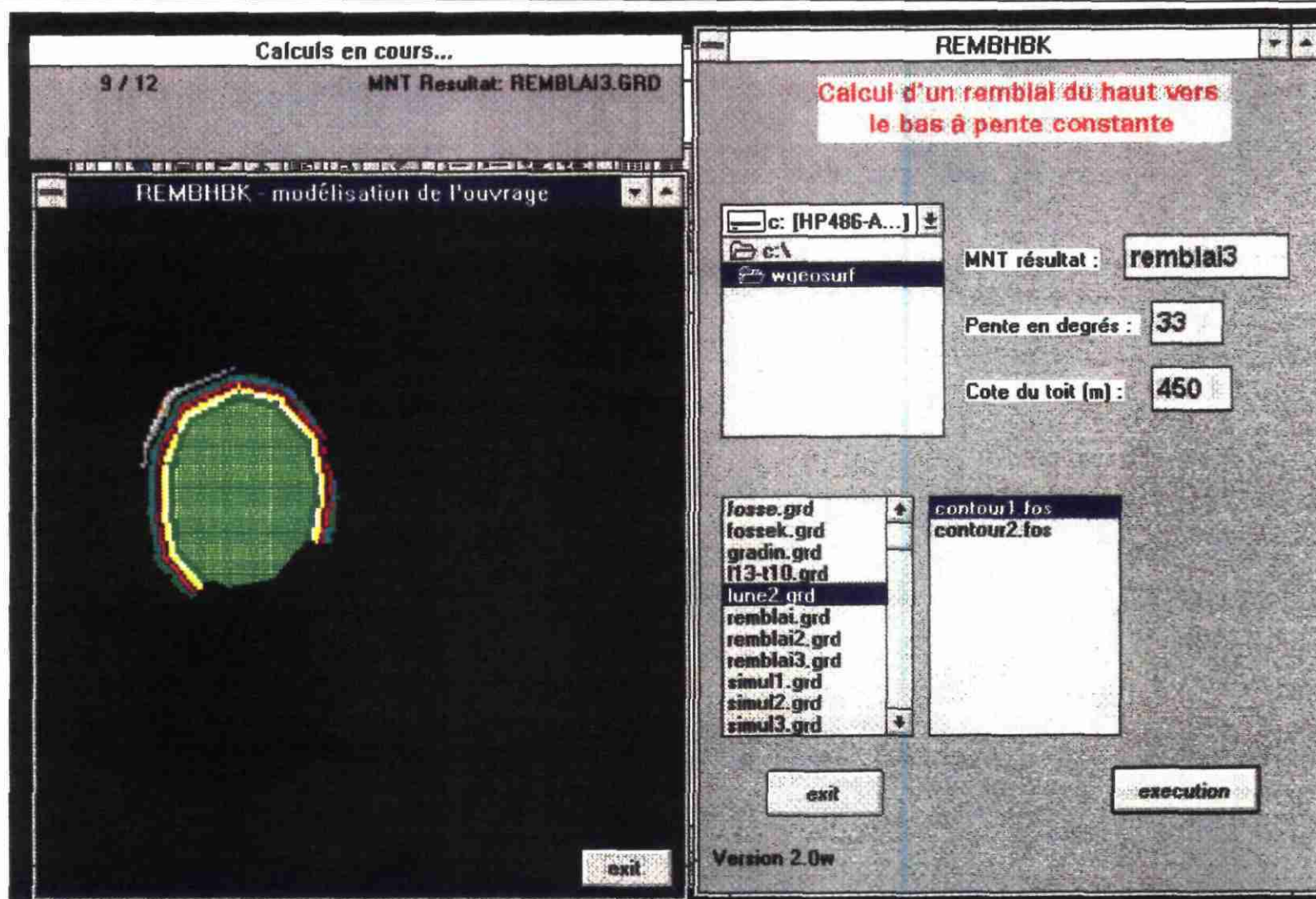
R 38098

CONSEIL REGIONAL PROVENCE-ALPES-COTE D'AZUR
CAO — Modélisation tridimensionnelle du sous-sol — Logiciel Geosurf

Planche 5

Modélisation d'une fosse à pente variable
Le contour inférieur est connu





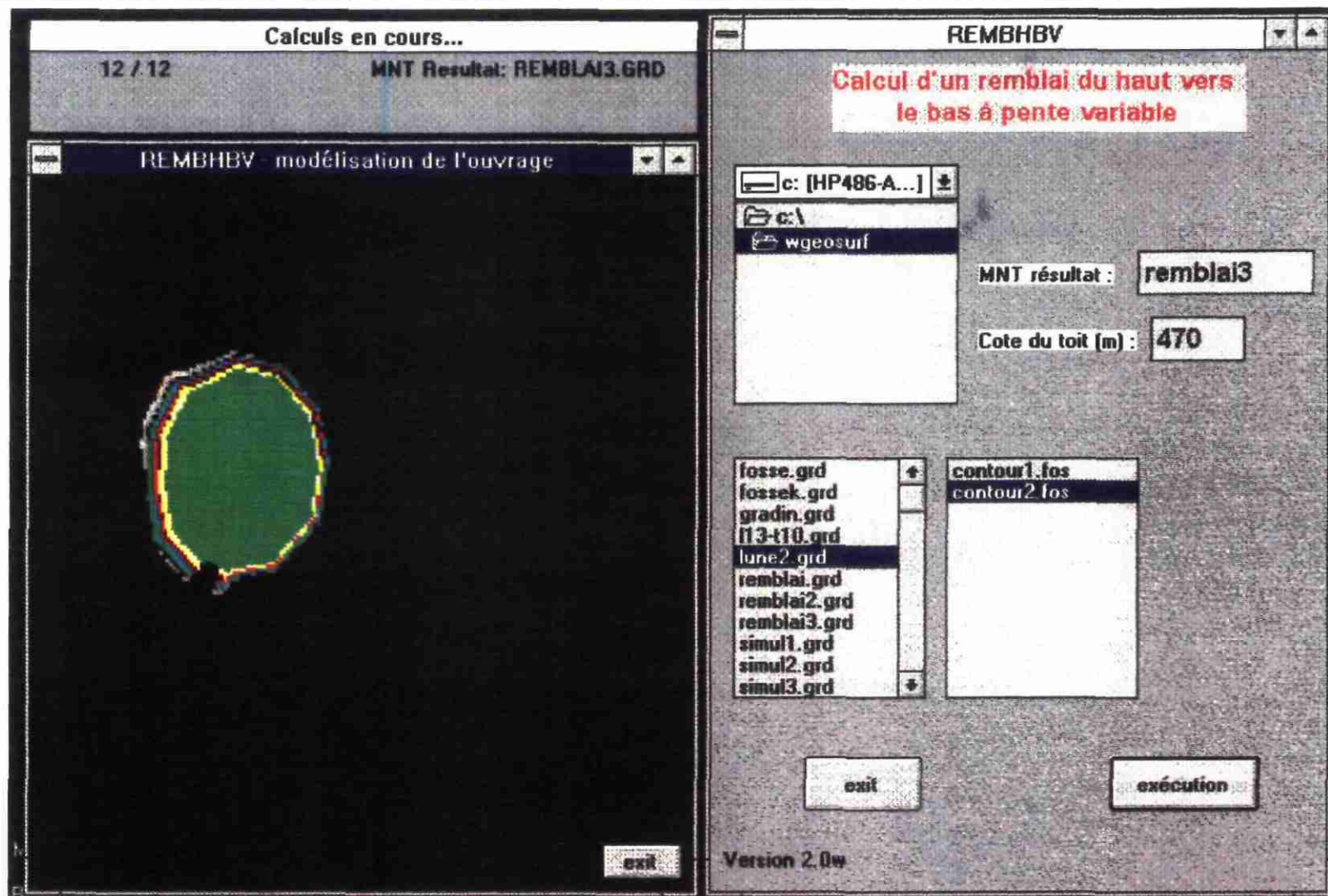
R 38098

CONSEIL REGIONAL PROVENCE-ALPES-COTE D'AZUR
CAO — Modélisation tridimensionnelle du sous-sol — Logiciel Geosurf

Planche 6

Modélisation d'un remblai à pente constante
Le contour supérieur est connu





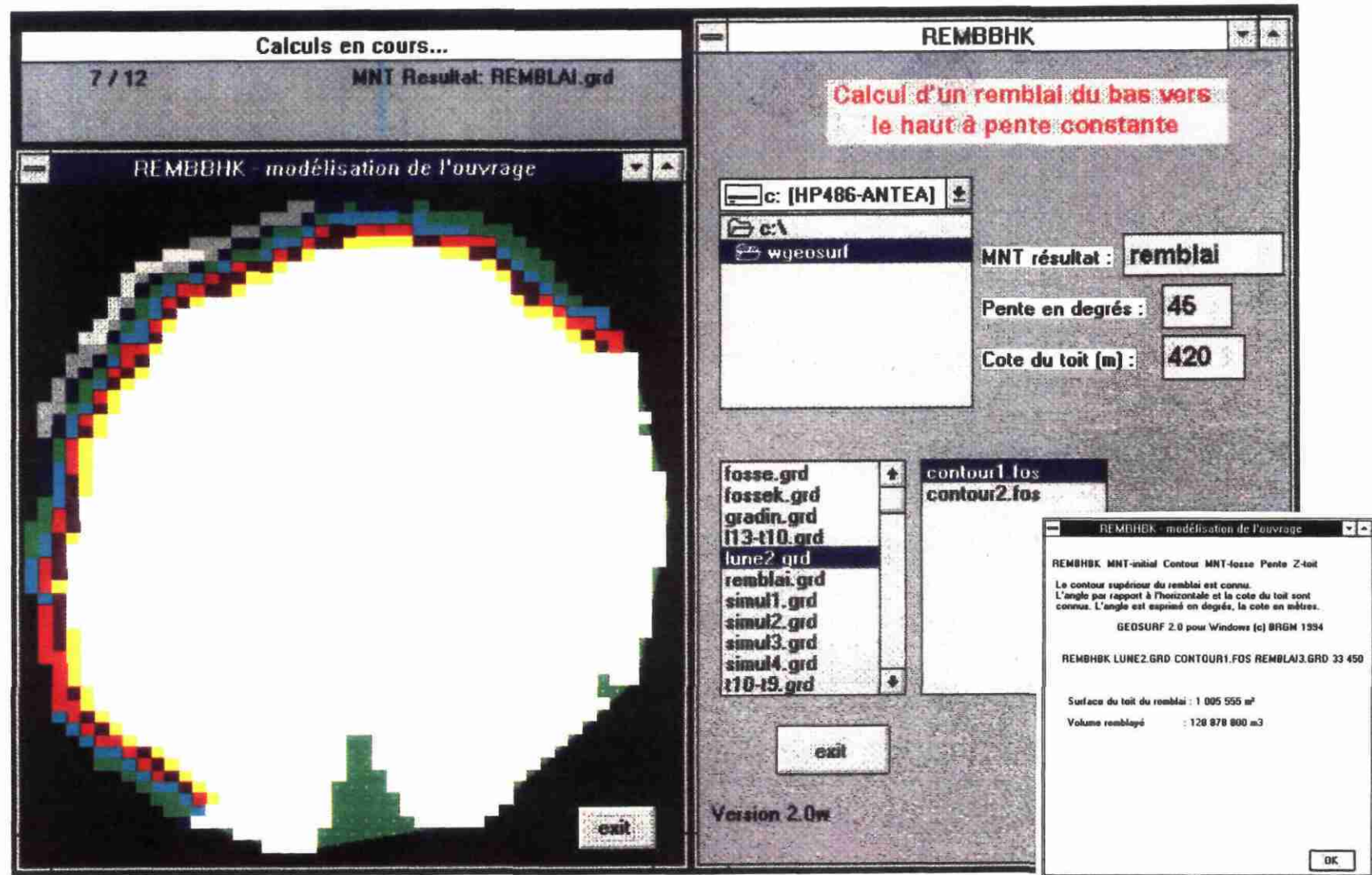
R 38098

CONSEIL REGIONAL PROVENCE-ALPES-COTE D'AZUR
CAO — Modélisation tridimensionnelle du sous-sol — Logiciel Geosurf

Planche 7

Modélisation d'un remblai à pente variable
Le contour supérieur est connu





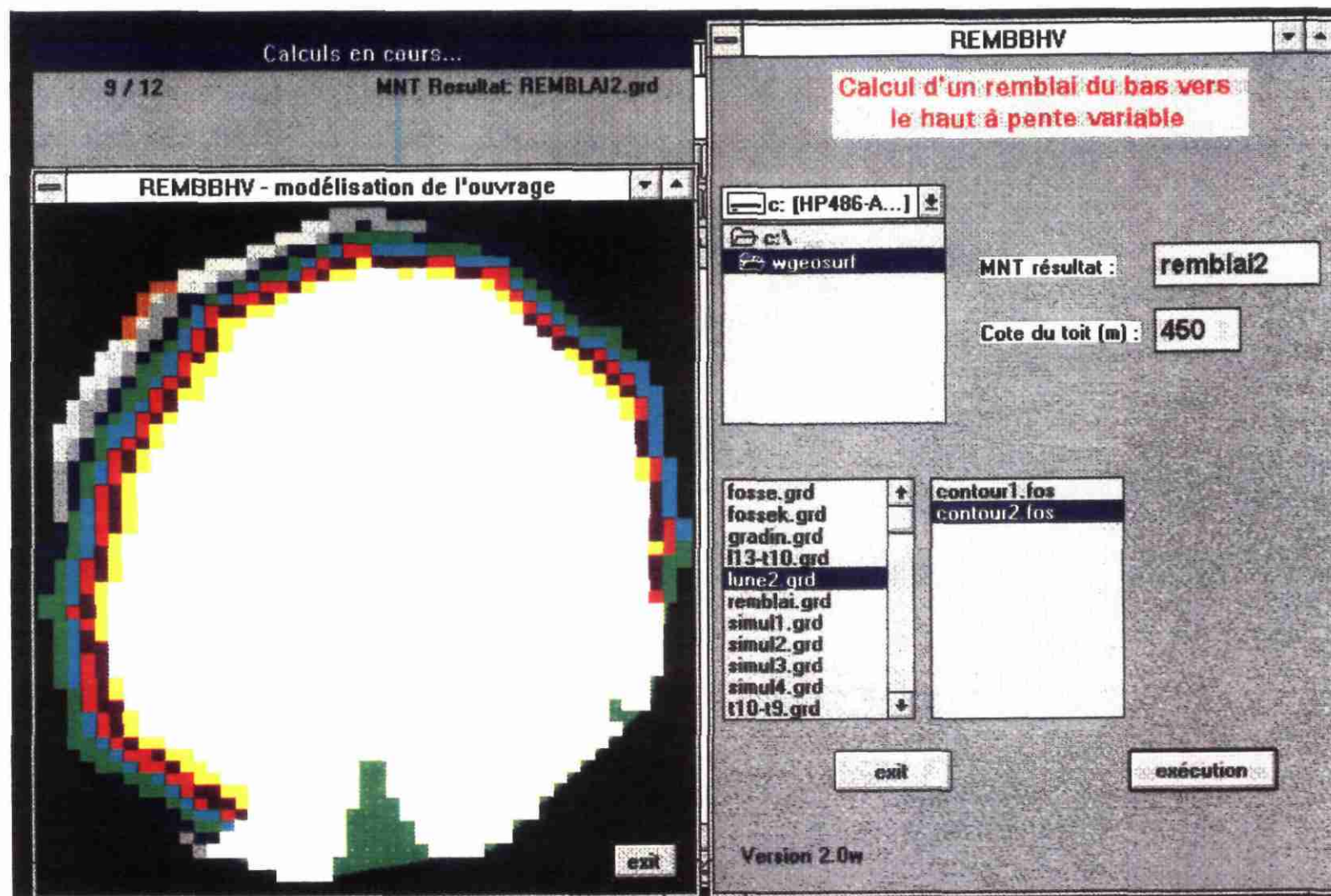
R 38098

CONSEIL REGIONAL PROVENCE-ALPES-COTE D'AZUR
 CAO — Modélisation tridimensionnelle du sous-sol — Logiciel Geosurf

