

Modélisation des Transports Urbains Approches Désagrégées

Laurent Pierre CHIGNAC

Année universitaire 2000

DESS Méthodes stochastiques et recherche opérationnelle ¹

Lieu de stage : CETE du Sud-Ouest ²



¹Université Bordeaux 1 – 351, Cours de la Libération – 33405 Talence cedex

²rue Pierre Ramond – Caupian – BP C 33165 Saint-Médard-en-Jalles cedex

Avant-propos

Au sujet de ce document

Il s'agit du rapport de stage que j'ai effectué au CETE du Sud Ouest au sein du département DAI³– GDU⁴. Il a été rédigé avec L^AT_EX sous windows. Les principaux outils nécessaires pour compiler des fichiers L^AT_EX sur cet environnement sont détaillés sur mon site web⁵ où ce document est aussi disponible.

Remerciements

Je remercie les personnes qui m'ont aidé à mener le projet à son terme :

- Le professeur Paul Morel, mon responsable de stage auprès de l'université de Bordeaux 1 ;
- Monsieur Gilles Dumartin, chef du GDU, qui a dirigé mon stage au sein du CETE ;
- Monsieur Jean-Michel Janaud, chargé d'études, pour ses conseils.

³Département Aménagement Infrastructures

⁴Groupe Déplacements Urbains

⁵pchignac@free.fr

Table des matières

Avant-propos	1
Au sujet de ce document	1
Remerciements	1
Introduction	4
Présentation du CETE	5
1 Précisions, définitions et généralités	10
1.1 Vocabulaire, notations	10
1.2 Typologie des données utilisées	10
1.2.1 L'enquête ménage	10
1.2.2 La matrice des coûts de déplacements	11
1.3 Une méthodologie classique : le modèle à 4 étapes	11
1.3.1 La génération	11
1.3.2 La distribution	11
1.3.3 La répartition modale	12
1.3.4 L'affectation	12
1.4 Pourquoi une approche désagrégée ?	12
2 Les modèles de déplacements, de comportements et de prévisions	14
2.1 Méthode d'estimation des matrices \mathcal{O}/\mathcal{D}	14
2.1.1 Le modèle gravitaire	14
2.1.2 Estimation de la fonction d'impédance	16
2.1.3 Calage du modèle gravitaire	16
2.2 Le modèle de choix discret	16
2.2.1 L'univers de choix	16
2.2.2 Forme et spécification de la fonction d'utilité	17
2.2.3 Les problèmes liés aux types des données	17
2.2.4 Le modèle Logit	18
2.2.5 Estimation des paramètres	19
2.2.6 Résolution numérique	21
2.2.7 Qualité du modèle	21
3 Le composant informatique	23
3.1 La classe Tmatrice	23
3.2 Les vecteurs "tetras"	23
3.3 Manuel d'utilisation du logiciel	24

4 Essais numériques	25
Conclusion	26
A Utilisation d'EXCEL pour estimer un modèle bilinéaire	27
B Code source	29
Programme principal	29
Unité de gestion matricielle	29
Module du modèle gravitaire	32
Définitions et outils nécessaires au modèle désagrégé	35
Unité du modèle désagrégé de répartition modale	40
Bibliographie	45

Introduction

Parmi les activités du GDU, on compte l'étude des comportements de mobilité en milieu urbain. Pour permettre le test de scénarii sur la mise en place d'infrastructures de transport, le groupe a souhaité avoir à sa disposition un outil spécifique permettant de simuler et de prévoir les déplacements d'individus sur l'ensemble d'une agglomération.

Cet outil de simulation devrait être expérimenté dans le cadre du plan de déplacement urbain de la ville de Pau. L'objectif du PDU étant d'orienter la politique de transport de l'agglomération vers une utilisation plus rationnelle de la voiture et de promouvoir l'usage des modes alternatifs : transports en commun, deux roues et marche à pied. Il s'agit dans ce cadre de définir des mesures nouvelles en matière d'offre de transports et d'évaluer leur impact sur les comportements de mobilité.

Les travaux effectués pendant le stage s'inscrivent dans la partie modélisation du plan de travail du PDU et doivent compléter les analyses du bureau d'étude retenu pour la réalisation du projet : le CODRA.

L'outil implémenté pour le CETE se présente sous forme d'un logiciel. Sa conception s'appuie sur une méthodologie détaillée dans le premier chapitre. Il doit être capable d'analyser les données issues d'une enquête ménage de manière à rendre compte de la situation actuelle des déplacements et du trafic sur les différents réseaux de transport et d'aboutir à une prévision multimodale des déplacements liée à des scénarios prospectifs d'offre de transports.

Le deuxième chapitre est consacré à l'étude théorique des modèles, chaque partie expose leurs lois, les données nécessaires à l'estimation statistique des paramètres ainsi que les méthodes de résolution numériques.

La troisième partie présente les moyens informatiques déployées pour optimiser les calculs. On décrira entre autre les méthodes objets et la gestion dynamique de la mémoire. Cette section inclura également un manuel d'utilisation du logiciel développé lors du stage.

Enfin un chapitre sera dédié à la validation des résultats d'après des jeux de données que l'on sait auparavant expliquer.

Présentation du C.E.T.E

Historique

Les C.E.T.E. (Centres d'Etudes Techniques de l'Equipement) ont été créés lors de la constitution d'un nouveau ministère : le Ministère de la Construction. Dans cette période de reconstruction de la France, le Bureau des Ponts et chaussées ne pouvant faire face à ce surplus de travail forme le Service d'Etude des Travaux Routiers et Autoroutiers qui est constitué de douze Laboratoires Régionaux. C'est dans les années soixante que sont créés les C.E.T.E. pour répondre à la nécessité d'avoir des moyens d'études centralisés. Ils naîtront du regroupement des Laboratoires Régionaux et des Bureaux Régionaux de Circulation pour permettre un travail simultané sur le plan de l'information des administrations et sur le plan autoroutier. Les C.E.T.E. sont rattachés au Ministère de l'Equipement.

Les C.E.T.E : une entité spécifique

Les C.E.T.E. sont spécifiques de par la place qu'ils occupent au sein du Ministère de l'Equipement, mais aussi en raison de leur implantation, leurs activités et leur financement.

Leur position au sein du Ministère de l'Equipement

Le Ministère de l'Equipement est divisé en trois grandes organisations :

- une administration centrale
- des services de terrain
- la Direction Régionale de l'Equipement (D.R.E.)
- la Direction Départementale de l'Equipement (D.D.E.)
- des réseaux techniques
- organismes concentrés dans lesquels le personnel élabore la théorie et le savoir-faire technique du Ministère. Ils sont composés d'un certain nombre de bureaux d'études comme le Service d'Etude Technique des Routes et Autoroutes (S.E.T.R.A.) pour les routes, les Laboratoires Centraux des Ponts et Chaussées (L.C.P.C.) pour les mesures, etc.
- organismes déconcentrés qui sont chargés d'intervenir sur le terrain : les C.E.T.E.

Leur implantation

Les C.E.T.E. sont répartis en sept zones qui ne tiennent pas compte du découpage régional :

- Aix-en-Provence
- Bordeaux (avec une antenne à Toulouse)
- Lille
- Lyon
- Metz
- Nantes
- Rouen

Ils ont chacun une zone d'action qui s'étend sur plusieurs départements ; ils couvrent la totalité du territoire métropolitain ainsi que les DOM-TOM. Ces centres sont la "porte d'entrée" du Réseau Scientifique et Technique. Proches du terrain, en liaison permanente avec les Directions Départementales de l'Équipement et leurs subdivisions, ils sont armés pour capitaliser les expériences locales et en faire bénéficier leurs autres partenaires.

Les C.E.T.E. rassemblent environ 4 000 personnes, dont la moitié travaille dans 17 laboratoires régionaux de Ponts et Chaussées. Ils disposent également de deux centres d'essais de construction de prototypes, à Rouen et Angers, d'un centre d'expérimentation routier à Rouen et d'une station d'essais des matériels routiers à Blois. Leurs équipes de spécialistes contribuent à l'élaboration des doctrines, des méthodes et des outils techniques, ainsi qu'à leur mise en œuvre par des actions de recherches ou d'études, d'assistance technique, de formation, de contrôle et d'essais en laboratoire ou sur le terrain. Ces activités sont réalisées d'abord pour le compte de l'État. Mais les collectivités territoriales, les secteurs parapublics (comme par exemple les concessionnaires d'autoroutes) et le secteur privé en bénéficient aussi, les C.E.T.E. veillant à la complémentarité de leurs interventions avec celles de l'ingénierie privée.