Planche 8

Modélisation d'un remblai à pente constante
 Le contour inférieur est connu





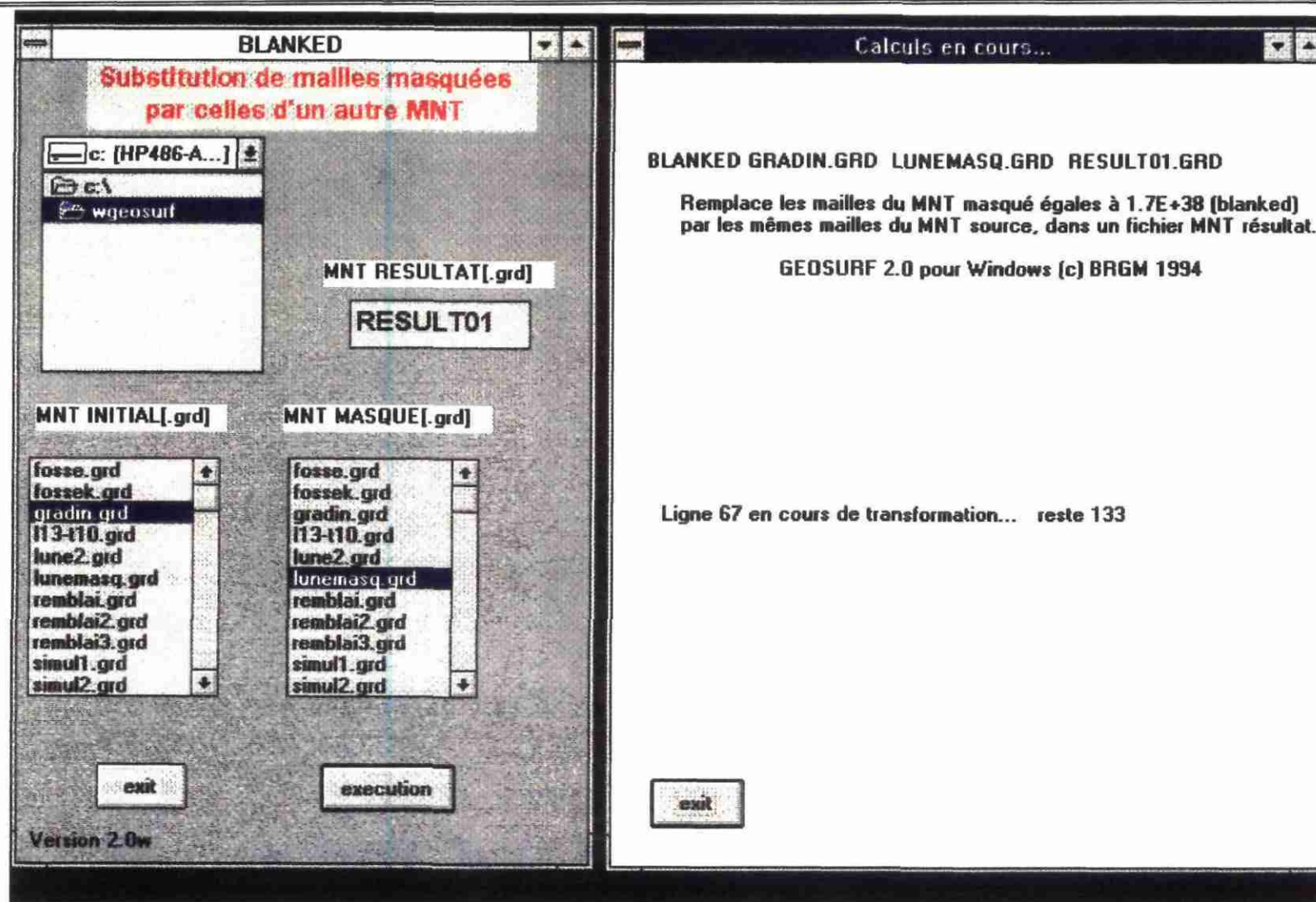
R 38098

CONSEIL REGIONAL PROVENCE-ALPES-COTE D'AZUR
CAO — Modélisation tridimensionnelle du sous-sol — Logiciel Geosurf

Planche 9

Modélisation d'un remblai à pente variable
Le contour inférieur est connu





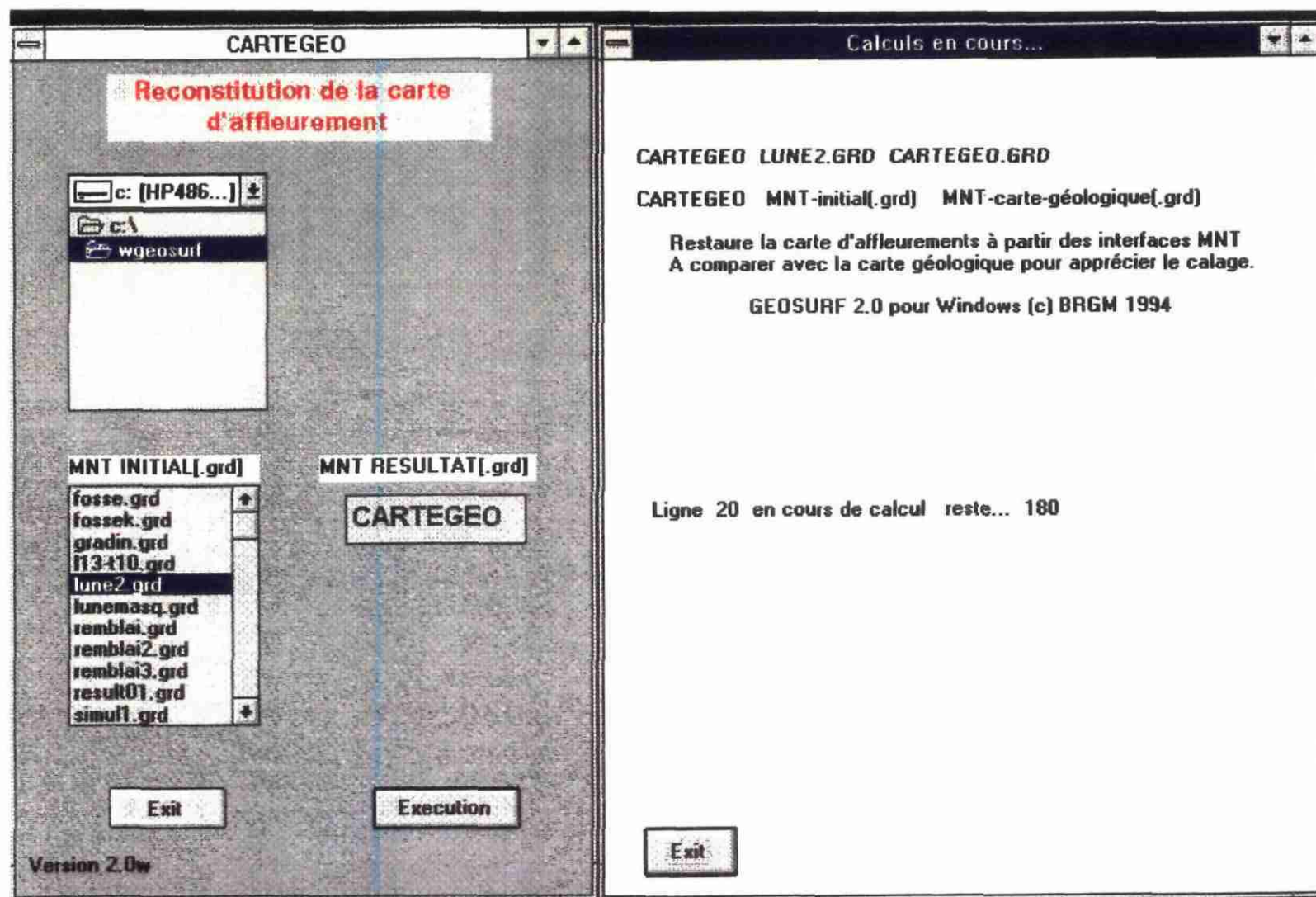
R 38098

CONSEIL REGIONAL PROVENCE-ALPES-COTE D'AZUR
CAO — Modélisation tridimensionnelle du sous-sol — Logiciel Geosurf

Planche 10

Utilitaire GEOSURF
Substitution des mailles masquées d'un MNT par les mêmes mailles d'un autre MNT





R 38098

CONSEIL REGIONAL PROVENCE-ALPES-COTE D'AZUR
CAO — Modélisation tridimensionnelle du sous-sol — Logiciel Geosurf

Planche 11

Utilitaire GEOSURF
Reconstitution de la carte d'affleurement depuis les données numériques du MNT



Calculs en cours...

CUBEGEOL LUNE2.GRD SIMUL1.GRD N

CUBEGEOL MNT-initial(.grd) MNT-fosse(.grd) Impression-calcul (O/N)

Calcule les cubatures par classe de terrain. Peut imprimer les résultats. Le fichier PARAMGED est indispensable.

GEOSURF 2.0 pour Windows (c) BRGM 1994

Ligne :100

Sinémurien-Hettangien	0	m3
Rhétien (marnes de	578 597	m3
Keuper supérieur	17 695 620	m3
Keuper moyen	19 372 470	m3
Keuper inférieur	566 377	m3

volume total extrait : 38 213 060 m3

Exit

CUBGEOL

Calcul des cubatures mobilisées

c: [HP486-ANT...]

c:\

wgeosurf

Impression-calcul:

Oui Non

MNT-initial(.grd):

cartgeo.grd
 fosse.grd
 fossek.grd
 gradin.grd
 l13-l10.grd
 lune2.grd
 lunemasq.grd
 remblai.grd
 remblai2.grd
 remblai3.grd
 result01.grd
 simul1.grd

MNT-fosse(.grd):

cartgeo.grd
 fosse.grd
 fossek.grd
 gradin.grd
 l13-l10.grd
 lune2.grd
 lunemasq.grd
 remblai.grd
 remblai2.grd
 remblai3.grd
 result01.grd
 simul1.grd

Version 2.0w

Exit

Execution

R 38098

CONSEIL REGIONAL PROVENCE-ALPES-COTE D'AZUR
CAO — Modélisation tridimensionnelle du sous-sol — Logiciel Geosurf

Planche 12

Utilitaire GEOSURF
Calcul des cubatures des différents terrains



Calcul des zones visibles depuis un point de vue

c: [HP486...]

c:\

☒ wgeosurf

Fichier-resultat(.dat):

MNT(.grd)

- cartegeo.grd
- fosse.grd
- fossek.grd
- gradin.grd
- l13-t18.grd
- lune2.grd**
- lunemasq.grd
- remblai.grd
- remblai2.grd
- remblai3.grd

Point de Vue:

X	Y	Z
897	2394.5	315

Incrément de l'angle de balayage :

Hauteur de l'oeil au niveau du point de vue en m (ex: 2m)

Zone-Proche (zone non noircie) en km :

Pas de trame initial en km (ex0.01km)

Version 2.0w

Calculs en cours...

RADAR lune2.grd visib04 897 2394.5 315 .002 1 .01 .2

Syntaxe:

RADAR MNT(.grd) Fich-pnts(.dat) X Y Z H A dX Zone

Extrait les points visibles du MNT depuis un point de vue P(X,Y,Z), avec une hauteur H, un angle de balayage A, un pas de progression dX, avec une Zone dégagée autour du point de vue pour le localiser

GEOSURF 2.0 pour Windows (c) BRGM 1994

Angle 276° en cours de calcul reste... 83°

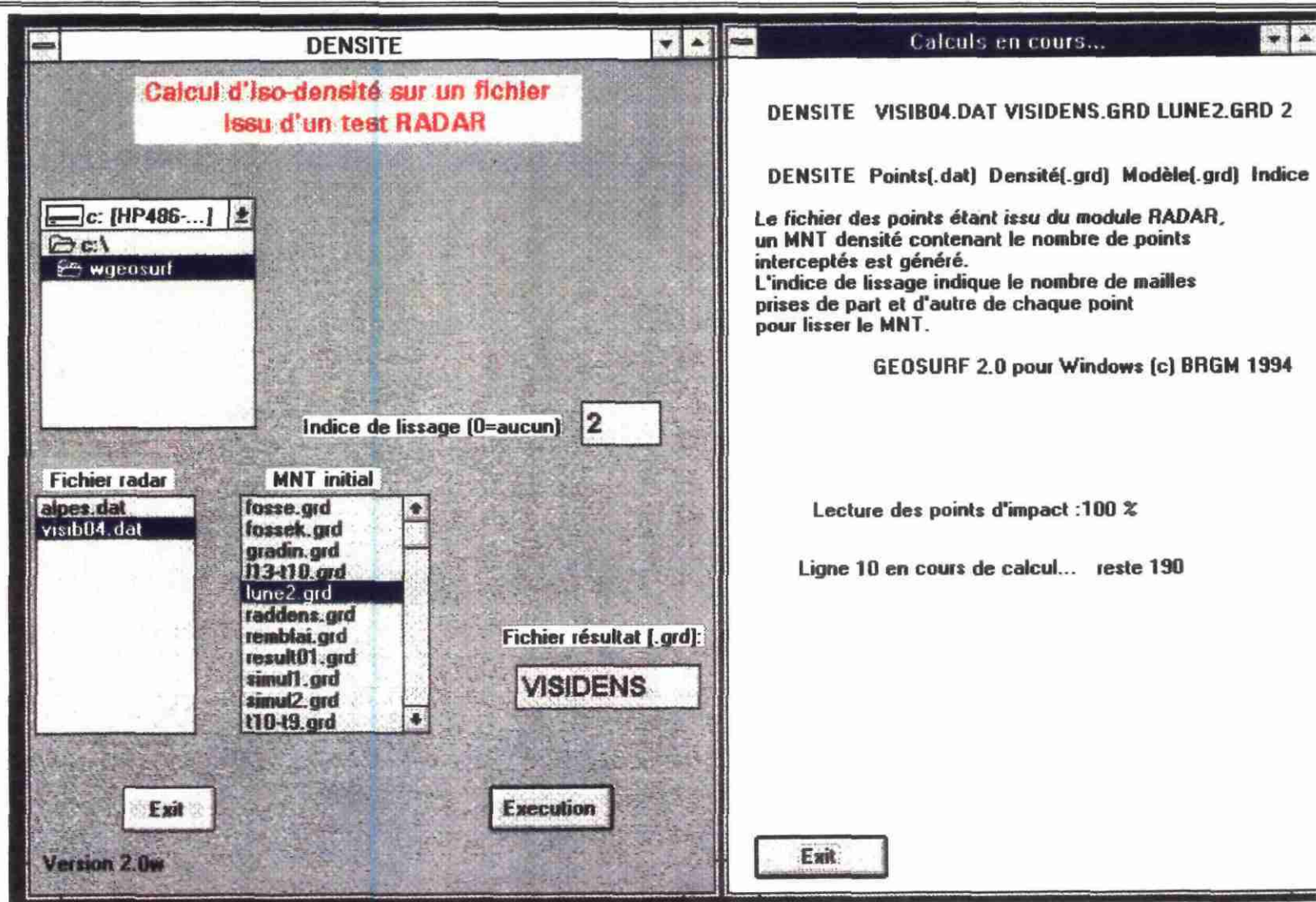
R 38098

CONSEIL REGIONAL PROVENCE-ALPES-COTE D'AZUR
CAO — Modélisation tridimensionnelle du sous-sol — Logiciel Geosurf

Planche 13

Utilitaire GEOSURF
Calcul des zones visibles depuis un point de vue





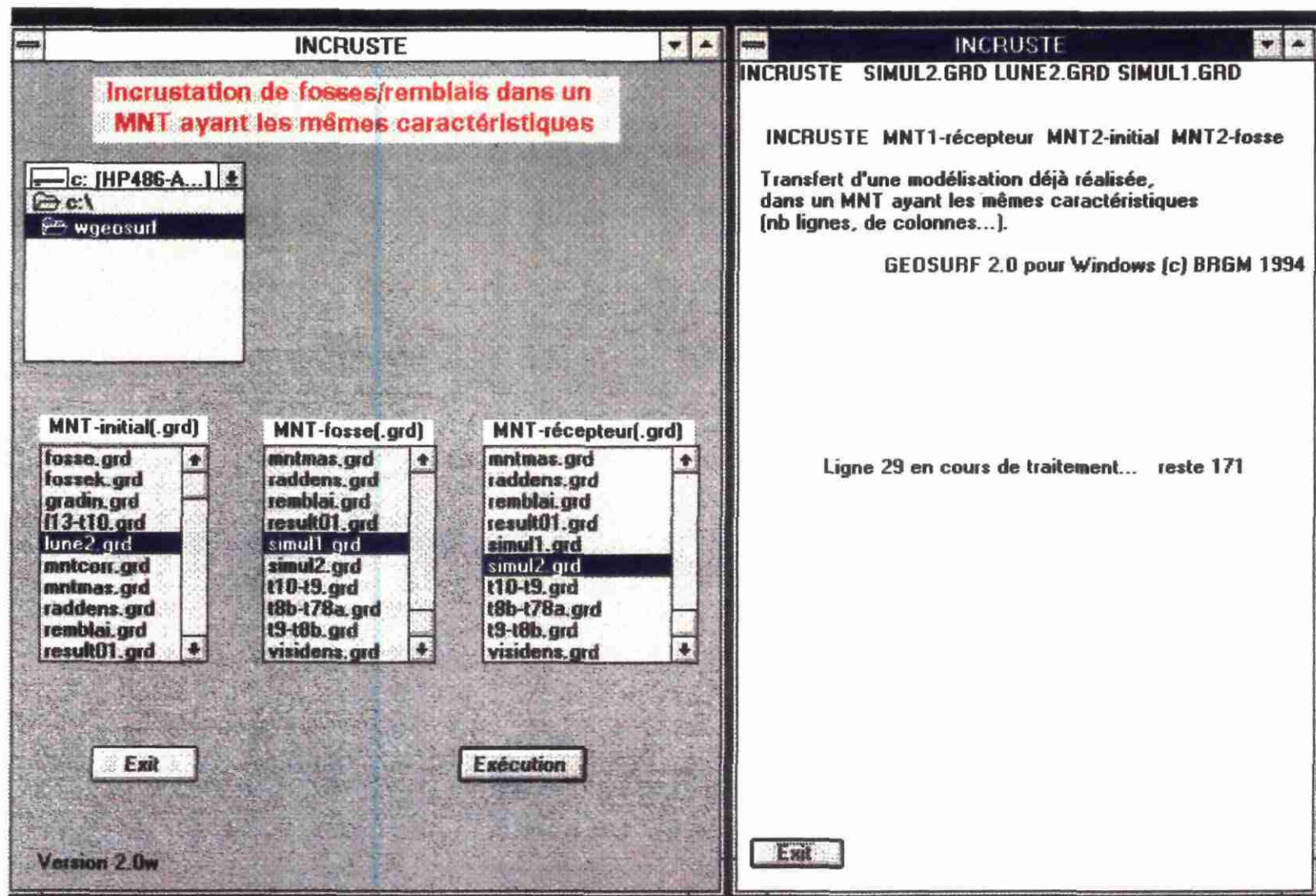
R 38098

CONSEIL REGIONAL PROVENCE-ALPES-COTE D'AZUR
CAO — Modélisation tridimensionnelle du sous-sol — Logiciel Geosurf

Planche 14

Utilitaire GEOSURF
Calcul des courbes d'iso-densité pour les zones visibles





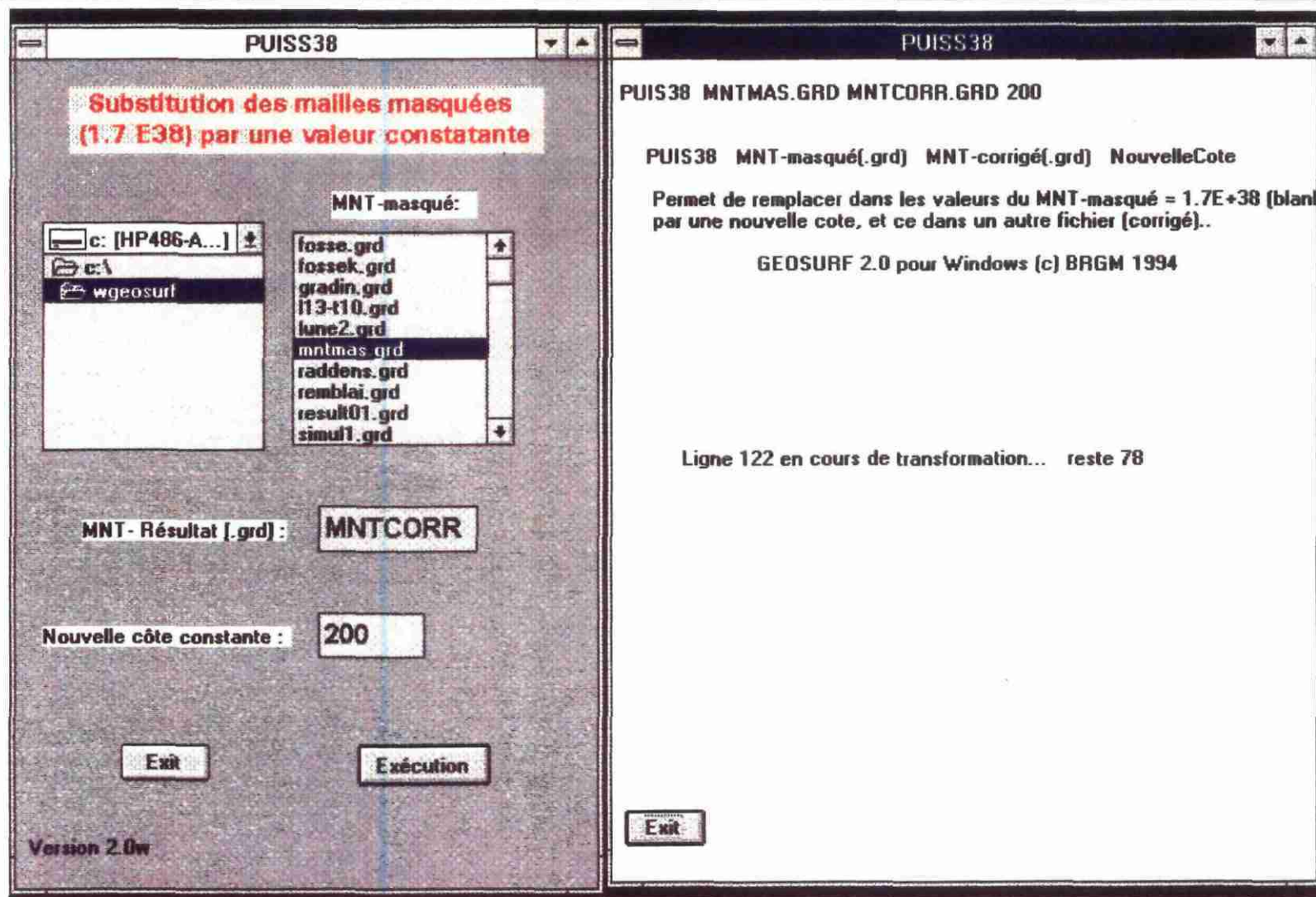
R 38098

CONSEIL REGIONAL PROVENCE-ALPES-COTE D'AZUR
CAO — Modélisation tridimensionnelle du sous-sol — Logiciel Geosurf

Planche 15

Utilitaire GEOSURF
Incrustation de plusieurs modélisations dans un même MNT





R 38098

CONSEIL REGIONAL PROVENCE-ALPES-COTE D'AZUR
CAO — Modélisation tridimensionnelle du sous-sol — Logiciel Geosurf

Planche 16

Utilitaire GEOSURF
Affectation d'une cote Z=constante à une zone masquée



GRADIN
▼ ▲

c: [HP486-A...]

**Modélisation de gradins
sur une fosse**

c:\

wgeosurf

MNT Résultat: [grd]

sim2grad

MNT initial

MNT contenant la fosse

fosse.grd fossek.grd gradin.grd l13-t10.grd lune2.grd mntcorr.grd mntmas.grd raddens.grd remblai.grd result01.grd	mntcorr.grd mntmas.grd raddens.grd remblai.grd result01.grd simul1.grd simul2.grd t10-t9.grd t8b-t78a.grd t9-t8b.grd
--	---

Cote du fond de la fosse (m)

120

Hauteur entre deux gradins (m)

15

Largeur des gradins (m)

10

Pente (en degrés)

35

Exit

Execution

Version 2.0w

Calculs en cours...
▼ ▲

GRADIN LUNE2.GRD SIMUL2.GRD SIM2GRAD 120 15 10 35

Syntaxe : GRADIN MNT-initial MNT-Fosse MNT-Gradin
Cotefond Hauteur-gradin Longueur-grading Pente

Modélise des gradins sur une fosse déjà calculée. La cote du fond de la fosse ainsi que la pente des talus doivent être précisées. La hauteur et la largeur des gradins sont exprimées en mètres.

GEOSURF 2.0 pour Windows (c) BRGM 1994

Ligne 34 en cours de calcul reste 166

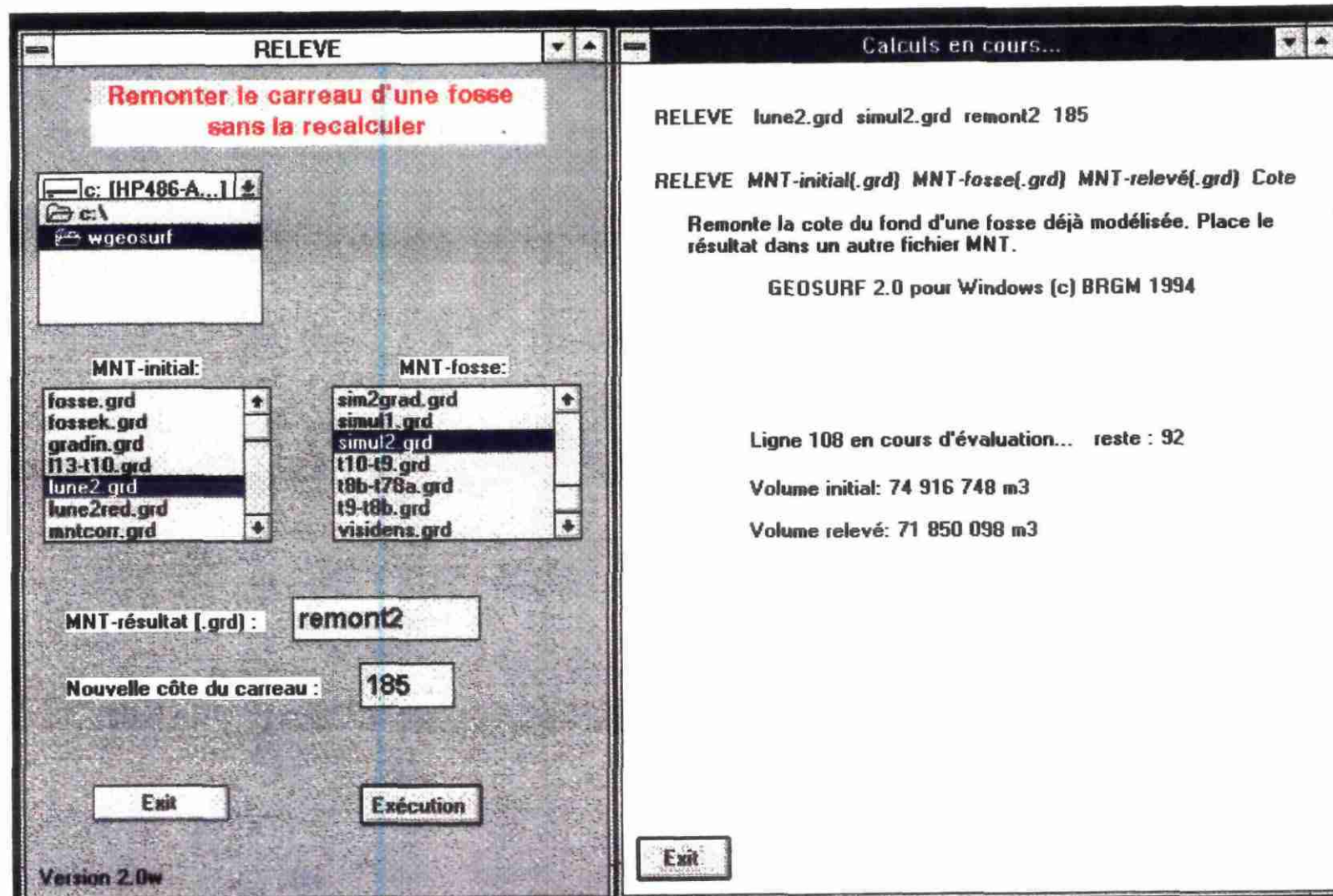
Exit

R 38098

CONSEIL REGIONAL PROVENCE-ALPES-COTE D'AZUR
CAO — Modélisation tridimensionnelle du sous-sol — Logiciel Geosurf