Leurs activités

Les champs de compétence des C.E.T.E. sont d'une grande diversité. Ils agissent soit seuls, soit en collaboration avec divers services techniques spécialisés. Leurs activités peuvent sommairement se regrouper en trois grands domaines :

Transports et infrastructures :

Les infrastructures de transport sont la première compétence des C.E.T.E. Cela consiste en l'étude des projets, les reconnaissances géotechniques, la conception, le contrôle d'exécution et l'auscultation des chaussées et ouvrages d'art. Leurs activités s'orientent de plus en plus vers la prise en compte de l'impact sur l'environnement, les problèmes d'entretien et d'exploitation des réseaux. Ils contribuent à la sécurité et au confort des usagers en améliorant la qualité des équipements de la route, qu'il s'agisse de la signalisation, de l'adhérence ou de la visibilité. Ils jouent un rôle essentiel de qualification et de normalisation. Les C.E.T.E. collaborent aux grands programmes européens qui, comme DRIVE, développent les applications de la télématique à la gestion

du trafic. Ils participent également au fonctionnement des Centres Régionaux d'Information et de Coordination Routières.

Aménagement, habitat, urbanisme et construction :

Dans ce domaine essentiel à notre cadre de vie, les C.E.T.E. s'impliquent de plus en plus dans des études de planification territoriale et de diagnostic allant d'un simple bassin de vie à une région entière. Celles-ci permettent à l'administration de jouer son rôle d'information et de conseil des collectivités. Elles sont nécessaires à l'élaboration des directives territoriales d'aménagement. Les C.E.T.E. s'investissent dans les Systèmes d'Information Géographique pour aider les acteurs de l'observation et de la gestion des territoires. Ils apportent leur appui technique à la conduite d'opérations et à la gestion des constructions publiques. Ils s'affirment comme des spécialistes de l'économie de l'habitat au sens large, c'est-à-dire incluant les problèmes de sécurité, de confort, de communication et de maintenance. Enfin, ils donnent le concours de leur expertise à la politique de la ville et à la réhabilitation économique et sociale des quartiers, ainsi qu'aux actions d'évaluation des politiques publiques.

Environnement et gestion des risques :

Une des facettes les plus remarquables de l'activité des C.E.T.E. est l'étude des risques naturels - comme les inondations, mouvements de terrains ou séismes - et des problèmes d'environnement, qu'il s'agisse de la pollution, de la faune, de la protection des paysages ou du bruit. Cette vocation des C.E.T.E. est favorisée par la permanence de leur implantation locale et la constitution progressive d'une mémoire des territoires à l'échelle interrégionale. Ce champ en plein développement leur offre l'occasion de multiplier des accords de partenariat avec des organismes techniques ou scientifiques.

Leur financement

Le financement peut être divisé en deux parties :

- les salaires qui sont garantis par le Ministère. En échange, les C.E.T.E. doivent collaborer aux travaux pour le Ministère de l'Équipement en général. C'est ce qui est appelé "droit à la prestation".
- les fournitures et les investissements : ils sont payés par l'activité commerciale des C.E.T.E., c'est-à-dire les commandes qu'ils exécutent pour les collectivités territoriales (mairies, conseil régional), les autres administrations ainsi que pour des entreprises privées.

Le C.E.T.E. doit équilibrer ses frais de personnel qui sont couverts par les autorisations de commande (A.C. C.E.T.E.) "payées" par les administrations et ses frais de fonctionnement (ordinateurs, frais de déplacement, électricité, fournitures de bureau) payées par les autres clients en "argent frais". Les C.E.T.E. travaillent en réseau : cela fait leur force.

Le C.E.T.E. du Sud-ouest

Généralités

Le C.E.T.E. du Sud-Ouest est un service déconcentré pluri-régional. Sa zone d'action s'étend sur quatre régions :

- Aquitaine
- Limousin
- Poitou-Charentes
- Midi-pyrénées

Et couvre vingt départements.

Le C.E.T.E. du Sud-Ouest a ses unités réparties sur deux départements et quatre communes :

- En Gironde :
 - Saint-Médard-en-Jalles (siège)
 - Mérignac (le C.R.I.C.R.)
 - Bordeaux-Caudéran (laboratoire)
- En Haute-Garonne :
 - Toulouse (antenne et laboratoire)

Son effectif est composé de 482 personnes réparties en :

- personnel non titulaire de l'Etat (P.N.T.) : 58%
- fonctionnaires : 42%
- personnel administratif : 18%
- personnel technique : 82%

Son organisation

Elle comprend :

- la Direction - où sont directement rattachés
- 2 consultants experts
- une chargées de communication
- la Division Secrétariat Général - 19 personnes - comprenant la gestion du personnel, la comptabilité, la formation, le comité local d'action sociale.
- la Division Logistique et Moyens Généraux - 23 personnes - avec un atelier bureautique, un service moyens généraux, un atelier de reprographie, un responsable sécurité et prévention.
- le Département Aménagement Infrastructures (D.A.I.) - 75 personnes - a en charge des projets routiers, l'environnement, l'économie des transports et des territoires, les transports urbains, le logement, la construction, la ville, l'habitat, la modélisation des transports interurbains.
- le Département Informatique et Modernisation (D.I.M.) - 70 personnes - est un immense centre serveur national ; il assure la coordination des actions commerciales, l'assistance aux services déconcentrés, la communication électronique et la formation, l'expertise en réseaux.
- la Division Terrassements, Chaussées, Exploitation, Sécurité (D.T.C.E.S.) - 53 personnes - comprend le C.R.I.C.R. (Centre Régional d'Information et de Circulation Routière), l'Observatoire de Circulation et de Sécurité s'occupe de gestion et d'informatique routières, de techniques et de sécurité de la route.

- La Division antenne de Toulouse - 20 personnes - s'occupe de transports, de circulation et d'aménagement pour la région Midi-Pyrénées.
- le Laboratoire Régional des Ponts et Chaussées de Bordeaux (L.R.P.C.) - 106 personnes - dont les activités sont la recherche et le développement technologique, les études dans les domaines des ouvrages d'art, de la géotechnique, des chaussées.
- le Laboratoire Régional des Ponts et Chaussées de Toulouse - 94 personnes - a les mêmes activités que le laboratoire de Bordeaux pour la région Midi-Pyrénées.

Chapitre 1

Précisions, définitions et généralités

1.1 Vocabulaire, notations

L'agglomération étudiée a été préalablement découpée en zones. En ce qui concerne la ville de Pau, un zonage en 51 parcelles a été défini.

Les déplacements de zone à zone sont représentés par des matrices, on note :

- \mathcal{O}/\mathcal{D} la matrice origine destination ;
- T_{ij} nombre de personnes allant de la zone i vers la zone j ;
- $O_i = \sum_j T_{ij}$ les émissions de la zone origine (marge verticale de \mathcal{O}/\mathcal{D}) ;
- $D_j = \sum_i T_{ij}$ les attractions vers la zone de destination (marge horizontale) ;
- c_{ij} le coût du trajet de i à j , ce peut être une distance, un coût TC, un coût généralisé...

$$\mathcal{O}/\mathcal{D} = \begin{pmatrix} & & j & & & \\ & & \vdots & & & \\ & & \vdots & & & \\ \dots & \dots & T_{ij} & \dots & \dots & \\ & & \vdots & & & \\ & & \vdots & & & \\ & & D_j & & & \end{pmatrix} \begin{pmatrix} \vdots \\ \vdots \\ O_i \\ \vdots \\ \vdots \end{pmatrix} i$$

FIG. 1.1 – Matrice des déplacements

1.2 Typologie des données utilisées

1.2.1 L'enquête ménage

Les principales ressources proviennent d'une enquête ménage effectuée en 1996 par le syndicat intercommunal des transports en commun de l'agglomération pa-

loise (SITAP). Un échantillon de 6800 personnes a été interrogé à domicile sur leurs déplacements de la veille. Après traitement, les données sont disponibles au travers d'une base Access. Les trois tables principales contiennent respectivement des renseignements sur : les ménages, les individus et leurs déplacements. Un système de clefs permet d'établir des relations entre les tables. Ainsi pour chaque déplacement, on en connaît l'origine, la destination, les informations sur la personne qui l'a effectué. . .

1.2.2 La matrice des coûts de déplacements

Nécessaire au calage des modèles, elle a été générée par DAVISUM. Ce logiciel permet de relier le zonage de Pau, établi à partir d'un SIG, à un réseau routier. Par des calculs de plus courts chemins, il calcule des coûts de déplacements sur le réseau chargé et renvoie une matrice.

1.3 Une méthodologie classique : le modèle à 4 étapes

Le modèle à quatre étapes est un schéma général classique des études de transport. Habituellement exploité dans sa dimension agrégée en France, nous avons souhaité nous orienter vers une approche désagrégée de la même manière que certains modèles déjà élaborés à Rennes, Grenoble ou Lyon. Les sections suivantes présentent les phases successives d'un tel modèle.

1.3.1 La génération

Il s'agit de quantifier les flux émis et reçus par chaque zone de l'agglomération. Arbitrairement, nous étudions les trajets pour motif domicile-travail extraits des tables de l'enquête ménage. A partir d'une matrice des déplacements observée, on calcule les marges comme indiqué dans la figure (1.1). Les vecteurs ainsi obtenus doivent être représentatifs des paramètres socio-économiques correspondants au motif choisi pour l'étude. Dans notre cas les émetteurs sont les actifs, les attracteurs : le nombre d'emplois par zone.

Remarque :

La matrice de déplacements observée ne peut-être directement utilisée. En effet celle-ci est souvent creuse du fait du faible taux d'échantillonnage utilisé : sur Pau, le zonage employé entraîne l'utilisation de matrices de dimension $51 * 51$. Il n'y a donc pas assez de données pour constater les déplacements de chaque zone à chaque zone. La prochaine étape présente une solution à ce problème.

1.3.2 La distribution

Cette phase consiste à reconstituer les matrices \mathcal{O}/\mathcal{D} représentatives des volumes de déplacements tous modes entre chaque couple de zones à partir des résultats précédents et au travers d'un modèle. On cherche à s'ajuster au mieux à la matrice

\mathcal{O}/\mathcal{D} observée. Dans son ouvrage sur les modèles de transport, Ortùzar [1] propose plusieurs méthodes. Nous retiendrons :

1. *Le maximum d'entropie* : inspiré des travaux de Wilson [3], c'est une méthode issue de la théorie de l'information et qui se fonde sur l'analyse combinatoire. L'entropie est donnée par :

$$\mathcal{W}(T_{ij}) = \frac{T!}{\prod_{i,j} T_{ij}!}$$

et représente un nombre de déplacements de zone à zone. Le principe est de maximiser l'entropie \mathcal{W} sujette à différents types de contraintes, citons par exemple :

- des contraintes sur les marges telles que :

$$O_i - \sum_j T_{ij} = 0 \forall i$$

$$D_j - \sum_i T_{ij} = 0 \forall j$$
- des contraintes issues de comptages (cf. Les Rapports de L'INRETS [4]).

Dans tous les cas, le maximum d'entropie est un problème d'optimisation non linéaire à contraintes. Il est résolu par le calcul du Lagrangien et de ses multiplicateurs.

2. *Le modèle gravitaire* : celui que nous avons choisi dans sa formulation la plus aboutie (cf. section 2.1).

1.3.3 La répartition modale

Une fois la matrice tous modes obtenue, elle doit être éclatée en matrices par modes afin d'estimer les volumes de déplacements pour chaque moyen de transport. Cela concerne les transports en commun, le trafic VP¹ et les modes lents (vélo et marche à pieds). Pour cette étape nous avons choisi d'établir un modèle de choix discret fondé sur une analyse désagrégée des données. Il s'agit d'expliquer le choix que fera un individu qui doit effectuer un trajet, parmi les alternatives qui lui sont proposées.

1.3.4 L'affectation

C'est l'estimation de la charge de trafic sur les arcs du (ou des) réseau(x) pour chaque mode. Cette étape n'a pas été implémentée, elle est réalisée à l'aide d'un progiciel existant : DAVISUM.

1.4 Pourquoi une approche désagrégée ?

On peut distinguer plusieurs manières de représenter les données concernant les individus. Considérons les trois états suivants :

¹voiture particulière

- Les *Micros États* qui renseignent directement sur les individus et les détails qui les concernent. Il s'agit en quelque sorte d'un niveau de désagrégation maximum. On parle alors de données de type désagrégé.
- Les *États Médians* qui correspondent à une agrégation des états précédents suivant des classes d'individus ou de modalités identiques.
- Les *Macros États* représentent des caractères propres à une population entière : nombre total de déplacements, moyenne d'âge ...

Ceci n'a rien de formel et il faut savoir qu'il y a toujours moyen d'obtenir des niveaux d'agrégation différents. Pour une enquête ménage par exemple, on peut considérer que les données sont dans un état désagrégé bien que les personnes interrogées aient au préalable été tirées au sort dans un zonage prédéfini. Ces données sont donc représentatives des différentes zones mais pas de la région dans son ensemble.

Il n'y a pas a priori un état (parmi les trois que l'on a mis en évidence) meilleur que les autres. A chaque niveau d'agrégation, on perd en consistance mais on gagne en stabilité. On peut envisager pour toutes les phases d'un modèle à quatre étapes d'utiliser au choix des données agrégées ou désagrégées.

Chapitre 2

Les modèles de déplacements, de comportements et de prévisions

2.1 Méthode d'estimation des matrices \mathcal{O}/\mathcal{D}

2.1.1 Le modèle gravitaire

C'est un moyen de déterminer les éléments T_{ij} de la matrice de déplacements que l'on souhaite reconstituer. L'idée est de reproduire le phénomène physique de l'attraction de Newton. Une des formulations les plus aboutie est littéralement donnée par :

$$T_{ij} = A_i O_i B_j D_j f(c_{ij}) \quad (2.1)$$

On fait ainsi intervenir :

- Les volumes d'entrée et sortie des zones origines et destinations (déterminés lors de la première étape),
- des "coefficients balances" A_i et B_j ,
- une fonction f à déterminer.

Les coefficients balance sont définis par :

$$\begin{cases} A_i = \frac{1}{\sum_j B_j D_j f(c_{ij})} \\ B_j = \frac{1}{\sum_i A_i O_i f(c_{ij})} \end{cases}$$

Ils sont déterminés par itérations successives, en prenant tous les B_j égaux à 1 comme conditions initiales par exemple et jusqu'à convergence du processus. Ceux-ci servent à se caler sur les marges. Effectivement dans l'équation (2.1), en sommant les T_{ij} sur j , on retrouve O_i en remplaçant les A_i et B_j par leurs valeurs respectives.

La fonction f dite fonction d'impédance met en évidence la relation au coût de déplacement, intrinsèque à l'éloignement respectif des zones. Les formes les plus utilisées pour cette fonction sont :

$$\begin{aligned}
 f(c_{ij}) &= e^{-\beta c_{ij}} \\
 f(c_{ij}) &= c_{ij}^n \\
 f(c_{ij}) &= \alpha c_{ij}^n e^{-\beta c_{ij}}
 \end{aligned}
 \tag{2.2}$$

La fonction à interpoler est obtenue à partir d'une analyse croisée de la table des déplacements. Ces derniers sont comptabilisés par tranches de 5 minutes par exemple. La représentation graphique du résultat est donnée par la figure (2.1).

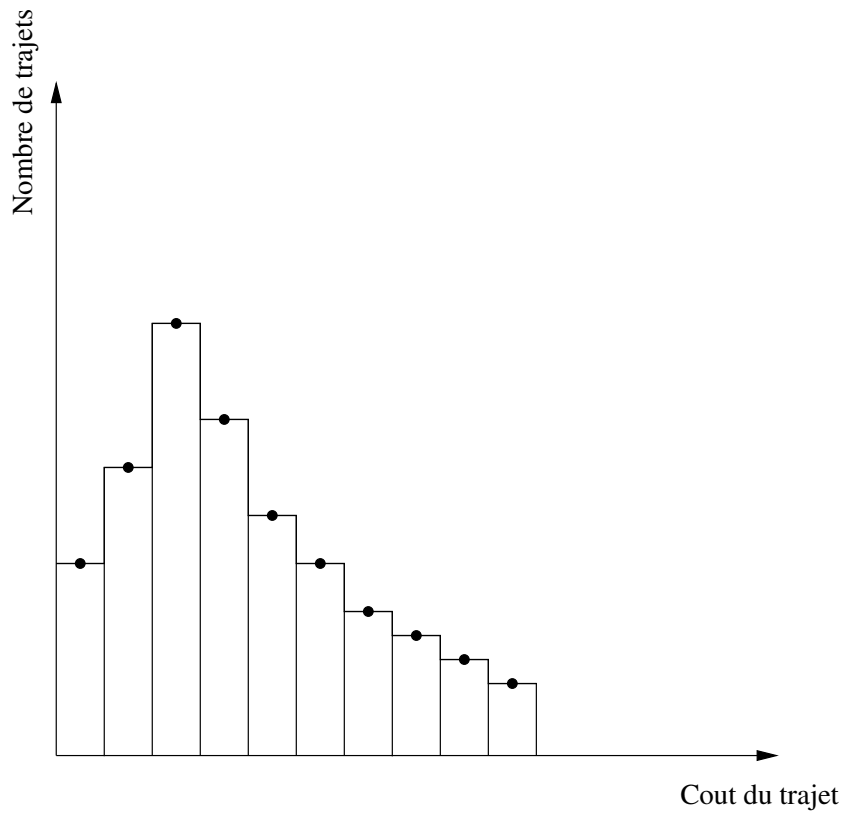


FIG. 2.1 – Distribution observée du coût des déplacements en zone urbaine

Etrangement, peut-être par facilité, la plupart des modèles mis en place lors d'études antérieures utilisent l'expression exponentielle seule. Pourtant il semble que la formule la plus élaborée soit celle donnée par l'équation (2.2). Une telle fonction combinant un terme polynomial et un terme exponentiel rend compte de la dépression initiale contrairement à une exponentielle pure qui est strictement décroissante.