Planche 17

Utilitaire GEOSURF
Modélisation des gradins d'exploitation sur une fosse



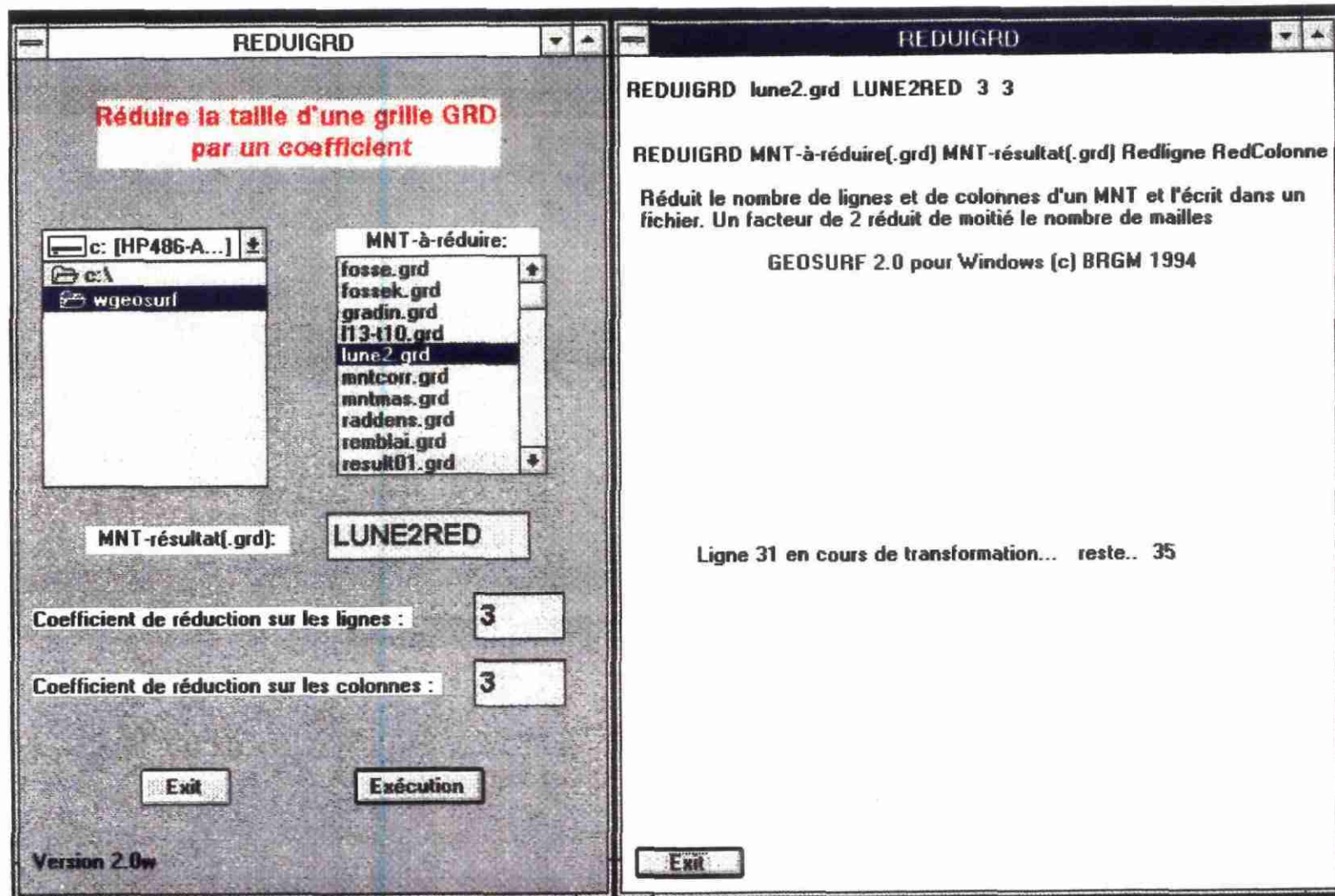
R 38098

CONSEIL REGIONAL PROVENCE-ALPES-COTE D'AZUR
CAO — Modélisation tridimensionnelle du sous-sol — Logiciel Geosurf

Planche 18

Utilitaire GEOSURF
Calcul des différentes phases d'exploitation par remonté du carreau de la fosse





R 38098

CONSEIL REGIONAL PROVENCE-ALPES-COTE D'AZUR
CAO — Modélisation tridimensionnelle du sous-sol — Logiciel Geosurf

Planche 19

Utilitaire GEOSURF
Réduction de la taille du MNT



ANNEXE 2

COMPARAISON DU CODAGE

VISUAL BASIC AVEC LE

BASIC PDS 7

VERSION 2.00

Begin Form Form2

```

BackColor = &H00808080&
Caption = "FOSSEBHV"
FillColor = &H00808080&
ForeColor = &H00808080&
Height = 7740
Left = 6630
LinkTopic = "Form2"
ScaleHeight = 7335
ScaleWidth = 5250
Top = 1095
Width = 5370

```

Begin CommandButton Command2

```

Caption = "exécution"
Default = -1 'True
Height = 390
Left = 3600
TabIndex = 1
Top = 6240
Width = 1350

```

End

Begin TextBox cote

```

BackColor = &H00C0C0C0&
FontBold = -1 'True
FontItalic = 0 'False
FontName = "Arial"
FontSize = 12
FontStrikethru = 0 'False
FontUnderline = 0 'False
Height = 420
Left = 4200
TabIndex = 9
Top = 2160
Width = 900

```

End

Begin TextBox result

```

BackColor = &H00C0C0C0&
FontBold = -1 'True
FontItalic = 0 'False
FontName = "Arial"
FontSize = 12
FontStrikethru = 0 'False
FontUnderline = 0 'False
Height = 420
Left = 3600
TabIndex = 6
Top = 1320
Width = 1575

```

End

Begin FileListBox fillist1

```

BackColor = &H00C0C0C0&
Height = 2175
Left = 2115
Pattern = "*.fos"
TabIndex = 5
Top = 3675
Width = 1740

```

End

Begin FileListBox fillist

```

BackColor = &H00C0C0C0&
Height = 2175
Left = 240
Pattern = "*.grd"
TabIndex = 4
Top = 3675
Width = 1740

```

End

Begin DirListBox dirlist

```

BackColor = &H00C0C0C0&
Height = 1830
Left = 75
TabIndex = 3
Top = 1365
Width = 2025

```

End

Begin DriveListBox drvlist

```

BackColor = &H00FFFFFF&
Height = 315

```

Visual Basic

Calcul d'une fosse du bas vers le haut à pente variable

c: [HP486-A...] |

c:\

- vbasic
- geosuit
- blanked
- cartegeo
- cubegeo
- densite
- fossebhk

MNT résultat :

Cote du carreau (m) :

Version 2.0w

```
Sub Command1_Click ()
    End
End Sub

Sub Command2_Click ()
    fossebhv
End Sub

Sub cote_Change ()
    If IsNumeric(cote.Text) Then
        zfond! = Val(cote.Text)
    Else
        If cote.Text <> "" Then
            MsgBox "not numeric"
        End If
    End If
End Sub

End Sub

Sub dirlist_Change ()
    filList.Path = dirlist.Path
    filList1.Path = dirlist.Path
    ChDir dirlist.Path
End Sub

Sub drvlist_Change ()
    dirlist.Path = drvlist.Drive
    ChDrive drvlist.Drive
End Sub

Sub filllist_Click ()
    nom1$ = UCase(filList.FileName)
End Sub

Sub filllist1_Click ()
    nom2$ = UCase(filList1.FileName)
End Sub

Sub result_Change ()
    nom3$ = UCase$(Result.Text)
End Sub
```

```
Left = 75
TabIndex = 2
Top = 1080
Width = 2025
End
Begin CommandButton Command1
Caption = "exit"
Height = 375
Left = 720
TabIndex = 0
Top = 6240
Width = 1065
End
Begin Label Label5
BackColor = &H00808080&
Caption = "Version 2.0w"
Height = 255
Left = 120
TabIndex = 11
Top = 6960
Width = 1215
End
Begin Label Label4
Alignment = 2 'Center
BackColor = &H00C0C0C0&
Caption = "Calcul d'une fosse du bas vers le haut à pente variable"
FontBold = -1 'True
FontItalic = 0 'False
FontName = "Univers"
FontSize = 12
FontStrikethru = 0 'False
FontUnderline = 0 'False
ForeColor = &H000000FF&
Height = 615
Left = 960
TabIndex = 10
Top = 240
Width = 4095
End
Begin Label Label3
AutoSize = -1 'True
BackColor = &H00FFFFFF&
Caption = "Cote du carreau (m) :"
Height = 195
Left = 2280
TabIndex = 8
Top = 2280
Width = 1815
End
Begin Label Label1
AutoSize = -1 'True
BackColor = &H00FFFFFF&
Caption = "MNT résultat :"
Height = 195
Left = 2280
TabIndex = 7
Top = 1440
Width = 1230
End
End
```

WFOSSBHV.FRM - 1

VERSION 2.00

Begin Form Form1

Caption = "FOSSBHV"
Height = 6525
Icon = (Icon)
Left = 15
LinkTopic = "Form1"
ScaleHeight = 6120
ScaleWidth = 6390
Top = 1065
Width = 6510

Begin CommandButton Command2

Caption = "exit"
Height = 375
Left = 5160
TabIndex = 0
Top = 5640
Width = 1185

End

End



VERSION 2.00

Begin Form Form3

```
AutoRedraw      = -1 'True
BackColor       = &H00C0C0C0&
BorderStyle     = 1 'Fixed Single
Caption        = "Calculs en cours"
ControlBox     = 0 'False
FillColor      = &H00C0C0C0&
ForeColor      = &H00C0C0C0&
Height         = 1185
Left           = 60
LinkTopic      = "Form3"
MaxButton      = 0 'False
MinButton      = 0 'False
ScaleHeight    = 780
ScaleWidth     = 6390
Top            = 1110
Width          = 6510
```

Begin Label Label3

```
AutoSize       = -1 'True
BackColor      = &H00C0C0C0&
ForeColor      = &H000000000&
Height         = 240
Left           = 1248
TabIndex       = 2
Top            = 432
Width          = 72
```

End

Begin Label Label2

```
AutoSize       = -1 'True
BackColor      = &H00C0C0C0&
ForeColor      = &H000000000&
Height         = 240
Left           = 3072
TabIndex       = 1
Top            = 48
Width          = 72
```

End

Begin Label Label1

```
AutoSize       = -1 'True
BackColor      = &H00C0C0C0&
ForeColor      = &H000000000&
Height         = 240
Left           = 624
TabIndex       = 0
Top            = 48
Width          = 72
```

End

End

```

Global xmi!, xma!, ymi!, yma!, zmin#, zmax#, pente, zfond!
Global minx!, maxx!, miny!, maxy!, nv%, f$
Global matx!(1000), maty!(1000), pent1(1000)
Global nom1$, nom2$, nom3$
Global commande$(4)
Global vx(2), vy(2)
Global char$
Declare Sub hmemcpy Lib "kernel" (hpvdest As Any, hpvsource As Any, ByVal cbcopy As Long)
Declare Function floodfill% Lib "GDI" (ByVal hdc%, ByVal x%, ByVal y%, ByVal crcolor%)
Declare Function getdc% Lib "user" (ByVal hwnd%)
Declare Function getactivewindow% Lib "user" ()
Declare Function createsolidbrush% Lib "GDI" (ByVal crcolor%)
Declare Function DeleteObject% Lib "GDI" (ByVal hObject%)
Declare Function selectobject% Lib "GDI" (ByVal hdc%, ByVal hObject%)

Function airContour (matx!(), maty!(), n)
    matx!(n + 1) = matx!(1)
    maty!(n + 1) = maty!(1)
    air = 0
    For i = 1 To n
        air = air + (matx!(i) + matx!(i + 1)) * (maty!(i) - maty!(i + 1))
    Next i
    airContour = Abs(air) / 2
End Function

Function chargerContour (nom$, matx!(), maty!(), pent1(), minx!, maxx!, miny!, maxy!, nv%)
    pi = 3.141592
    f$ = nom$
    Open nom$ For Input As 1
        minx = 1E+30
        maxx = -1E+30
        miny = 1E+30
        maxy = -1E+30
        Input #1, nv
        res = 1
        For i = 1 To nv
            Input #1, matx(i), maty(i), pent1(i)
            If matx(i) <= xmi Or matx(i) >= yma Or maty(i) <= ymi Or maty(i) >= yma Then
                res = -1
            End If
            If pent1(i) < 90 Then
                pent1(i) = Tan(pent1(i) / 180 * pi)
            Else
                pent1(i) = 10000000000#
            End If
            pent1(i) = pent1(i) * 1000
            If matx(i) > maxx Then maxx = matx(i)
            If matx(i) < minx Then minx = matx(i)
            If maty(i) > maxy Then maxy = maty(i)
            If maty(i) < miny Then miny = maty(i)
        Next
        If minx < xmi Then minx = xmi
        If maxx > xma Then maxx = xma
        If miny < ymi Then miny = ymi
        If maxy > yma Then maxy = yma

        Close (1)
        chargerContour = res
    End Function

Function chargerMaille (src, xmi, xma, ymi, yma, zmin, zmax, nl, nc, dest)
    a$ = Input$(4, src)
    If a$ <> "DSBB" Then
        res = -1
    Else
        nc = cvi(Input$(2, src))
        nl = cvi(Input$(2, src))
        xmi = cvd(Input$(8, src))
        xma = cvd(Input$(8, src))
        ymi = cvd(Input$(8, src))
        yma = cvd(Input$(8, src))
        zmin = cvd(Input$(8, src))
        zmax = cvd(Input$(8, src))
        res = 0
    End If
    chargerMaille = res

```

End Function

```
Function couleur (Z!)
    couleur = ((Z! - zmin#) / (zmax# - zmin#) * 14) Mod 14 + 2
End Function
```

```
Sub creerBln (matx!(), maty!(), nv%, nom$)
    f$ = nom$
    Open nom$ For Output As #5
    Print #5, nv
    For i = 1 To nv
        Print #5, matx(i), maty(i)
    Next i
    Close #5
End Sub
```

```
Function cvd (x As String) As Double
    If Len(x) <> 8 Then
        MsgBox "illégalle fonction dans CVD"
        Stop
    End If
    hmemcpy temp#, ByVal x, 8
    cvd = temp#
End Function
```

```
Function cvi (x As String) As Integer
    If Len(x) <> 2 Then
        MsgBox "illégalle fonction dans CVi"
        Stop
    End If
    hmemcpy temp%, ByVal x, 2
    cvi = temp%
End Function
```

```
Function cvs (x As String) As Single
    If Len(x) <> 4 Then
        MsgBox "illégalle fonction dans CVs"
        Stop
    End If
    hmemcpy temp!, ByVal x, 4
    cvs = temp!
End Function
```

```
Sub decode (nbargument, commande$())
    nom$ = Command$
    nom$ = LTrim$(RTrim$(nom$))
    For i = 1 To nbargument - 1
        If nom$ <> "" Then
            v = InStr(nom$, " ")
            If v = 0 Then
                syntaxe
                End
            End If
            commande$(i) = Left$(nom$, v - 1)
            nom$ = LTrim$(Right$(nom$, Len(nom$) - v))
        Else
            syntaxe
            End
        End If
    Next i
    If nom$ <> "" Then
        commande$(nbargument) = LTrim$(RTrim$(nom$))
    Else
        syntaxe
        End
    End If
End Sub
```