2.1.2 Estimation de la fonction d'impédance

Les coefficients n et β de f sont estimés par régression bilinéaire en modifiant l'aspect de cete application.

En effet, en passant au logarithme :

$$f(x) = \alpha x^n e^{-\beta x}$$

↓

$$\log(f(x)) = n \log(x) - \beta x + \log(\alpha)$$

et par le changement de variables :

$$\begin{cases} x_1 = x, \\ x_2 = \log(x), \\ y = \log(f(x)), \\ p = \log(\alpha), \end{cases}$$

on obtient un modèle linéaire :

$$y = nx_1 - \beta x_2 + p \tag{2.3}$$

La fonction DROITEREG d'EXCEL permet d'effectuer des régressions linéaires multiples, une notice d'utilisation se trouve à la fin du mémoire en annexe.

2.1.3 Calage du modèle gravitaire

Une fois obtenus les paramètres qui décrivent f il faut prévoir une procédure de calcul pour les A_i et les B_j .

On peut alors appliquer directement la formule (2.1). On obtient ainsi une matrice de déplacements lissée par rapport celle issue de la situation observée.

2.2 Le modèle de choix discret

Ce type de modèle désagrégé est implémenté dans l'optique d'une prévision multimodale des déplacements. Il s'agit de déterminer les mécanismes qui régissent les comportements individuels et de pouvoir isoler les critères de sensibilité. Dans le cas de l'exploitation d'une enquête ménage, nous prendrons en compte les variables définies par les différents champs d'une table des déplacements. Il peut s'agir de temps d'accès, de temps de parcours par modes de transports, de caractéristiques socio-démographiques de celui qui effectue le déplacement ...

2.2.1 L'univers de choix

Dans notre étude, cet univers se compose de trois options de mode de transport :

- La voiture particulière,
- les transports en commun,
- les modes lents.

Nous cherchons à quantifier l'intérêt de toute personne pour chacune des ces éventualités. On définit ainsi une fonction d'utilité propre à chaque mode. Elle est sensée traduire le niveau de satisfaction que procure un mode à un usager selon les différentes origines-destinations considérées.

2.2.2 Forme et spécification de la fonction d'utilité

Par la suite, nous désignerons :

- par i, j des indices relatifs à des modes,
- par n des individus.

Soit U_{in} l'utilité du mode i pour l'individu n . Nous faisons l'hypothèse que c'est une variable aléatoire qui peut se décomposer sous la forme :

$$U_{in} = V_{in} + \epsilon_{in}$$

où :

- V_{in} est une variable aléatoire représentative des caractéristiques de l'individu n et de l'offre pour le mode i . Généralement, V_{in} est décrite comme une fonction linéaire de variables qualitatives et quantitatives i.e. :

$$V_{in} = \theta_{i_1} X_{i_1 n} + \theta_{i_2} X_{i_2 n} + \dots + \theta_{i_k} X_{i_k n}$$

- ϵ_{in} désigne la perturbation liée à l'incertitude des mesures...

La spécificité du logiciel développé est de gérer dynamiquement les variables que l'utilisateur souhaite introduire dans ses fonctions d'utilité, c'est lui qui les désigne lors de l'exécution du programme. Cela a plusieurs avantages :

- le modèle n'est pas "figé" sur une disposition particulière des données,
- il simplifie les tests répétitifs pour justifier du choix des variables qui semblent importantes à retenir,
- il peut être réutilisé pour d'autres études.

2.2.3 Les problèmes liés aux types des données

Comme nous l'avons vu précédemment, la partie déterministe de la fonction d'utilité se décompose en fonction des variables que l'on tient à prendre en compte.

Dans le cas de variables qualitatives comme la CSP¹, il serait malvenu de vouloir déterminer un paramètre θ unique tel que $V_{in} = \dots + \theta X_{\text{CSP}} + \dots$, X_{CSP} prenant les valeurs qui déterminent la CSP de l'individu et qui sont issues d'un codage (cf. exemple tableau (2.1)). Il convient alors de vectoriser la variable et d'associer un paramètre à chaque modalité de manière à avoir :

$$V_{in} = \dots + \langle \vec{\theta}, \vec{X} \rangle + \dots$$

Ainsi $\vec{X} = (0, 0, 1, 0, 0, 0, 0, 0, 0)$ désignera un cadre...

Dès lors, le système utilisé pour le codage ne rentre plus en compte numériquement.

¹Catégorie socio-professionnelle

identifiant	CSP
1	agriculteur
2	artisans, commerçants, chefs d'entreprises
3	cadres, professions intellectuelles supérieures
4	professions intermédiaires
5	employés
6	ouvriers
7	retraités
8	autres, sans activité prof.
9	étudiants, scolaires

TAB. 2.1 – Exemple de variable polytomique non ordonnée

2.2.4 Le modèle Logit

Compte tenu du caractère aléatoire des fonctions d'utilité, le modèle donnera pour chaque mode de transport la probabilité qu'un individu le choisisse. Dérivée de la théorie micro-économique, la procédure de choix se fonde ainsi sur le concept *d'utilité aléatoire* : nous ferons l'hypothèse qu'un individu choisira l'alternative qui lui semble la meilleure. En effet dans la pratique, les individus ne choisissent pas toujours ce qui est objectivement le meilleur pour eux.

Notons :

$P_n(i)$: la probabilité que l'individu n choisisse le mode i.

D'après l'hypothèse précédente, il vient :

$$P_n(i) = P(U_{in} > U_{jn} \forall j \neq i)$$

\Downarrow

$$P_n(i) = P(\epsilon_{jn} - \epsilon_{in} < V_{in} - V_{jn} \forall j \neq i)$$

Le modèle Logit repose sur une hypothèse sur la distribution de la composante aléatoire. On suppose que les ϵ sont iid et suivent une loi de Weibull de fonction de distribution :

$$\mathcal{F}(\epsilon) = e^{-e^{-\epsilon}}$$

Cette loi est similaire à une loi normale et permet d'énoncer le théorème suivant :

Théorème 1 *Sous les hypothèses décrites ci-dessus, $P_n(i)$ s'écrit comme un modèle Logit multinomial :*

$$P_n(i) = \frac{e^{V_{in}}}{\sum_j e^{V_{jn}}} \quad (2.4)$$

Preuve Voir ([2],annexe 2)

Remarque :

Pour ne pas négliger les variables non prises en compte dans la spécification des V_{in} , nous sommes amenés à introduire un terme constant pour chacune des utilités. On a alors :

$$V_{in} = \theta_{i_0} + \theta_{i_1} X_{i_1n} + \dots$$

Arbitrairement, nous prendrons la constante associée à l'utilité ML égale à 0. En effet d'après l'équation (2.4), on montre rapidement que le terme constant est défini à une translation près. Le logiciel crée ajoute ces constantes systématiquement par défaut.

2.2.5 Estimation des paramètres

Une fois faite la spécification des variables, il s'agit d'estimer la valeur des paramètres des fonctions d'utilité. La méthode employée est celle du maximum de vraisemblance. Nous allons chercher les θ tels que la situation observée (échantillon extrait de l'enquête ménage) ait la plus grande probabilité de se produire, connaissant la loi du modèle (ici le Logit).

La vraisemblance de l'échantillon s'écrit :

$$\mathcal{L}(\vec{X}, \vec{\theta}) = \prod_n \left(\prod_j (P_n(i))^{g_{jn}} \right) \quad (2.5)$$

avec :

$$- g_{jn} = \begin{cases} 1 & \text{si l'individu } n \text{ choisit le mode } j, \\ 0 & \text{sinon.} \end{cases}$$

$$- \theta = \underbrace{(\theta_{1_0}, \dots, \theta_{1_{k_1}})}_{U_{VP}}, \underbrace{(\theta_{2_0}, \dots, \theta_{2_{k_2}})}_{U_{TC}}, \underbrace{(\theta_{3_1}, \dots, \theta_{3_{k_3}})}_{U_{ML}}$$

et on cherche :

$$\text{Max}_{\theta} \mathcal{L}(\vec{X}, \vec{\theta}) \Leftrightarrow \text{Max}_{\theta} \ln \mathcal{L}(\vec{X}, \vec{\theta})$$

Il s'agit d'un problème non linéaire sans contraintes.

Soit $\Gamma(\theta) = \ln \mathcal{L}(\vec{X}, \vec{\theta})$, la log-vraisemblance. On a alors :

$$\Gamma(\theta) = \ln \left(\prod_n \left(\prod_j (P_{nj}(\theta))^{g_{jn}} \right) \right) = \sum_n \sum_j g_{jn} \cdot \ln (P_{nj}(\theta))$$

$$\text{avec : } P_{nj}(\theta) = \frac{e^{V_{jn}(\theta)}}{\sum_p e^{V_{pn}(\theta)}}$$

d'où :

$$\Gamma(\theta) = \sum_n \sum_j \left[g_{jn} \cdot \ln \left(\frac{e^{V_{jn}(\theta)}}{\sum_p e^{V_{pn}(\theta)}} \right) \right]$$

$$\Rightarrow \Gamma(\theta) = \sum_n \sum_j \left[g_{jn} \cdot V_{jn}(\theta) - g_{jn} \cdot \ln \left(\sum_p e^{V_{pn}(\theta)} \right) \right]$$

$$\begin{aligned} \Rightarrow \Gamma(\theta) &= \sum_n \sum_j (g_{jn} \cdot V_{jn}(\theta)) - \sum_n \sum_j \left[g_{jn} \cdot \ln \left(\sum_p e^{V_{pn}(\theta)} \right) \right] \\ \Rightarrow \Gamma(\theta) &= \sum_n \sum_j (g_{jn} \cdot V_{jn}(\theta)) - \sum_n \ln \left(\sum_p e^{V_{pn}(\theta)} \right) \underbrace{\sum_j g_{jn}}_{=1} \end{aligned}$$

Au total :

$$\Gamma(\theta) = \sum_n \sum_j (g_{jn} \cdot V_{jn}(\theta)) - \sum_n \ln \left(\sum_p e^{V_{pn}(\theta)} \right) \quad (2.6)$$

Nous cherchons donc à maximiser la fonction Γ telle qu'elle est formulée dans l'équation (2.6) en fonction des paramètres θ .

Théorème 2 Dans le cas d'un Logit multinomial, la log-vraisemblance est concave.

Preuve Voir [5], page 16.

Cette propriété est fondamentale, elle garantit l'unicité de la solution au problème. Nous allons donc chercher à résoudre $\nabla\Gamma(\theta) = 0$, condition nécessaire à l'optimalité de la solution.

$$\nabla\Gamma(\theta) = 0 \Leftrightarrow \frac{\partial\Gamma(\theta)}{\partial\theta_{i_k}} = 0 \quad \forall i, k$$

Déterminons les dérivées partielles :

$$\begin{aligned} \frac{\partial\Gamma(\theta)}{\partial\theta_{i_k}} &= \frac{\partial}{\partial\theta_{i_k}} \left(\sum_n \sum_j (g_{jn} \cdot V_{jn}(\theta)) - \sum_n \ln \left(\sum_p e^{V_{pn}(\theta)} \right) \right) \\ &= \sum_n \left(\frac{\partial}{\partial\theta_{i_k}} \left(\sum_j (g_{jn} \cdot V_{jn}(\theta)) \right) \right) - \sum_n \left(\frac{\partial}{\partial\theta_{i_k}} \left(\ln \left(\sum_p e^{V_{pn}(\theta)} \right) \right) \right) \\ &= \sum_n (g_{in} X_{i_k n}) - \sum_n \frac{\frac{\partial}{\partial\theta_{i_k}} \left(\sum_p e^{V_{pn}(\theta)} \right)}{\sum_p e^{V_{pn}(\theta)}} \end{aligned}$$

d'où :

$$\frac{\partial\Gamma(\theta)}{\partial\theta_{i_k}} = \sum_n (g_{in} X_{i_k n}) - \sum_n \frac{X_{i_k n} \cdot e^{V_{in}(\theta)}}{\sum_p e^{V_{pn}(\theta)}} \quad (2.7)$$

Remarque :

- θ_{i_k} est un paramètre qui peut correspondre à :
 - une variable continue,
 - une variable à modalités,

– la constante de l'utilité TC ou de l'utilité VP.

La formule 2.7 est valable dans le premier cas.

Dans le deuxième la formule est la même, seulement, $X_{i_k n}$ vaut 1 si l'individu n a pour caractéristique la modalité qui correspond à θ_{i_k} , 0 sinon.

Dans le dernier cas, $X_{i_k n}$ vaut 1.

2.2.6 Résolution numérique

La méthode employée pour maximiser la vraisemblance est celle de Fletcher-Reeves. Nous en rappelons l'algorithme :

1. Etape 0 :

Fixer θ^0 , calculer $g_0 = \nabla\Gamma(\theta^0)$

poser $m_0 = g_0$, pour définir une direction de montée.

2. Etape k :

$\theta^{k+1} = \theta^k + \lambda_k m_k$ où λ_k maximise $\mathcal{G}(\lambda) = \Gamma(\theta^k + \lambda m_k)$

Calculer $m_{k+1} = g_{k+1} + \beta_k m_k$ avec $\begin{cases} g_{k+1} = \nabla\Gamma(\theta^{k+1}) \\ \beta_k = \frac{\|g_{k+1}\|^2}{\|g_k\|^2} \end{cases}$, pour déterminer

une direction de montée conjuguée à la précédente.

Cet algorithme nécessite une procédure de maximisation 1-D afin de déterminer l'optimum de \mathcal{G} à l'étape k. On supposera que \mathcal{G} est unimodale afin d'appliquer une méthode dichotomique, en effet :

Propriété Si \mathcal{G} est unimodale sur un intervalle $[A, B]$, alors elle admet un maximum $\bar{\lambda}$ tel que $\forall \lambda_1, \lambda_2 \in [A, B]$ avec $\lambda_1 < \lambda_2$ on ait :

$$\mathcal{G}(\lambda_1) < \mathcal{G}(\lambda_2) \Rightarrow \lambda_2 > \bar{\lambda}$$

$$\mathcal{G}(\lambda_1) > \mathcal{G}(\lambda_2) \Rightarrow \lambda_1 < \bar{\lambda}$$

Algorithme :

1. Etape 0 :

Partir d'un intervalle $[a_0, b_0]$.

Calculer $c_0 = \frac{a_0 + b_0}{2}$, $d_0 = \frac{a_0 + c_0}{2}$, $e_0 = \frac{c_0 + b_0}{2}$.

2. Etape k :

Par la propriété d'unimodalité, on élimine deux des quatre sous intervalles.

2.2.7 Qualité du modèle

Pour apprécier la qualité du modèle, on introduit un indice comparable à un coefficient de corrélation et littéralement donné par :

$$\rho^2 = 1 - \frac{\Gamma(\tilde{\theta})}{\Gamma(0)} \quad (2.8)$$

avec $\left\{ \begin{array}{l} \Gamma(\tilde{\theta}) \text{ la valeur de la log-raisemblance obtenue pour } \tilde{\theta} \text{ solution du modèle.} \\ \Gamma(0) \text{ où tous les paramètres sont nuls sauf pour les termes constants.} \end{array} \right.$

Chapitre 3

Le composant informatique

Le logiciel a été écrit sous DELPHI 5 en langage PASCAL. Il utilise la programmation orientée objet et la gestion dynamique de la mémoire à l'aide de pointeurs. Les sections suivantes présentent les types de données élaborés pour mettre en place les modèles que nous avons étudiés au chapitre 2.

3.1 La classe Tmatrice

Dérivée de la classe TForm, la classe Tmatrice hérite de toutes ses structures. Des méthodes et des propriétés viennent compléter ces dernières. On peut citer la transposition, le calcul des marges . . . , en ce qui concerne les méthodes ; un grille pour le stockage des données pour les propriétés.

De plus, un tableau conserve l'adresse mémoire de chaque matrice créée, ainsi au cours de l'exécution on peut faire appel à n'importe laquelle.

3.2 Les vecteurs "tetas"

L'estimation des paramètres des fonctions d'utilité est réalisée par la méthode du maximum de vraisemblance, que l'on résoud numériquement à l'aide de l'algorithme de Fletcher-Reeves. A la base, cet algorithme nécessite :

- Une structure pour la variable θ .
- Des outils algébriques pour les opérations élémentaires sur ces variables comme l'addition de deux vecteurs, la multiplication d'un vecteur par un scalaire . . .
- Des outils fonctionnels : le calcul de $\Gamma(\theta)$ et de $\nabla\Gamma(\theta)$.