End Sub

```
Function droite (x1, y1, x2, y2, angle)
    pi = 3.141592
    If x1 <> x2 And y1 <> y2 Then
        a = (y1 - y2) / (x1 - x2)
        angle = Abs(Atn(a))
        If x1 < x2 Then
            If y1 < y2 Then
                angle = angle
            Else
                angle = 360 - angle
            End If
        End If
    End If
End Function
```

```

        angle = 2 * pi - angle
    End If
Else
    If y1 < y2 Then
        angle = pi - angle
    Else
        angle = pi + angle
    End If
End If
End If
droite = a
End Function

Sub ErrMess (n, f$)
    Select Case n
        Case 4
            message$ = "Données épuisées."
        Case 5
            message$ = "Appel de fonction interdit."
        Case 6
            message$ = "Dépassement de capacité."
        Case 7
            message$ = "Mémoire insuffisante."
        Case 9
            message$ = "Indice en dehors des limites pour un tableau."
        Case 11
            message$ = "Une division par zéro a été demandée."
        Case 24
            message$ = "Délai d'attente d'un périphérique écoulé."
        Case 25
            message$ = "Défaillance du périphérique."
        Case 27
            message$ = "Attente chargement du papier."
        Case 51
            message$ = "Erreur interne."
        Case 52
            message$ = "Nom ou numéro de fichier incorrect : " + UCase$(f$)
        Case 53
            message$ = "Le fichier spécifié n'a pas été trouvé : " + UCase$(f$)
        Case 54
            message$ = "Mode d'accès au fichier incorrect : " + UCase$(f$)
        Case 55
            message$ = "Fichier déjà ouvert : " + UCase$(f$)
        Case 57
            message$ = "Erreur d'entrée / sortie sur le périphérique."
        Case 58
            message$ = "Fichier qui existe déjà : " + UCase$(f$)
        Case 61
            message$ = "Disque saturé."
        Case 62
            message$ = "Entrée hors des limites d'un fichier : " + UCase$(f$)
        Case 64
            message$ = "Nom de fichier incorrect : " + UCase$(f$)
        Case 67
            message$ = "Trop de fichiers ouvert dans le répertoire."
        Case 68
            message$ = "Périphérique non disponible."
        Case 69
            message$ = "Dépassement de capacité du tampon de communication."
        Case 70
            message$ = "Accès refusé : " + UCase$(f$)
        Case 71
            message$ = "Disque nom prêt."
        Case 72
            message$ = "Disque défectueux."
        Case 75
            message$ = "Erreur de chemin d'accès/d'accès au fichier."
        Case 76
            message$ = "Chemin d'accès introuvable."
        Case Else
            message$ = "Erreur inconnue..."
    End Select
    form1.Scale
    form1.Cls
    form1.Command2.Caption = "OK"
    form1.AutoRedraw = True
    form1.Hide
    form1.Print "FOSSEBHV " + nom1$ + nom2$ + nom3$ & zfond!
    form1.Print "Une erreur d'exécution a été détectée lors du déroulement du programme."

```

```

form1.Print "Code de l'erreur : " & Err
form1.Print "Signification      : " + message$
form1.Print "Ecriture dans le fichier PACA-ERR"
form1.Print "Références         : " + " le " & Date$ + " à " & Time$
free = FreeFile

Open "paca-err" For Append As free
Print #free, "-----"
Print #free, "FOSSEBHV  " + nom1$ + nom2$ + nom3$ & zfond!
Print #free, "Code de l'erreur : "; Err
Print #free, "Signification   : "; message$
Print #free, "Références      : "; "le " + Date$ + " à " + Time$
Close free
form1.Show 1

End
End Sub

Sub ErrProgram (n, f$)
Select Case n
Case 1
message$ = "Le contour doit être dans le MNT."
Case 2
message$ = "Le pente doit être positive ."
Case 3
message$ = "Le fichier " + f$ + " n'est pas un MNT ."

Case Else
message$ = "Erreur inconnue..."
End Select
form1.Scale
form1.Cls
form1.Command2.Caption = "OK"
form1.AutoRedraw = True
form1.Hide
form1.Print "      FOSSEBHV  " + nom1$ + nom2$ + nom3$ & zfond!
form1.Print "Code de l'erreur : " & n
form1.Print "Signification   : "; message$
form1.Print "Ecriture dans le fichier PACA-ERR"
form1.Print "Références      : "; "le " + Date$ + " à " + Time$
free = FreeFile
Open "paca-err" For Append As free
Print #free, "-----"
Print #free, "FOSSEBHV  " + Command$
Print #free, "Code de l'erreur : "; n
Print #free, "Signification   : "; message$
Print #free, "Références      : "; "le " + Date$ + " à " + Time$
Close free
form1.Show 1
End
End Sub

Function ext$ (sett$, sym$)
sym$ = "." + sym$
If InStr(1, sett$, ".") = 0 Then sett$ = sett$ + sym$
ext$ = UCase$(sett$)
End Function

Sub fossebhv ()
form3.Show
vx(1) = 1: vx(2) = 638: vy(1) = 16: vy(2) = 348
pi = 3.141592
contour% = 15

pospt = InStr(1, nom2$, ".")
If pospt <> 0 Then
bln$ = Mid$(nom2$, 1, pospt - 1) + ".bln"
Else
bln$ = nom2$ + ".bln"
End If

nom1$ = ext$(nom1$, "grd")
nom2$ = ext$(nom2$, "fos")
nom3$ = ext$(nom3$, "grd")
form3.Label2.Caption = "MNT Resultat: " + nom3$
form1.Cls
trace

```

```

src = 2
dest = 3
f$ = nom1$

Open nom1$ For Binary As src
If chargerMaille(src, xmi, xma, ymi, yma, zmin, zmax, nl, nc, dest) = -1 Then
    ErrProgram 3, nom1$: End
    form1.Print "le fichier "; nom1$; " n'est pas un mnt": End
End If
Close #src
FileCopy nom1$, nom3$
f$ = nom1$
Open nom1$ For Binary As src
f$ = nom3$
Open nom3$ For Binary As dest
epsilon = (xma - xmi) / 1000
eps = 1 * E - 10
tm = (xma - xmi) / (nc - 1) / 2
If chargerContour(nom2$, matx(), maty(), pent1(), minx, maxx, miny, maxy, nv) = -1 Then
    ErrProgram 1, ""
End
End If

pointintern matx!(), maty!(), nv, minx!, maxx!, miny!, maxy!, xp!, yp!
creerBln matx!(), maty!(), nv, bln$
If zmin > zfond! Then zmin = zfond!
Seek #dest, 10 * 4 + 1
Put #dest, , zmin#
nv = nv - 1
form1.Show
form1.Line (0, 0)-(1, 1), QBColor(contour%), B
delta = (xma! - xmi!) / 50
form1.Scale (xmi, yma)-(xma, ymi)
hdc% = getdc%(getactivewindow())
hbrush = createsolidbrush(QBColor(2))
koko = selectobject(hdc%, hbrush)
bk_color = floodfill(hdc%, 0, 0, QBColor(1))

For i = 1 To nv
    If matx!(i) = matx!(i + 1) Then matx!(i + 1) = matx(i + 1) + epsilon
    If maty!(i) = maty!(i + 1) Then maty!(i + 1) = maty(i + 1) + epsilon
    form1.Line (matx!(i), maty!(i))-(matx!(i + 1), maty!(i + 1)), QBColor(1)
Next i
del_st = DeleteObject(hdc%)
hbrush = createsolidbrush(QBColor(7))
koko = selectobject(hdc%, hbrush)
bk_color = floodfill(hdc%, 0, 0, QBColor(1))
c1 = Int((minx! - xmi!) / (xma! - xmi!) * (nc - 1) + 1.5)
c2 = Int((maxx! - xmi!) / (xma! - xmi!) * (nc - 1) + 1.5)
l1 = Int((miny! - ymi!) / (yma! - ymi!) * (nl - 1) + 1.5)
l2 = Int((maxy! - ymi!) / (yma! - ymi!) * (nl - 1) + 1.5)
ReDim t(l1 - 2 To l2 + 2, c1 - 2 To c2 + 2)
ReDim t1(l1 - 2 To l2 + 2, c1 - 2 To c2 + 2)
ReDim t2(l1 - 2 To l2 + 2, c1 - 2 To c2 + 2)
insurface = 1
For l = l1 To l2
    For c = c1 To c2
        Seek #src, (l - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
        Seek #dest, (l - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
        lez$ = String$(4, " ")
        Get #src, , lez$
        m! = cvs(lez$)
        x = (c - 1) * (xma - xmi) / (nc - 1) + xmi
        y = (l - 1) * (yma - ymi) / (nl - 1) + ymi
        v = form1.Point(x, y)
        If v = 8388608 Or v = 32768 Then
            If v = 32768 Then
                t2(l, c) = -1E+15
                form1.PSet (x, y), QBColor(15)
            End If
            If m! < zfond! Then
                insurface = 0
                Put #dest, , m!
            Else
                Put #dest, , zfond!
            End If
        Else
            t2(l, c) = -2E+15
        End If
    Next c
Next l

```

```

        Put #dest, , m!
        form1.PSet (x, y), QBColor(7)
    End If
Next c
Next l
col = couleur(zfond!)
del_st = DeleteObject(hdc%)
hbrush = createsolidbrush(QBColor(col))
koko = selectobject(hdc%, hbrush)
bk_color = floodfill(hdc%, 0, 0, QBColor(8))

If insurface = 1 Then
    For i = 1 To nv
        form1.Line (matx(i), maty(i))-(matx(i + 1), maty(i + 1)), QBColor(1)
    Next i
    del_st = DeleteObject(hdc%)
    hbrush = createsolidbrush(QBColor(0))
    koko = selectobject(hdc%, hbrush)
    bk_color = floodfill(hdc%, 0, 0, QBColor(1))
End If
matx(nv + 1) = matx(1)
maty(nv + 1) = maty(1)
pent1(nv + 1) = pent1(1)
matx(nv + 2) = matx(2)
maty(nv + 2) = maty(2)
pent1(nv + 2) = pent1(2)
minxp = 1E+30
maxxp = -1E+30
minyp = 1E+30
maxyp = -1E+30

For v = 1 To nv
    DoEvents
    form3.Label1.Caption = Format$(v, " ###") + " / " + Format$(nv, "###")
    x1 = matx(v)
    y1 = maty(v)
    x2 = matx(v + 1)
    y2 = maty(v + 1)
    pente = pent1(v + 1)
    a = droite(x1, y1, x2, y2, angle)
    B = y1 - a * x1
    pas1 = Abs(Cos(angle) * tm)
    pas2 = Abs(Sin(angle) * tm)
    If x1 > x2 Then pas1 = -pas1
    If y1 > y2 Then pas2 = -pas2
    For x = x1 - pas1 To x2 + pas1 Step pas1
        y = a * x + B
        a2 = -1 / a
        b2 = y + x / a
        If insurface = 0 Then
            xx = x
            xx = xx + pas2
            yy = a2 * xx + b2
            c = Int((xx - xmi) / (xma - xmi) * (nc - 1) + 1.5)
            l = Int((yy - ymi) / (yma - ymi) * (nl - 1) + 1.5)
            compt = 1
            While t2(1, c) <> 0 Or compt = 1
                If t2(1, c) = -1E+15 Then
                    Seek #dest, (1 - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
                    Get #dest, , lez$
                    m! = cvs(lez$)
                    col = couleur(m!)
                    xa = (c - 1.5) * (xma - xmi) / (nc - 1) + xmi
                    ya = (l - 1.5) * (yma - ymi) / (nl - 1) + ymi
                    xb = (c - .5) * (xma - xmi) / (nc - 1) + xmi
                    yb = (l - .5) * (yma - ymi) / (nl - 1) + ymi
                    form1.Line (xa, ya)-(xb, yb), QBColor(col), BF
                End If
                xx = xx + pas2
                yy = a2 * xx + b2
                zz = zfond!
                c = Int((xx - xmi) / (xma - xmi) * (nc - 1) + 1.5)
                l = Int((yy - ymi) / (yma - ymi) * (nl - 1) + 1.5)
                compt = compt + 1
            Wend
            DoEvents
        End If
        xx = x + pas2
    Next x

```

```

yy = a2 * xx + b2
c = Int((xx - xmi) / (xma - xmi) * (nc - 1) + 1.5)
l = Int((yy - ymi) / (yma - ymi) * (nl - 1) + 1.5)
Seek #src, (1 - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
Get #src, , lez$
Z! = cvs(lez$)
z1! = pente * Sqr((xx - x) * (xx - x) + (yy - y) * (yy - y))
zz = Z! - z1! - zfond!
compt = 1
Do While zz > 0 Or compt <= 2
  compt = compt + 1
  Seek #dest, (1 - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
  Get #dest, , lez$
  z2! = cvs(lez$)
  If z2! > z1! + zfond! Then
    ztemp! = z1! + zfond!
  Else
    ztemp! = z2!
  End If
  Seek #dest, (1 - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
  Put #dest, , ztemp!
  col = couleur(ztemp!)
  xa = (c - 1.5) * (xma - xmi) / (nc - 1) + xmi
  ya = (1 - 1.5) * (yma - ymi) / (nl - 1) + ymi
  xb = (c - .5) * (xma - xmi) / (nc - 1) + xmi
  yb = (1 - .5) * (yma - ymi) / (nl - 1) + ymi
  form1.Line (xa, ya)-(xb, yb), QBColor(col), BF
  xx = xx - pas2
  yy = a2 * xx + b2
  c = Int((xx - xmi) / (xma - xmi) * (nc - 1) + 1.5)
  l = Int((yy - ymi) / (yma - ymi) * (nl - 1) + 1.5)
  If xx <= xmi Or xx >= xma Or yy <= ymi Or yy > yma Then Exit Do
  Seek #src, (1 - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
  Get #src, , lez$
  Z! = cvs(lez$)
  z1! = pente * Sqr((xx - x) * (xx - x) + (yy - y) * (yy - y))
  zz = Z! - z1! - zfond!
  compt = compt + 1
Loop
If xx > maxxp Then maxxp = xx
If xx < minxp Then minxp = xx
If yy > maxyp Then maxyp = yy
If yy < minyp Then minyp = yy
DoEvents
Next x
If insurface = 0 Then
  For a = 0 To 2 * pi Step pi / 180
    c = Int((x2 - xmi) / (xma - xmi) * (nc - 1) + 1.5)
    l = Int((y2 - ymi) / (yma - ymi) * (nl - 1) + 1.5)
    r = 0
    r = r + tm
    x = x2 + Cos(a) * r
    y = y2 + Sin(a) * r
    compt = 1
    While t2(l, c) <> 0 Or compt = 1
      If t2(l, c) = -1E+15 Then
        Seek #dest, (1 - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
        Get #dest, , lez$
        m! = cvs(lez$)
        col = couleur(m!)
        xa = (c - 1.5) * (xma - xmi) / (nc - 1) + xmi
        ya = (1 - 1.5) * (yma - ymi) / (nl - 1) + ymi
        xb = (c - .5) * (xma - xmi) / (nc - 1) + xmi
        yb = (1 - .5) * (yma - ymi) / (nl - 1) + ymi
        form1.Line (xa, ya)-(xb, yb), QBColor(col), BF
      End If
      r = r + tm
      x = x2 + Cos(a) * r
      y = y2 + Sin(a) * r
      zz = zfond!
      c = Int((x - xmi) / (xma - xmi) * (nc - 1) + 1.5)
      l = Int((y - ymi) / (yma - ymi) * (nl - 1) + 1.5)
      compt = compt + 1
    DoEvents
  Wend
Next a
End If
angle1 = preAngle(v + 1)
angle2 = postAngle(v + 1)