θ doit être un vecteur de taille non définie. On emploie donc un tableau dynamique dont chaque cellule contient toutes les informations propres à chaque θ_i .

Les cellules sont composées des champs suivants :

place correspond au numéro du champ dans la base.

utilite attribue 1, 2 ou 3 si ce θ_i appartient respectivement à l'utilité VP, TC ou ML.

valeur valeur du paramètre à estimer.

modal =1 si la variable aléatoire correspondante est à modalités.

modal_val valeur de la modalité si c'est le cas.

test validation statistique après estimation(non implémenté).

Une fois faite la spécification des fonctions d'utilité par l'utilisateur, le programme prends en charge la construction du vecteur θ . Il alloue la taille nécessaire en mémoire et associe un pointeur sur l'adresse de cette structure dynamique. Ainsi, les procédures ne nécessitent en paramètres que les adresses des variables.

3.3 Manuel d'utilisation du logiciel

```
\begin{figure}[htbp]
\begin{center}
\includegraphics[417pt,249pt]{menu.eps}
\end{center}
\caption{Menu principal}
\label{img:menu}
\end{figure}
```

```
\begin{figure}[htbp]
\begin{center}
\includegraphics[450pt,432pt]{matrice.eps}
\end{center}
\caption{Exemple de matrice}
\label{img:matrice}
\end{figure}
```

```
\begin{figure}[htbp]
\begin{center}
\includegraphics[450pt,416pt]{repmo.d.eps}
\end{center}
\caption{Spécification manuelle des fonctions d'utilité.}
\label{img:repmo.d}
\end{figure}
```

Chapitre 4

Essais numériques

Pour juger de la pertinence du modèle désagrégé, des tests ont été effectués sur des données extraites d'une enquête ménage faite à Bordeaux en 1998. L'intérêt de ce chapitre n'est pas de réaliser une étude complète mais juste de vérifier comment le logiciel se comporte si on lui demande d'expliquer les choix modaux selon des critères triviaux. Entre autre, nous écrirons les utilités VP, TC et ML respectivement en fonction des variables :

- le voyageur possède le permis B,
- il a un abonnement TC,
- la longueur du déplacement qu'il effectue.

La figure ?? présente les résultats.

Conclusion

Annexe A

Utilisation d'EXCEL pour estimer un modèle bilinéaire

Mise en forme des données

On rappelle la forme générale du modèle :

$$y = nx_1 - \beta x_2 + p$$

Dans cet exemple, les déplacements ont été comptabilisés par tranches de 2.5 minutes.

	A	B	C	D
	x_1	$x_2 = \ln(x_1)$	y_{obs}	$y = \ln(y_{\text{obs}})$
1	1.25	0.22314355	2785	7.93200315
2	3.75	1.32175584	2455	7.80588204
3	6.25	1.83258146	4943	8.50572771
4	8.75	2.1690537	3584	8.18423477
5	11.25	2.42036813	3022	8.01367414
6	13.75	2.62103882	2339	7.75747877
7	16.25	2.78809291	1192	7.08338785
8	18.75	2.93119375	447	6.10255859
9	21.25	3.0563569	658	6.48920493
10	23.75	3.16758253	771	6.64768837
11	26.25	3.26766599	241	5.48479693
12	28.75	3.35863777	236	5.46383181
13	31.25	3.44201938	29	3.36729583

TAB. A.1 – Disposition des données sous EXCEL

Définition de la fonction DROITEREG

Cette formule s'utilise comme suit :

$$\text{DROITEREG}(y,x,\text{VRAI},\text{VRAI}) \tag{A.1}$$

où :

y désigne la plage contenant les y observés, D1 :D13 dans notre exemple,

x désigne les plages contenant les x_1 et les x_2 , ici A1 :B13

options mettre VRAI pour calculer les tests statistiques.

DROITEREG est une "formule matricielle" qui s'applique à une plage de cellules.

Les étapes pour l'utiliser sont les suivantes :

1. Définir à l'aide de la souris une plage de 3 colonnes par 5 lignes,
2. taper la formule (A.1) dans l'emplacement réservé sous EXCEL,
3. appuyer sur "Ctrl+shift+Entrée".

Présentation des résultats

La fonction donne en résultats l'estimation des paramètres et des tests statistiques. Se référer à l'aide EXCEL directement pour plus d'informations ...

n	β	p
erreur type de n	de β	de p
ρ^2
...
...

TAB. A.2 – Disposition des résultats

Attention : on rappelle que $p = \ln(\alpha) \rightarrow \alpha = e^p$

Annexe B

Code source

Programme principal

```
{
*****
*
**                Pierre CHIGNAC
***                Stage de DESS
***                Avirl-Septembre 2000
**                CETE du SUD-OUEST --- South-West CITY
*****
}

program modele_trafic_urbain;

uses
  Forms,
  mod_grav in 'mod_grav.pas' {Form1},
  matrix in 'matrix.pas' {matrice},
  param in 'param.pas' {param_f},
  matrix_stat in 'matrix_stat.pas' {Form2},
  repartition in 'repartition.pas' {Form3},
  repartition_utils in 'repartition_utils.pas' {Form4};

{$R *.RES}

//-----
begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.CreateForm(Tparam_f, param_f);
  Application.CreateForm(TForm2, Form2);
  Application.CreateForm(TForm3, Form3);
  Application.CreateForm(TForm4, teta_view);
  Application.Run;
end.
```

Unité de gestion matricielle

```
unit matrix;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Grids, Menus, Db, DBTables, DBGrids, ComCtrls, ToolWin, ExtCtrls,
  StdCtrls, Math, DbiProcs, DbiTypes, DbiErrs;

type

Tmatrice = class(TForm)
  StringGrid1: TStringGrid;
  MainMenu1: TMainMenu;
  Fichiers1: TMenuItem;
  Ouvrir1: TMenuItem;
  N1: TMenuItem;
  Quitter1: TMenuItem;
  OpenFileDialog1: TOpenDialog;
  Table1: TTable;
  ToolBar1: TToolBar;
  ToolButton1: TToolButton;
  ToolButton2: TToolButton;
  Enregistreurs1: TMenuItem;
  Enregister1: TMenuItem;
  N2: TMenuItem;
  db: TDatabase;
end;
```

```

ComboBox1: TComboBox;
SaveDialog1: TSaveDialog;
ToolButton4: TToolButton;

//procedure du menu
procedure Ouvrir1Click(Sender: TObject);
procedure Quitter1Click(Sender: TObject);
procedure Enregistresous1Click(Sender: TObject);
//transposer une matrice
procedure Transpose(Sender: TObject);
//crée un matrice avec les marges
procedure VectoriseMarges(Sender: TObject);
//renvoie la valeur réelle en position (i,j)
function val(i,j : integer) : real;
//détermine les marges de la matrice
procedure CalculeMarges;
//multiplie la diagonale par un nombre
procedure ToolButton4Click(Sender: TObject);
//ouvre un table si une base est ouverte
procedure ComboBox1Change(Sender: TObject);
//recalcule les marges sur modification
procedure StringGrid1KeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
//adapte la taille de la grille et du formulaire
procedure FormResize(Sender: TObject);
procedure Enregister1Click(Sender: TObject);
private

public
  nc : integer;//taille de la matrice-nombre de colonnes
  nl : integer;//nombre de lignes
end;
//fonction qui renvoie 1 si deux matrices sont de meme dimension
function compare(mat1,mat2 : Tmatrice) : boolean;

implementation

uses mod_grav;

//-----
function compare(mat1,mat2 : Tmatrice) : boolean;
begin
  if ((mat1.nc<>mat2.nc) OR (mat1.nl<>mat2.nl))
  then compare := FALSE
  else compare := TRUE;
end;

{$R *.DFM}

//-----
//-----
//          gestionnaire matriciel
//-----
//enrees-sorties
//-----
//inscrit dans TableListe les tables de la base ACCESS ouverte
procedure fDbiOpenTableList(hTmpDb: hDBIdb; TableList: TStrings);

var
  hCursor : hDBICur;
  ListDesc : TBLBaseDesc;
begin
  Check(DbiOpenTableList(hTmpDb, False, False, '*.mdb', hCursor));
  TableList.Clear;
  while (DbiGetNextRecord(hCursor, dbiNOLOCK, @ListDesc, nil) = dbiErr_None) do
    TableList.Add(ListDesc.szName);
  end;
end;
//-----
procedure Tmatrice.Ouvrir1Click(Sender: TObject);

begin
  if OpenDialog1.Execute then
  begin
    //definition de l'Alias pour la base access
    db.DatabaseName:=caption;
    db.drivername:='MSACCESS';
    db.params.clear;
    db.params.add('LANGDRIVER=Access General');
    db.params.add('DATABASE NAME='+OpenDialog1.FileName);
    db.params.add('OPEN MODE=READ/WRITE');
    db.params.add('SYSTEM DATABASE=');
    db.params.add('PASSWORD=');
    db.Connected:=True;
    Table1.databasesname:=caption;
    ComboBox1.Text:='Choisir une table';
    fDbiOpenTableList(db.handle, ComboBox1.Items);
  end;
end;
//-----
procedure Tmatrice.Quitter1Click(Sender: TObject);
begin
  close;
end;
//-----
procedure Tmatrice.Enregistresous1Click(Sender: TObject);
begin

```

```

SaveDialog1.Execute;

end;
//-----
procedure Tmatrice.ComboBox1Change(Sender: TObject);
var
  i,j : integer;
  temp : real;
begin
  Table1.TableName:=ComboBox1.Text;
  Table1.Active:=True;
  //la table ouverte est stokee ds une grille
  nc:=Table1.FieldList.count;
  StringGrid1.ColCount:=nc+2;
  nl:=Table1.RecordCount;
  StringGrid1.RowCount:=nl+2;
  for i := 1 to nl do
    begin
      for j := 1 to nc do
        begin
          //on remplit les trous eventuellement
          if (Table1.Fields[j-1].Value = NULL)
            then temp:=0
             else temp:=Table1.Fields[j-1].Value;
          StringGrid1.Cells[j+1,i+1]:=floattostr(temp);
        end;
      Table1.Next;
    end;
  Table1.Active:=False;
  db.connected:=False;
  CalculeMarges;
end;
//-----
//operations algebriques
//-----
procedure Tmatrice.Transpose(Sender: TObject);
var
  maxi : integer;
  i,j : integer;
  temp : string;
begin
  maxi:=max(nl,nc);
  StringGrid1.RowCount:=maxi+2;
  StringGrid1.ColCount:=maxi+2;
  for i := 1 to maxi+2 do
    begin
      for j := 0 to (i-1) do
        begin
          temp:=StringGrid1.Cells[i,j];
          StringGrid1.Cells[i,j]:=StringGrid1.Cells[j,i];
          StringGrid1.Cells[j,i]:=temp;
        end;
      end;
    StringGrid1.RowCount:=nc+2;
    StringGrid1.ColCount:=nl+2;
    nc:=nl;nl:=StringGrid1.RowCount-2;
  end;
//-----
procedure Tmatrice.CalculeMarges;
var
  i,j : integer;
  sigma : real;
begin
  sigma:=0;
  for j := 1 to nc do
    begin
      for i := 1 to nl do
        sigma := sigma+strtofloat(StringGrid1.Cells[j+1,i+1]);
        StringGrid1.Cells[j+1,0]:=floattostr(sigma);
        sigma:=0;
      end;
      for i := 1 to nl do
        begin
          for j := 1 to nc do
            sigma := sigma+strtofloat(StringGrid1.Cells[j+1,i+1]);
            StringGrid1.Cells[0,i+1]:=floattostr(sigma);
            sigma:=0;
          end;
          for i := 1 to nl do
            begin
              sigma:=sigma+strtofloat(StringGrid1.Cells[0,i+1]);
            end;
            StringGrid1.Cells[0,0]:=floattostr(sigma);
            for i := 1 to nc do StringGrid1.Cells[i+1,1]:=inttostr(i);
            for i := 1 to nl do StringGrid1.Cells[1,i+1]:=inttostr(i);
          end;
        //-----
        procedure Tmatrice.VectoriseMarges(Sender: TObject);
        var
          i : integer;
          matos : Tmatrice;
        begin
          if (nl<>nc) then ShowMessage('La matrice n''est pas carrée !!!')
            else
              begin
                Application.CreateForm(Tmatrice,matos);
                matos.caption:='Marges de : '+caption;
                Form1.newmat(matos);
              end;
            end;
          end;

```



```

      matos.nc:=2;
      matos.nl:=nl;
      matos.StringGrid1.RowCount:=nc+2;
      matos.StringGrid1.ColCount:=4;
      for i:= 1 to nl do
      begin
        matos.StringGrid1.Cells[1,i+1]:=inttostr(i);
        matos.StringGrid1.Cells[2,i+1]:=StringGrid1.Cells[0,i+1];
        matos.StringGrid1.Cells[3,i+1]:=StringGrid1.Cells[i+1,0];
      end;
      matos.CalculeMarges;
      matos.show;
    end;
  end;
end;
//-----
function Tmatrice.val(i,j : integer) : real;
begin
  val:=StrToFloat(StringGrid1.Cells[j+1,i+1]);
end;
//-----
procedure Tmatrice.ToolButton4Click(Sender: TObject);
var
  i : integer;
  temp : real;
  rep : String;
begin
  if (nl<>nc) then ShowMessage('La matrice n'est pas carrée !!!')
  else
  begin
    rep:=InputBox('Multiplier la diagonale par : ','','');
    if (rep <> '') then
    begin
      for i := 1 to nc do
      begin
        temp:=StrtoFloat(rep)*StrtoFloat(StringGrid1.Cells[i+1,i+1]);
        StringGrid1.Cells[i+1,i+1]:=floattostr(temp);
      end;
      calculemarges;
    end;
  end;
end;
//-----
procedure Tmatrice.StringGrid1KeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  calculemarges;
end;
//-----
procedure Tmatrice.FormResize(Sender: TObject);
begin
  StringGrid1.Height:=ClientHeight-33;
  StringGrid1.Width:=ClientWidth-2;
end;
//-----
procedure Tmatrice.Enregister1Click(Sender: TObject);
var i,j:integer;
begin
  db.connected:=true;
  table1.active:=true;
  table1.First;
  for i := 1 to nl do
  begin
    Table1.edit;
    for j := 1 to nc do
    begin
      Table1.Fields[j-1].value:=strtoFloat(stringgrid1.Cells[j+1,i+1]);
    end;
    table1.post;
    table1.next;
  end;
  table1.active:=false;
  db.connected:=false;
end;
//-----
end.

```

Module du modèle gravitaire

```

unit mod_grav;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, Grids, StdCtrls ,Math ,matrix, param, matrix_stat, ExtCtrls,
  repartition;

const

  maxi = 20;

type

```

```

TForm1 = class(TForm)
  MainMenu1: TMainMenu;
  Fichiers1: TMenuItem;
  Quitter1: TMenuItem;
  matrice1: TMenuItem;
  creer1: TMenuItem;
  Modle1: TMenuItem;
  Distribution1: TMenuItem;
  Label1: TLabel;
  marges: TLabel;
  cbox: TListBox;
  mbox: TListBox;
  Parametres1: TMenuItem;
  Fonctionimpdance1: TMenuItem;
  Statistiques1: TMenuItem;
  Synchroniser1: TMenuItem;
  Rpartitionmodale1: TMenuItem;
  eclater1: TMenuItem;
  eclater2: TMenuItem;
  multiplier1: TMenuItem;
  procedure Quitter1Click(Sender: TObject);
  procedure creer1Click(Sender: TObject);
  procedure Distribution1Click(Sender: TObject);
  procedure Fonctionimpdance1Click(Sender: TObject);
  procedure Statistiques1Click(Sender: TObject);
  //mboxKeyDown et cboxKeyDown gèrent la suppression des matrices
  procedure mboxKeyDown(Sender: TObject; var Key: Word; Shift: TShiftState);
  procedure cboxKeyDown(Sender: TObject; var Key: Word; Shift: TShiftState);
  procedure mboxDb1Click(Sender: TObject);
  procedure cboxDb1Click(Sender: TObject);
  procedure Rpartitionmodale1Click(Sender: TObject);
  procedure eclater2Click(Sender: TObject);
  procedure multiplier1Click(Sender: TObject);
private

public
  procedure newmat(mat : Tmatrice);
end;

var
  Form1 : TForm1;
  alpha : real = 1.198037556;
  beta : real = 0.237794387;
  p : real = 7.693570373;
  liste_mat : array [0..30] of Tmatrice;

implementation

{$R *.DFM}

//-----
//-----
//boutons et outils
//-----
//-----
procedure TForm1.Quitter1Click(Sender: TObject);
begin
  Application.Terminate;
end;
//-----
procedure TForm1.newmat(mat : Tmatrice);
begin
  liste_mat[mbox.items.count]:=mat;
  mbox.items.addobject(mat.caption,mat);
  cbox.items.addobject(mat.caption,mat);
end;
//-----
procedure TForm1.creer1Click(Sender: TObject);
var
  mat : Tmatrice;
  rep : string;
begin
  Application.CreateForm(Tmatrice,mat);
  rep:=InputBox('Nom de la matrice','','');
  mat.caption:=rep;
  newmat(mat);
  mat.show;
end;

//-----
procedure TForm1.Fonctionimpdance1Click(Sender: TObject);
begin
  param_f.showmodal;
end;
//-----
procedure TForm1.Rpartitionmodale1Click(Sender: TObject);
begin
  Form3.showmodal;
end;
//-----
procedure TForm1.Statistiques1Click(Sender: TObject);
begin
  Form2.calculer_stat(liste_mat[cbox.ItemIndex],liste_mat[mbox.ItemIndex]);
end;
//-----