```

```

If pi / 2 >= angle1 And angle2 > 3 * pi / 2 Then
  angle2 = angle2 - 2 * pi
End If
pentel = pent1(v + 1)
pente2 = pent1(v + 2)
For a = angle1 To (angle2 - pi / 180) Step -pi / 180
  alpha1 = (angle1 - a) / (angle1 - angle2)
  alpha2 = (a - angle2) / (angle1 - angle2)
  pente = (alpha2 * pentel + alpha1 * pente2)
  r = 0
  x = x2 + Cos(a) * r
  y = y2 + Sin(a) * r
  c = Int((x - xmi) / (xma - xmi) * (nc - 1) + 1.5)
  l = Int((y - ymi) / (yma - ymi) * (nl - 1) + 1.5)
  Seek #src, (1 - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
  Get #src, , lez$
  Z! = cvs(lez$)
  z1! = pente * r
  zz = Z! - z1! - zfond!
  compt = 1
  Do While zz > 0 Or compt <= 2
    compt = compt + 1
    Seek #dest, (1 - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
    Get #dest, , lez$
    z2! = cvs(lez$)
    If z2! > z1! + zfond! Then
      ztemp! = z1! + zfond!
    Else
      ztemp! = z2!
    End If
    Seek #dest, (1 - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
    Put #dest, , ztemp!
    col = couleur(ztemp!)
    xa = (c - 1.5) * (xma - xmi) / (nc - 1) + xmi
    ya = (l - 1.5) * (yma - ymi) / (nl - 1) + ymi
    xb = (c - .5) * (xma - xmi) / (nc - 1) + xmi
    yb = (l - .5) * (yma - ymi) / (nl - 1) + ymi
    form1.Line (xa, ya)-(xb, yb), QBColor(col), BF
    r = r + tm
    x = x2 + Cos(a) * r
    y = y2 + Sin(a) * r
    c = Int((x - xmi) / (xma - xmi) * (nc - 1) + 1.5)
    l = Int((y - ymi) / (yma - ymi) * (nl - 1) + 1.5)
    If x <= xmin Or x >= xma Or y <= ymi Or y > yma Then Exit Do
    Seek #src, (1 - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
    Get #src, , lez$
    Z! = cvs(lez$)
    z1! = pente * r
    zz = Z! - z1! - zfond!
  DoEvents
  Loop
  If xx > maxxp Then maxxp = xx
  If xx < minxp Then minxp = xx
  If yy > maxyp Then maxyp = yy
  If yy < minyp Then minyp = yy
Next a
rem1:
If maxxp > xma Then maxxp = xma
If minxp < xmi Then minxp = xmi
If maxyp > yma Then maxyp = yma
If minyp < ymi Then minyp = ymi

Next v
For i = 1 To nv
  form1.Line (matx(i), maty(i))-(matx(i + 1), maty(i + 1)), QBColor(1)
Next i
c1 = Int((minxp - xmi) / (xma - xmi) * (nc - 1) + 1.5)
c2 = Int((maxxp - xmi) / (xma - xmi) * (nc - 1) + 1.5)
l1 = Int((minyp - ymi) / (yma - ymi) * (nl - 1) + 1.5)
l2 = Int((maxyp - ymi) / (yma - ymi) * (nl - 1) + 1.5)
vol! = 0
rap! = (xma - xmi) / (nc - 1) * 1000 * (yma - ymi) * 1000 / (nl - 1)
If l1 <= 1 Then l1 = 2
If l2 >= nl Then l2 = nl - 1
If c1 <= 1 Then c1 = 2
If c2 > nc Then c2 = nc - 1
For l = 1 To l1
  Next l
For l = l1 To l2

```

```

For c = c1 To c2
    Seek #src, (1 - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
    Get #src, , lez$
    m1 = cvs(lez$)
    Seek #dest, (1 - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
    Get #dest, , lez$
    m2 = cvs(lez$)
    vol! = vol! + (m1 - m2)
Next c
form3.Label3.Caption = "Volume extrait: " + Format$(vol! * rap!, "#####,# m3")
Next l
For l = 12 To n1
Next l
Close #src
Close #dest
Beep
Close
cr = Chr$(13)
form1.Scale
form1.Cls
form1.Command2.Caption = "OK"
form1.AutoRedraw = True
form1.Hide
form1.Print " FOSSEBHV MNT-initial(.grd) Contour(.fos) MNT-fosse(.grd) Z-Fond"
form1.Print : form1.Print
form1.Print " Le contour inférieur de la fosse est connu."
mes$ = mes$ + cr
mes$ = mes$ + " L'angle par rapport à l'horizontale est contenu dans le fichier"
mes$ = mes$ + cr
mes$ = mes$ + " contour. L'angle est exprimé en degrés, la cote en mètres."
mes$ = mes$ + cr + cr + Space$(15) + "GEOSURF 1.0 (c) BRGM 4S/PAC 1992"

form1.Print mes$
form1.Print : form1.Print
form1.Print " FOSSEBHV " + nom1$ + nom2$ + nom3$ & zfond!
form1.Print
form1.Print : form1.Print

form1.Print " Surface du fond de la fosse : " + Format$(airContour(matx!(), maty!(), nv) * 100
0000, "#####,#m2")
form1.Print
form1.Print " Volume extrait : " + Format$(vol! * rap!, "#####,#m3")
form1.Show 1
volume = Format$(vol! * rap!, "#####,### m3")
MsgBox "volume = " & volume

End
erreur:
ErrMess Err, f$

End Sub

Function intern (x, y, matx!(), maty!(), n)
    i = 1
    While x <> matx(i) And y <> maty(i) And i <= (n - 1)
        i = i + 1
    Wend
    If i >= n Then
        For i = 1 To n - 1
            If matx(i) < matx(i + 1) Then
                minix = matx(i)
                maxix = matx(i + 1)
            Else
                minix = matx(i + 1)
                maxix = matx(i)
            End If
            If maty(i) < maty(i + 1) Then
                miniy = maty(i)
                maxiy = maty(i + 1)
            Else
                miniy = maty(i + 1)
                maxiy = maty(i)
            End If
            If matx(i + 1) <> matx(i) Then
                a = (maty(i + 1) - maty(i)) / (matx(i + 1) - matx(i))
                B = maty(i) - a * matx(i)
                If a <> 0 Then
                    xx = (y - B) / a
                Else

```

```

        xx = matx(i)
    End If
    YY = Y
End If
If xx < x Then
    If (minix < xx) And (xx < maxix) And (miniy < yy) And (yy < maxiy) Then
        cpt = cpt + 1
    End If
End If
Next
End If
intern = cpt Mod 2
End Function

```

```

Sub pointintern (matx!(), maty!(), n%, minx!, maxx!, miny!, maxy!, xp!, yp!)
    cpt = 0
    deltax = (maxx - minx) / (n - 2)
    deltax = (maxy - miny) / (n - 2)
    yy = miny - deltax
    While (cpt <> 1) And (yy <= maxy)
        xx = minx
        yy = yy + deltax
        While (xx <= maxx) And (cpt <> 1)
            cpt = intern(xx, yy, matx!(), maty!(), n)
            If cpt = 1 Then
                xp = xx
                yp = yy
            End If
            xx = xx + deltax
        Wend
        yy = yy + deltax
    Wend
End Sub

```

```
End Sub
```

```

Function postAngle (v)
    pi = 3.141592
    a = droite(matx(v), maty(v), matx(v + 1), maty(v + 1), angle)
    angle = angle + pi / 2
    If angle > 2 * pi Then
        angle = angle - 2 * pi
    End If
    postAngle = angle
End Function

```

```
End Function
```

```

Function preAngle (v)
    pi = 3.141592
    a = droite(matx(v - 1), maty(v - 1), matx(v), maty(v), angle)
    angle = angle + pi / 2
    If angle > 2 * pi Then
        angle = angle - 2 * pi
    End If
    preAngle = angle
End Function

```

```

Sub syntaxe ()
    form1.Print : form1.Print
    form1.Scale
    form1.Cls
    form1.Command2.Caption = "OK"
    form1.AutoRedraw = True
    form1.Hide

    Beep
    mes$ = " Syntaxe : " + Chr$(13) + Chr$(13) + " FOSSEBHV MNT-initial(.grd) Contour(
.fos) MNT-fosse(.grd) Z-Fond"

    mes$ = mes$ + Chr$(13) + Chr$(13)
    mes$ = mes$ + " Le contour inférieur de la fosse est connu."
    mes$ = mes$ + Chr$(13)
    mes$ = mes$ + " L'angle par rapport à l'horizontale est contenu dans le fichi
er"
    mes$ = mes$ + Chr$(13)
    mes$ = mes$ + " contour. L'angle est exprimé en degrés, la cote en mètres."
    mes$ = mes$ + Chr$(13) + Chr$(13) + Space$(22) + "GEOSURF 1.0 (c) BRGM 4S/PAC 1992"
    form1.Print mes$
    form1.Print
    form1.Show 1

```

End

End Sub

Sub trace ()

f\$ = "log"

Open "log" For Append As 66

Print #66, Date\$ + " - " + Time\$; " - FOSSEBHV " + nom1\$ + nom2\$ + nom3\$ & zfond!

Close 66

End Sub

```

'-----
'-----
'----- Programme : FOSSEBHV.bas -----
'----- Date de mise à jour : décembre 1993 -----
'----- Opérateur : Alain Bourrousse -----
'----- Numéro de la version : 1.01a version MSDOS -----
'-----
'-----
DECLARE SUB syntaxe ()
      DECLARE FUNCTION ext$(set$, sym$)
DECLARE FUNCTION chargerMaille! (src!, xmi!, xma!, ymi!, yma!, zmin!, zmax!, nl!, nc!, dest!)
DECLARE FUNCTION couleur! (z!)
DECLARE SUB creerBln (x!(), y!(), nv!, nom$)
DECLARE SUB decode (nbargument!, commande$())
DECLARE FUNCTION droite! (x1!, y1!, x2!, y2!, angle!)
DECLARE FUNCTION intern! (x!, y!, x!(), y!(), n!)
DECLARE SUB pointintern (x!(), y!(), n!, minx!, maxx!, miny!, maxy!, xp!, yp!)
DECLARE FUNCTION postAngle! (v!)
DECLARE FUNCTION preAngle! (v!)
DECLARE SUB trace ()
DECLARE SUB ErrMess (n!, f$)
DECLARE FUNCTION chargerContour! (nom$, x!(), y!(), pente!(), minx!, maxx!, miny!, maxy!, nv!)
DECLARE FUNCTION airContour! (x!(), y!(), n!)
DECLARE SUB ErrProgram (n!, f$)
COMMON SHARED xmi, xma, ymi, yma, zmin, zmax, pente, zfond
COMMON SHARED minx, maxx, miny, maxy, nv, f$
ON ERROR GOTO erreur
nbargument = 4
DIM commande$(nbargument)
DIM SHARED x(1000), y(1000), pente(1000)
DIM vx(2), vy(2)
vx(1) = 1: vx(2) = 638: vy(1) = 16: vy(2) = 348
pi = 3.141592
contour% = 15

decode nbargument, commande$()

      pospt = INSTR(1, commande$(2), ".")
      IF pospt <> 0 THEN
        bln$ = MID$(commande$(2), 1, pospt - 1) + ".bln"
      ELSE
        bln$ = commande$(2) + ".bln"
      END IF

nom1$ = ext$(commande$(1), "grd")
nom2$ = ext$(commande$(2), "fos")
nom3$ = ext$(commande$(3), "grd")
zfond = VAL(commande$(4))
CLS
trace
SCREEN 9

src = 2
dest = 3
f$ = nom1$
OPEN "b", src, nom1$
IF chargerMaille(src, xmi, xma, ymi, yma, zmin, zmax, nl, nc, dest) = -1 THEN
  ErrProgram 3, nom1$: END
  PRINT "le fichier "; nom1$; " n'est pas un mnt": END
END IF
CLOSE #src
com$ = "copy " + nom1$ + " " + nom3$ + " >nul"
SHELL com$
f$ = nom1$
OPEN "b", src, nom1$
f$ = nom3$
OPEN "b", dest, nom3$
epsilon = (xma - xmi) / 1000
eps = 1 * E - 10
tm = (xma - xmi) / (nc - 1) / 2

pointintern x(), y(), nv, minx, maxx, miny, maxy, xp, yp
creerBln x(), y(), nv, bln$
IF zmin > zfond THEN zmin = zfond
SEEK #dest, 10 * 4 + 1
PUT #dest, , zmin
nv = nv - 1
CLS 0
VIEW (vx(1) - 1, vy(1) - 1)-(vx(2) + 1, vy(2) + 1)
WINDOW (0, 0)-(1, 1)
LINE (0, 0)-(1, 1), contour%, B

```

```

VIEW (vx(1), vy(1))-(vx(2), vy(2))
delta = (xma - xmi) / 50
WINDOW (xmi, ymi)-(xma, yma)
PAINT (xp, yp), 7, 8
FOR i = 1 TO nv
  IF x(i) = x(i + 1) THEN x(i + 1) = x(i + 1) + epsilon
  IF y(i) = y(i + 1) THEN y(i + 1) = y(i + 1) + epsilon
  LINE (x(i), y(i))-(x(i + 1), y(i + 1)), 1
NEXT i
PAINT (xp, yp), 2, 1
c1 = INT((minx - xmi) / (xma - xmi) * (nc - 1) + 1.5)
c2 = INT((maxx - xmi) / (xma - xmi) * (nc - 1) + 1.5)
l1 = INT((miny - ymi) / (yma - ymi) * (nl - 1) + 1.5)
l2 = INT((maxy - ymi) / (yma - ymi) * (nl - 1) + 1.5)
DIM t(l1 - 2 TO l2 + 2, c1 - 2 TO c2 + 2)
DIM t1(l1 - 2 TO l2 + 2, c1 - 2 TO c2 + 2)
DIM t2(l1 - 2 TO l2 + 2, c1 - 2 TO c2 + 2)
insurface = 1
FOR l = l1 TO l2
  FOR c = c1 TO c2
    SEEK #src, (l - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
    SEEK #dest, (l - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
    GET #src, , m
    x = (c - 1) * (xma - xmi) / (nc - 1) + xmi
    y = (l - 1) * (yma - ymi) / (nl - 1) + ymi
    v = POINT(x, y)
    IF v = 2 OR v = 1 THEN
      IF v = 2 THEN
        t2(l, c) = -1E+15
        PSET (x, y), 15
      END IF
      IF m < zfond THEN
        insurface = 0
        PUT #dest, , m
      ELSE
        PUT #dest, , zfond
      END IF
    ELSE
      t2(l, c) = -2E+15
      PUT #dest, , m
      PSET (x, y), 7
    END IF
  NEXT c
NEXT l
CLS
PAINT (xp, yp), 0, 8
IF insurface = 1 THEN
  FOR i = 1 TO nv
    LINE (x(i), y(i))-(x(i + 1), y(i + 1)), 1
  NEXT i
  col = couleur(zfond)
  PAINT (xp, yp), col, 1
END IF
x(nv + 1) = x(1)
y(nv + 1) = y(1)
pente(nv + 1) = pente(1)
x(nv + 2) = x(2)
y(nv + 2) = y(2)
pente(nv + 2) = pente(2)
minxp = 1E+30
maxxp = -1E+30
minyp = 1E+30
maxyp = -1E+30

FOR v = 1 TO nv
  LOCATE 1, 34: PRINT USING "### /###"; v; nv
  x1 = x(v)
  y1 = y(v)
  x2 = x(v + 1)
  y2 = y(v + 1)
  pente = pente(v + 1)
  a = droite(x1, y1, x2, y2, angle)
  B = y1 - a * x1
  pas1 = ABS(COS(angle) * tm)
  pas2 = ABS(SIN(angle) * tm)
  IF x1 > x2 THEN pas1 = -pas1
  IF y1 > y2 THEN pas2 = -pas2
  FOR x = x1 - pas1 TO x2 + pas1 STEP pas1
    y = a * x + B
    a2 = -1 / a
    b2 = y + x / a
    IF insurface = 0 THEN

```

```

xx = x
xx = xx + pas2
yy = a2 * xx + b2
c = INT((xx - xmi) / (xma - xmi) * (nc - 1) + 1.5)
l = INT((yy - ymi) / (yma - ymi) * (nl - 1) + 1.5)
compt = 1
WHILE t2(l, c) <> 0 OR compt = 1
  IF t2(l, c) = -1E+15 THEN
    SEEK #dest, (l - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
    GET #dest, , m
    col = couleur(m)
    xa = (c - 1.5) * (xma - xmi) / (nc - 1) + xmi
    ya = (l - 1.5) * (yma - ymi) / (nl - 1) + ymi
    xb = (c - .5) * (xma - xmi) / (nc - 1) + xmi
    yb = (l - .5) * (yma - ymi) / (nl - 1) + ymi
    LINE (xa, ya)-(xb, yb), col, BF
  END IF
  xx = xx + pas2
  yy = a2 * xx + b2
  zz = zfond
  c = INT((xx - xmi) / (xma - xmi) * (nc - 1) + 1.5)
  l = INT((yy - ymi) / (yma - ymi) * (nl - 1) + 1.5)
  compt = compt + 1
  IF INKEY$ = CHR$(3) THEN END
WEND
END IF
xx = x + pas2
yy = a2 * xx + b2
c = INT((xx - xmi) / (xma - xmi) * (nc - 1) + 1.5)
l = INT((yy - ymi) / (yma - ymi) * (nl - 1) + 1.5)
SEEK #src, (l - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
GET #src, , z
z1 = pente * SQR((xx - x) * (xx - x) + (yy - y) * (yy - y))
zz = z - z1 - zfond
compt = 1
DO WHILE zz > 0 OR compt <= 2
  compt = compt + 1
  SEEK #dest, (l - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
  GET #dest, , z2
  IF z2 > z1 + zfond THEN
    ztemp = z1 + zfond
  ELSE
    ztemp = z2
  END IF
  SEEK #dest, (l - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
  PUT #dest, , ztemp
  col = couleur(ztemp)
  xa = (c - 1.5) * (xma - xmi) / (nc - 1) + xmi
  ya = (l - 1.5) * (yma - ymi) / (nl - 1) + ymi
  xb = (c - .5) * (xma - xmi) / (nc - 1) + xmi
  yb = (l - .5) * (yma - ymi) / (nl - 1) + ymi
  LINE (xa, ya)-(xb, yb), col, BF
  xx = xx - pas2
  yy = a2 * xx + b2
  c = INT((xx - xmi) / (xma - xmi) * (nc - 1) + 1.5)
  l = INT((yy - ymi) / (yma - ymi) * (nl - 1) + 1.5)
  IF xx <= xmi OR xx >= xma OR yy <= ymi OR yy > yma THEN EXIT DO
  SEEK #src, (l - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
  GET #src, , z
  z1 = pente * SQR((xx - x) * (xx - x) + (yy - y) * (yy - y))
  zz = z - z1 - zfond
  compt = compt + 1
LOOP
IF xx > maxx THEN maxx = xx
IF xx < minxp THEN minxp = xx
IF yy > maxyp THEN maxyp = yy
IF yy < minyp THEN minyp = yy

IF INKEY$ = CHR$(3) THEN END
NEXT x
IF insurface = 0 THEN
  FOR a = 0 TO 2 * pi STEP pi / 180
    c = INT((x2 - xmi) / (xma - xmi) * (nc - 1) + 1.5)
    l = INT((y2 - ymi) / (yma - ymi) * (nl - 1) + 1.5)
    r = 0
    r = r + tm
    x = x2 + COS(a) * r
    y = y2 + SIN(a) * r
    compt = 1
    WHILE t2(l, c) <> 0 OR compt = 1
      IF t2(l, c) = -1E+15 THEN
        SEEK #dest, (l - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4

```

```

        GET #dest, , m
        col = couleur(m)
        xa = (c - 1.5) * (xma - xmi) / (nc - 1) + xmi
        ya = (1 - 1.5) * (yma - ymi) / (nl - 1) + ymi
        xb = (c - .5) * (xma - xmi) / (nc - 1) + xmi
        yb = (1 - .5) * (yma - ymi) / (nl - 1) + ymi
        LINE (xa, ya)-(xb, yb), col, BF
    END IF
    r = r + tm
    x = x2 + COS(a) * r
    y = y2 + SIN(a) * r
    zz = zfond
    c = INT((x - xmi) / (xma - xmi) * (nc - 1) + 1.5)
    l = INT((y - ymi) / (yma - ymi) * (nl - 1) + 1.5)
    compt = compt + 1
    IF INKEY$ = CHR$(3) THEN END
WEND
NEXT a
END IF
angle1 = preAngle(v + 1)
angle2 = postAngle(v + 1)
IF pi / 2 >= angle1 AND angle2 > 3 * pi / 2 THEN
    angle2 = angle2 - 2 * pi
END IF
pente1 = pente(v + 1)
pente2 = pente(v + 2)
FOR a = angle1 TO (angle2 - pi / 180) STEP -pi / 180
    alpha1 = (angle1 - a) / (angle1 - angle2)
    alpha2 = (a - angle2) / (angle1 - angle2)
    pente = (alpha2 * pente1 + alpha1 * pente2)
    r = 0
    x = x2 + COS(a) * r
    y = y2 + SIN(a) * r
    c = INT((x - xmi) / (xma - xmi) * (nc - 1) + 1.5)
    l = INT((y - ymi) / (yma - ymi) * (nl - 1) + 1.5)
    SEEK #src, (1 - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
    GET #src, , z
    z1 = pente * r
    zz = z - z1 - zfond
    compt = 1
    DO WHILE zz > 0 OR compt <= 2
        compt = compt + 1
        SEEK #dest, (1 - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
        GET #dest, , z2
        IF z2 > z1 + zfond THEN
            ztemp = z1 + zfond
        ELSE
            ztemp = z2
        END IF
        SEEK #dest, (1 - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
        PUT #dest, , ztemp
        col = couleur(ztemp)
        xa = (c - 1.5) * (xma - xmi) / (nc - 1) + xmi
        ya = (1 - 1.5) * (yma - ymi) / (nl - 1) + ymi
        xb = (c - .5) * (xma - xmi) / (nc - 1) + xmi
        yb = (1 - .5) * (yma - ymi) / (nl - 1) + ymi
        LINE (xa, ya)-(xb, yb), col, BF
        r = r + tm
        x = x2 + COS(a) * r
        y = y2 + SIN(a) * r
        c = INT((x - xmi) / (xma - xmi) * (nc - 1) + 1.5)
        l = INT((y - ymi) / (yma - ymi) * (nl - 1) + 1.5)
        IF x <= xmin OR x >= xma OR y <= ymi OR y > yma THEN EXIT DO
        SEEK #src, (1 - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
        GET #src, , z
        z1 = pente * r
        zz = z - z1 - zfond
        IF INKEY$ = CHR$(3) THEN END
    LOOP
    IF xx > maxx THEN maxx = xx
    IF xx < minx THEN minx = xx
    IF yy > maxy THEN maxy = yy
    IF yy < miny THEN miny = yy
NEXT a
rem1:
IF maxx > xma THEN maxx = xma
IF minx < xmi THEN minx = xmi
IF maxy > yma THEN maxy = yma
IF miny < ymi THEN miny = ymi

NEXT v
FOR i = 1 TO nv

```

```

LINE (x(i), y(i))-(x(i + 1), y(i + 1)), 1
NEXT i
c1 = INT((minxp - xmi) / (xma - xmi) * (nc - 1) + 1.5)
c2 = INT((maxxp - xmi) / (xma - xmi) * (nc - 1) + 1.5)
l1 = INT((minyp - ymi) / (yma - ymi) * (nl - 1) + 1.5)
l2 = INT((maxyp - ymi) / (yma - ymi) * (nl - 1) + 1.5)
vol = 0
LOCATE 1, 1: PRINT "Ecriture.... /"; nl
rap = (xma - xmi) / (nc - 1) * 1000 * (yma - ymi) * 1000 / (nl - 1)
IF l1 <= 1 THEN l1 = 2
IF l2 >= nl THEN l2 = nl - 1
IF c1 <= 1 THEN c1 = 2
IF c2 > nc THEN c2 = nc - 1
FOR l = 1 TO l1
    LOCATE 1, l2: PRINT l
NEXT l
FOR l = l1 TO l2
    LOCATE 1, l2: PRINT l
    FOR c = c1 TO c2
        SEEK #src, (l - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
        GET #src, , m1
        SEEK #dest, (l - 1) * nc * 4 + (c - 1) * 4 + 1 + 14 * 4
        GET #dest, , m2
        vol = vol + (m1 - m2)
    NEXT c
    LOCATE 1, 50: PRINT USING "Volume extrait #####, m3"; vol * rap
NEXT l
FOR l = l2 TO nl
    LOCATE 1, l2: PRINT l
NEXT l
CLOSE #src
CLOSE #dest
BEEP
CLOSE
SCREEN 0
SCREEN 0
COLOR 11, 1, 1: CLS
PRINT : PRINT
me$ = "          FOSSEBHV MNT-initial(.grd) Contour(.fos) MNT-fosse(.grd) Z-Fond"

me$ = me$ + CHR$(13) + CHR$(13)
me$ = me$ + "          Le contour inférieur de la fosse est connu."
me$ = me$ + CHR$(13)
me$ = me$ + "          L'angle par rapport à l'horizontale est contenu dans le
fichier"
me$ = me$ + CHR$(13)
me$ = me$ + "          contour. L'angle est exprimé en degrés, la cote en mètres."
me$ = me$ + CHR$(13) + CHR$(13) + SPACE$(22) + "GEOSURF 1.0 (c) BRGM 4S/PAC 1992"

PRINT me$
PRINT : PRINT
PRINT "          FOSSEBHV " + COMMAND$
PRINT
PRINT : PRINT

PRINT USING "          Surface du fond de la fosse : #####, m²"; airContour(x(),
y(), nv) * 1000000
PRINT
PRINT USING "          Volume extrait          : #####, m3"; vol * rap
LOCATE 24, 1

END
erreur:
ErrMess ERR, f$

FUNCTION airContour (x(), y(), n)
    x(n + 1) = x(1)
    y(n + 1) = y(1)
    air = 0
    FOR i = 1 TO n
        air = air + (x(i) + x(i + 1)) * (y(i) - y(i + 1))
    NEXT i
    airContour = ABS(air) / 2
END FUNCTION

FUNCTION chargerContour (nom$, x(), y(), pente(), minx, maxx, miny, maxy, nv)
    pi = 3.141592
    f$ = nom$
    OPEN nom$ FOR INPUT AS 1
    minx = 1E+30

```

```

maxx = -1E+30
miny = 1E+30
maxy = -1E+30
INPUT #1, nv
res = 1
FOR i = 1 TO nv
    INPUT #1, x(i), y(i), pente(i)
    IF x(i) <= xmi OR x(i) >= yma OR y(i) <= ymi OR y(i) >= yma THEN
        res = -1
    END IF
    IF pente(i) < 90 THEN
        pente(i) = TAN(pente(i) / 180 * pi)
    ELSE
        pente(i) = 1E+10
    END IF
    pente(i) = pente(i) * 1000
    IF x(i) > maxx THEN maxx = x(i)
    IF x(i) < minx THEN minx = x(i)
    IF y(i) > maxy THEN maxy = y(i)
    IF y(i) < miny THEN miny = y(i)
NEXT
IF minx < xmi THEN minx = xmi
IF maxx > xma THEN maxx = xma
IF miny < ymi THEN miny = ymi
IF maxy > yma THEN maxy = yma

CLOSE (1)
chargerContour = res

END FUNCTION

FUNCTION chargerMaille (src, xmi, xma, ymi, yma, zmin, zmax, nl, nc, dest)
    a$ = INPUT$(4, src)
    IF a$ <> "DSBB" THEN
        res = -1
    ELSE
        nc = CVI(INPUT$(2, src))
        nl = CVI(INPUT$(2, src))
        xmi = CVD(INPUT$(8, src))
        xma = CVD(INPUT$(8, src))
        ymi = CVD(INPUT$(8, src))
        yma = CVD(INPUT$(8, src))
        zmin = CVD(INPUT$(8, src))
        zmax = CVD(INPUT$(8, src))
        res = 0
    END IF
    chargerMaille = res

END FUNCTION

FUNCTION couleur (z)
    couleur = ((z - zmin) / (zmax - zmin) * 14) MOD 14 + 2
END FUNCTION

SUB creerBln (x(), y(), nv, nom$)
    f$ = nom$
    OPEN "o", #5, nom$
    PRINT #5, nv
    FOR i = 1 TO nv
        PRINT #5, x(i), y(i)
    NEXT i
    CLOSE #5

END SUB

SUB decode (nbargument, commande$())
    nom$ = COMMAND$
    nom$ = LTRIM$(RTRIM$(nom$))
    FOR i = 1 TO nbargument - 1
        IF nom$ <> "" THEN
            v = INSTR(nom$, " ")
            IF v = 0 THEN
                syntaxe
            END IF
            commande$(i) = LEFT$(nom$, v - 1)
            nom$ = LTRIM$(RIGHT$(nom$, LEN(nom$) - v))
        ELSE
            syntaxe
        END IF
    NEXT i

```

```

IF nom$ <> "" THEN
    commande$(nbargument) = LTRIM$(RTRIM$(nom$))
ELSE
    syntaxe
END
END IF

END SUB

FUNCTION droite (x1, y1, x2, y2, angle)
pi = 3.141592
IF x1 <> x2 AND y1 <> y2 THEN
    a = (y1 - y2) / (x1 - x2)
    angle = ABS(ATN(a))
IF x1 < x2 THEN
    IF y1 < y2 THEN
        angle = angle
    ELSE
        angle = 2 * pi - angle
    END IF
ELSE
    IF y1 < y2 THEN
        angle = pi - angle
    ELSE
        angle = pi + angle
    END IF
END IF
END IF
    droite = a
END FUNCTION

SUB ErrMess (n, f$)
SELECT CASE n
CASE 4
    message$ = "Données épuisées."
CASE 5
    message$ = "Appel de fonction interdit."
CASE 6
    message$ = "Dépassement de capacité."
CASE 7
    message$ = "Mémoire insuffisante."
CASE 9
    message$ = "Indice en dehors des limites pour un tableau."
CASE 11
    message$ = "Une division par zéro a été demandée."
CASE 24
    message$ = "Délai d'attente d'un périphérique écoulé."
CASE 25
    message$ = "Défaillance du périphérique."
CASE 27
    message$ = "Attente chargement du papier."
CASE 51
    message$ = "Erreur interne."
CASE 52
    message$ = "Nom ou numéro de fichier incorrect : " + UCASE$(f$)
CASE 53
    message$ = "Le fichier spécifié n'a pas été trouvé : " + UCASE$(f$)
CASE 54
    message$ = "Mode d'accès au fichier incorrect : " + UCASE$(f$)
CASE 55
    message$ = "Fichier déjà ouvert : " + UCASE$(f$)
CASE 57
    message$ = "Erreur d'entrée / sortie sur le périphérique."
CASE 58
    message$ = "Fichier qui existe déjà : " + UCASE$(f$)
CASE 61
    message$ = "Disque saturé."
CASE 62
    message$ = "Entrée hors des limites d'un fichier : " + UCASE$(f$)
CASE 64
    message$ = "Nom de fichier incorrect : " + UCASE$(f$)
CASE 67
    message$ = "Trop de fichiers ouvert dans le répertoire."
CASE 68
    message$ = "Périphérique non disponible."
CASE 69
    message$ = "Dépassement de capacité du tampon de communication."
CASE 70
    message$ = "Accès refusé : " + UCASE$(f$)
CASE 71
    message$ = "Disque nom prêt."
CASE 72

```

```

        message$ = "Disque défectueux."
    CASE 75
        message$ = "Erreur de chemein d'accès/d'accès au fichier."
    CASE 76
        message$ = "Chemin d'accès introuvable."
    CASE ELSE
        message$ = "Erreur inconnue..."
END SELECT
SCREEN 0
COLOR 11, 1, 1
CLS
LOCATE 5, 5: PRINT "FOSSE2NK " + COMMAND$
LOCATE 8, 5
PRINT "Une erreur d'execution a été détectée lors du déroulement du programme."

LOCATE 10, 8: PRINT "Code de l'erreur : "; ERR
LOCATE 13, 8: PRINT "Signification      : "; message$
LOCATE 15, 8: PRINT "Ecriture dans le fichier PACA-ERR"
LOCATE 17, 8: PRINT "Références         : "; "le " + DATE$ + " à " + TIME$

PLAY "l16eo5ed#c#o4bo5"

free = FREEFILE

OPEN "paca-err" FOR APPEND AS free
    PRINT #free, "-----"
    PRINT #free, "FOSSEBHV " + COMMAND$
    PRINT #free, "Code de l'erreur : "; ERR
    PRINT #free, "Signification      : "; message$
    PRINT #free, "Références         : "; "le " + DATE$ + " à " + TIME$
CLOSE free

END
END SUB

SUB ErrProgram (n, f$)
    SELECT CASE n
    CASE 1
        message$ = "Le contour doit être dans le MNT."
    CASE 2
        message$ = "Le pente doit être positive ."
    CASE 3
        message$ = "Le fichier " + f$ + " n'est pas un MNT ."

    CASE ELSE
        message$ = "Erreur inconnue..."
    END SELECT
    SCREEN 0
    COLOR 11, 1, 1
    CLS
    LOCATE 5, 5
    PRINT "      FOSSEBHV " + COMMAND$
    LOCATE 10, 8: PRINT "Code de l'erreur : "; n
    LOCATE 13, 8: PRINT "Signification      : "; message$
    LOCATE 15, 8: PRINT "Ecriture dans le fichier PACA-ERR"
    LOCATE 17, 8: PRINT "Références         : "; "le " + DATE$ + " à " + TIME$
    PLAY "l16eo5ed#c#o4bo5"
    free = FREEFILE
    OPEN "paca-err" FOR APPEND AS free
        PRINT #free, "-----"
        PRINT #free, "FOSSEBHV " + COMMAND$
        PRINT #free, "Code de l'erreur : "; n
        PRINT #free, "Signification      : "; message$
        PRINT #free, "Références         : "; "le " + DATE$ + " à " + TIME$
    CLOSE free
END
END SUB

FUNCTION ext$ (set$, sym$)

    sym$ = "." + sym$
    IF INSTR(1, set$, ".") = 0 THEN set$ = set$ + sym$
    ext$ = UCASE$(set$)

END FUNCTION

FUNCTION intern (x, y, x(), y(), n)
    i = 1
    WHILE x <> x(i) AND y <> y(i) AND i <= (n - 1)
        i = i + 1
    WEND
    IF i >= n THEN

```

```

'x n'appartient pas au contour
FOR i = 1 TO n - 1
  IF x(i) < x(i + 1) THEN
    minix = x(i)
    maxix = x(i + 1)
  ELSE
    minix = x(i + 1)
    maxix = x(i)
  END IF
  IF y(i) < y(i + 1) THEN
    miniy = y(i)
    maxiy = y(i + 1)
  ELSE
    miniy = y(i + 1)
    maxiy = y(i)
  END IF
  IF x(i + 1) <> x(i) THEN
    a = (y(i + 1) - y(i)) / (x(i + 1) - x(i))
    B = y(i) - a * x(i)
    IF a <> 0 THEN
      xx = (y - B) / a
    ELSE
      xx = x(i)
    END IF
    YY = y
  END IF
  IF xx < x THEN
    IF (minix < xx) AND (xx < maxix) AND (miniy < yy) AND (yy < maxiy) THEN
      cpt = cpt + 1
    END IF
  END IF
NEXT
END IF
intern = cpt MOD 2
END FUNCTION

SUB pointintern (x!(), y!(), n!, minx!, maxx!, miny!, maxy!, xp!, yp!)
  cpt = 0
  deltax = (maxx - minx) / (n - 2)
  deltax = (maxy - miny) / (n - 2)
  yy = miny - deltax
  WHILE (cpt <> 1) AND (yy <= maxy)
    xx = minx
    YY = yy + deltax
    WHILE (xx <= maxx) AND (cpt <> 1)
      cpt = intern(xx, yy, x!(), y!(), n)
      IF cpt = 1 THEN
        xp = xx
        yp = yy
      END IF
      xx = xx + deltax
    WEND
  WEND
END SUB

FUNCTION postAngle (v)
  pi = 3.141592
  a = droite(x(v), y(v), x(v + 1), y(v + 1), angle)
  angle = angle + pi / 2
  IF angle > 2 * pi THEN
    angle = angle - 2 * pi
  END IF
  postAngle = angle
END FUNCTION

FUNCTION preAngle (v)
  pi = 3.141592
  a = droite(x(v - 1), y(v - 1), x(v), y(v), angle)
  angle = angle + pi / 2
  IF angle > 2 * pi THEN
    angle = angle - 2 * pi
  END IF
  preAngle = angle
END FUNCTION

SUB syntaxe
  PRINT : PRINT
  COLOR 11
  BEEP

```

```

me$ = " Syntaxe :" + CHR$(13) + CHR$(13) + " FOSSEBHV MNT-initial(.grd)
Contour(.fos) MNT-fosse(.grd) Z-Fond"

me$ = me$ + CHR$(13) + CHR$(13)
me$ = me$ + " Le contour inférieur de la fosse est connu."
me$ = me$ + CHR$(13)
me$ = me$ + " L'angle par rapport à l'horizontale est contenu dans le
fichier"
me$ = me$ + CHR$(13)
me$ = me$ + " contour. L'angle est exprimé en degrés, la cote en mètres."
me$ = me$ + CHR$(13) + CHR$(13) + SPACE$(22) + "GEOSURF 1.0 (c) BRGM 4S/PAC 1992"
PRINT me$
PRINT

END
END SUB

SUB trace
f$ = "log"
OPEN "log" FOR APPEND AS 66
PRINT #66, DATE$ + " - " + TIME$; " - FOSSEBHV " + COMMAND$
CLOSE 66
END SUB

```

ANNEXE 3

PROCEDURE DE SAISIE EN 3D

SOUS MICROSTATION

```
debut:
MSG 'PRentrez le z de la courbe :'
```

```
GET K,saisie
go fin
```

```
saisie:
tst key eq "", fin
set A3=key
go couleur
```

```
fincouleur:
set c0='PRz='+key
MSG c0
msg 'PR'
Key 'Place lstring'
newpoint:
get p,ok,r,reset
```

```
go fin
reset:
key 'reset'
go debut
ok:
set A1=XUR/100000
set A2=yur/100000
set c0='prxy='+A1
set c0=c0+', '
set c0=c0+A2
set c0=c0+', '
set c0=c0+A3
msg c0
set c0='xy='+A1
set c0=c0+', '
set c0=c0+A2
set c0=c0+', '
set c0=c0+A3
key c0
go newpoint
go debut
```

```
fin:
msg 'pr'
```

```
end
couleur:
tst a3 gt 100,s10
key 'co=1'
go fincouleur
s10:
```

```
tst a3 gt 200,s20
key 'co=2'
go fincouleur
s20:
```

```
tst a3 gt 300,s30
key 'co=3'
go fincouleur
s30:
```

```
tst a3 gt 400,s40
key 'co=4'
go fincouleur
s40:
```

```
key 'co=9'
go fincouleur
```

ANNEXE 4

GEODISC

MCM/mm



INSTITUT GEOGRAPHIQUE NATIONAL
SECRETARIAT GEODISC

Saint-Mandé, le 23 mars 1992

MC-AB

26 MARS 1992

BRGM-PACA
Monsieur BOURROUSSE
117, avenue de Luminy

13009 MARSEILLE

Monsieur,

Vous trouverez ci-joint le procès verbal de notre réunion de délibération du concours Géodisc 92 décernant un label aux 12 sociétés lauréates.

En attendant de vous retrouver tous au CNIT le 31 mars, je vous présente déjà mes félicitations.

Je vous prie d'agréer, Monsieur, l'expression de ma considération distinguée.

Le Secrétariat Géodisc

Marie-Cécile MAHIEU



27.03.92



INSTITUT GEOGRAPHIQUE NATIONAL
SECRETARIAT GEODISC

Opération Géodisc

09 mars 1992

I.G.N. - 136 bis rue de Grenelle

Délibération du jury - Procès verbal.

Présents :

Président du jury : . Michel PETIT, délégué Général à l'Espace,
Ministère de l'Équipement, du Logement, des Transports
et de l'Espace.

Membres :

- . Jean-François CARREZ, Directeur Général de l'IGN,
- . Fernand VERGER, Directeur du Laboratoire Imageo au CNRS,
- . Hervé THERY, Directeur de la Maison de la Cartographie,
- . Jacques NAVES - Agence de l'Eau Seine-Normandie,
(remplaçant de Michel ROUX, Agence de l'Eau Adour-Garonne),
- . Frédéric LENNE (remplaçant de Philippe CHAGNON)
Le Moniteur,
- . Pierre BERGER, journaliste, Le Monde Informatique,
- . Philippe PARMENTIER, journaliste, Le Moniteur des Villes,
- . François SALGE, Chef de Groupe Technique - CERCO,
- . Sylvie LAMY, Responsable de la cellule SIG, IGN.



INSTITUT GEOGRAPHIQUE NATIONAL
SECRETARIAT GEODISC

9h30 - 9h45

Accueil et rappel de la mécanique de l'opération par J-F. Carrez, Directeur Général de l'IGN et Marie-Cécile Mahieu, responsable du Secrétariat Géodisc.

9h45 - 10h

Présentation des dossiers par Marie-Cécile Mahieu.

Nombre de dossiers pré-sélectionnés au 26.06.91 :

41 candidats pour 45 applications.

Nombre de dossiers présentés au jury au 09.03.92 :

21 candidats pour 23 applications.

10h-13h

Le retour des dossiers de présentation des applications s'est effectué entre le 15 et le 31 janvier. A la demande des candidats, les expertises se sont effectuées le plus tard possible dans la deuxième quinzaine de février, ce qui n'a pas permis de transmettre les dossiers aux membres du jury avant la journée de délibération. Aussi, le secrétariat Géodisc a mis le plus grand soin à présenter :

- une fiche synthétique de chaque dossier présentant la société candidate et le projet retenu à la présélection ;

- un compte rendu d'expertise technique de 2 pages maximum rédigé par l'expert IGN désigné par le secrétariat Géodisc sur les conseils de la Direction Technique. Chaque dossier a été étudié par 2, voire 3 ingénieurs spécialisés dans les domaines d'applications présentées pour garantir une bonne objectivité d'analyse ;

- les documents fournis par les candidats, dessins, photos d'écran etc... ont circulé en tour de table pendant la présentation de chaque dossier.

14h - 15h

Après délibérations et consensus des membres du jury présents, les 12 lauréats, par ordre alphabétique, retenus à recevoir le label Géodisc 92, sont les suivants :



INSTITUT GEOGRAPHIQUE NATIONAL
SECRETARIAT GEODISC

- ABAK systèmes
- APIC systèmes
- BRGM-PACA
- CLEMESSY
- DATA IMAGE
- INFOGRAPH
- INFOSYS
- IOS Région sud
- Laboratoire de Robotique CNRS
- OKTAL
- SAFEGE
- SIAGE

3 lauréats recevront le 31 mars 1992, lors du forum final un prix spécial confirmant l'esprit d'innovation dans le domaine de l'application développée sur les données géographiques numériques de l'IGN.

Le président du Jury et M. Carrez, notre directeur général, adresse tous ses encouragements aux sociétés qui, malgré le travail fourni, n'ont pas été sélectionnées à ce jour, ou qui, pour des contraintes bien compréhensibles ont dû renoncer à leur participation.

Le Secrétariat Géodisc

M-C. MAHIEU

LE DIRECTEUR GÉNÉRAL

244 IGN/DG

12 MARS 1992

Monsieur,

Au terme de la journée du 9 mars, le jury GEODISC a délibéré et vous a sélectionné parmi les lauréats.

Je tiens à vous féliciter pour la qualité de votre travail et pour l'importance de l'effort que vous avez fourni dans le cadre de notre concours.

Vous êtes maintenant convié à recevoir des mains du jury présidé par Monsieur Paul Quilès, le label IGN-GEODISC, en avant-première du Salon MARI "l'information géographique numérique" au CNIT - Paris La Défense, le mardi 31 mars 1992.

C'est ce même jour que les Prix Spéciaux seront remis à trois lauréats pour récompenser les applications retenues, en particulier pour leurs qualités techniques, leur innovation ou leur performance.

Le nom de ces lauréats sera tenu secret jusqu'à cette date.

.../...

Monsieur BOURROUSSE
BRGM - PACA
117 avenue de Luminy
13009 MARSEILLE

Par ailleurs, nous vous proposons à cette occasion, un espace de "démonstration" dans lequel vous pourrez aménager un stand : vos travaux pourront ainsi être présentés à un vaste public d'aménageurs, de décideurs, publics ou privés, et d'utilisateurs potentiels.

Je serais très heureux de pouvoir vous redire toutes mes félicitations au cours de cette journée du 31 mars.

Je vous prie d'agréer, Monsieur, l'expression de ma considération distinguée.

A handwritten signature in black ink, appearing to read 'J. Carrez', with a stylized flourish at the end.

Jean-François CARREZ