```

```

procedure TForm1.mboxKeyDown(Sender: TObject; var Key: Word; Shift: TShiftState);
var
  i : integer;
begin
  if ((Key=VK_DELETE) AND (mbox.ItemIndex<>-1)) then
  begin
    for i := mbox.ItemIndex to 29 do
    begin
      liste_mat[i]:=liste_mat[i+1];
    end;
    mbox.Items.Objects[mbox.ItemIndex].Free;
    mbox.Items.Delete(mbox.ItemIndex);
    end;
  end;
  //-----
procedure TForm1.cboxKeyDown(Sender: TObject; var Key: Word; Shift: TShiftState);
var
  i : integer;
begin
  if ((Key=VK_DELETE) AND (cbox.ItemIndex<>-1)) then
  begin
    for i := cbox.ItemIndex to 29 do
    begin
      liste_mat[i]:=liste_mat[i+1];
    end;
    cbox.Items.Objects[cbox.ItemIndex].Free;
    cbox.Items.Delete(cbox.ItemIndex);
    end;
  end;
  //-----
procedure TForm1.mboxDbClick(Sender: TObject);
begin
  liste_mat[mbox.ItemIndex].WindowState:=wsNormal;
end;
  //-----
procedure TForm1.cboxDbClick(Sender: TObject);
begin
  liste_mat[cbox.ItemIndex].WindowState:=wsNormal;
end;
  //-----
//-----
//
// implementation du modele gravitaire
//-----
//
// Fonction d'impédance
function f(x : real) : real;
begin
  f := power(x,alpha)*exp(-x*beta*p);
end;
  //-----
procedure TForm1.Distribution1Click(Sender: TObject);
var
  i,j,k      : integer;
  taille     : integer;
  sigma      : real;
  model      : Tmatrice;
  couts,marges : Tmatrice;
  temp       : Tmatrice;
begin
  Application.CreateForm(Tmatrice,model);
  Application.CreateForm(Tmatrice,temp);
  model.caption:='Modèle gravitaire';
  Form1.newmat(model);
  couts:=liste_mat[cbox.ItemIndex];
  marges:=liste_mat[mbox.ItemIndex];
  temp.StringGrid1.RowCount:=marges.StringGrid1.RowCount;
  temp.StringGrid1.ColCount:=marges.StringGrid1.ColCount;
  taille:=couts.StringGrid1.RowCount-2;
  model.StringGrid1.RowCount:=taille+2;
  model.StringGrid1.ColCount:=taille+2;
  model.nc:=taille;
  model.nl:=taille;
  //-----initialisations-----
  for i := 1 to taille do temp.StringGrid1.Cells[3,i+1]:='1';
  //-----calcul des Ai et Bi-----
  for k := 1 to 20 do
  begin
    for i := 1 to taille do
    begin
      sigma:=0;
      for j := 1 to taille do
      begin
        sigma:=sigma+temp.val(j,2)*marges.val(j,2)*f(couts.val(i,j));
      end;
      temp.StringGrid1.Cells[2,i+1]:=FloatToStr(1/sigma);
    end;
    for j := 1 to taille do
    begin
      sigma:=0;
      for i := 1 to taille do
      begin
        sigma:=sigma+temp.val(i,1)*marges.val(i,1)*f(couts.val(i,j));
      end;
      temp.StringGrid1.Cells[3,j+1]:=FloatToStr(1/sigma);
    end;
  end;
end;

```

```

//-----calcul des Tij=Ai*Oi*Bj*Dj*f(Cij)
for i := 1 to taille do
begin
for j := 1 to taille do
begin
model.StringGrid1.Cells[j+1,i+1]:=
//Ai      Oi      Bi      Di
floattostr(round(temp.val(i,1)*marges.val(i,1)*temp.val(j,2)*marges.val(j,2)*f(couts.val(i,j))));
end;
end;
model.calculmarges;
model.show;
temp.destroy;
end;
//-----
//proc pour jean-michel
//-----
procedure TForm1.eclater2Click(Sender: TObject);
var
source, relat, cible : Tmatrice;
rep : string;
idx : integer;
i, j : integer;
ir, jr : integer;
begin
source:=liste_mat[cbox.ItemIndex];
relat:=liste_mat[mbox.ItemIndex];
rep:=inputbox('nom de la matrice cible','','');
idx:=mbox.items.indexof(rep);
cible:=liste_mat[idx];

for i := 1 to cible.nc do
begin
ir:=strtoint( relat.stringgrid1.cells[3,i+1]);
for j := 1 to cible.nl do
begin
jr:=strtoint( relat.stringgrid1.cells[3,j+1]);
if ((ir>source.nc) or (jr>source.nl)) then
cible.StringGrid1.cells[j+1,i+1]:='1'
else
cible.StringGrid1.cells[j+1,i+1]:=source.StringGrid1.cells[ir+1,jr+1];
end;
end;
end;
//-----
procedure TForm1.multiplier1Click(Sender: TObject);
var
mat1, mat2, cible : Tmatrice;
i, j : integer;
rep : string;
idx : integer;
a, b : real;
begin
mat1:=liste_mat[cbox.ItemIndex];
mat2:=liste_mat[mbox.ItemIndex];
rep:=inputbox('nom de la matrice cible','','');
idx:=mbox.items.indexof(rep);
cible:=liste_mat[idx];

for i := 1 to cible.nc do
begin
for j := 1 to cible.nl do
begin
a:=strtofloat(mat1.StringGrid1.cells[i+1,j+1]);
b:=strtofloat(mat2.StringGrid1.cells[i+1,j+1]);
cible.StringGrid1.cells[i+1,j+1]:=floattostr(round(a*b));
end;
end;
end;
//-----
end.

```

Définitions et outils nécessaires au modèle désagrégé

```

// cette unité contient la définition de la structure nécessaire
// au modèle et les procédures de manipulation des données dynamiques
// initialisées dans l'unité 'repartition'

```

```
unit repartition_utils;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
CheckLst, StdCtrls, Dialogs, Menus, Grids;
```

```
//-----
//structure de donnée dynamique associée aux paramètres
//-----
```

```
Type
```

```
cellule = RECORD
```

```

    place : integer; //n° du champ dans la base [1..n]
    utilite : integer; //1=UVP 2=UTC 3=UML
    valeur : double; //valeur du parametre à estimer
    modal : bool; // 1=variable a valeurs discretes 0 sinon
    modal_val : integer; //valeur de la modalite correspondante
    test : double;
end;

taille_modalite = integer;

Const
    MAX = (65520 div Sizeof(cellule));
    MAX_modalite = (65520 div Sizeof(taille_modalite));

Type
    geniteur_model = ARRAY [1..MAX] of cellule;
    geniteur_taille_modalite = ARRAY [1..MAX_modalite] of taille_modalite;
//-----
//formulaire
//-----
type
    TForm4 = class(TForm)
        MainMenu1: TMainMenu;
        Fichier1: TMenuItem;
        Sauver1: TMenuItem;
        N1: TMenuItem;
        Fermer1: TMenuItem;
        StringGrid1: TStringGrid;
        SaveDialog1: TSaveDialog;
        procedure Fermer1Click(Sender: TObject);
        procedure affiche(p : pointer);
        procedure Sauver1Click(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

//-----
//variables
//-----
var
    _place_mode : integer = 1;

    nteta : integer;
    nteta_vp : integer;
    nteta_tc : integer;
    nteta_ml : integer;

    teta_view : TForm4;

    teta : ^geniteur_model;
    gk : ^geniteur_model;
    mk : ^geniteur_model;
    teta_temp : ^geniteur_model;

    modalites : ^geniteur_taille_modalite;

    lteta : double;
    lzero : double;
    compteur : integer;
//utils algebriques
procedure copie(source, cible : pointer);
procedure plus(cible, teta_1, teta_2 : pointer);
procedure fois(lambda : double; cible : pointer);
function norme(p : pointer) : double;

//utils fonctionnels
function pnj(p : pointer; u : integer) : double;
function f(p : pointer) : double;
function utilite(p : pointer; u : integer) : double;
procedure gradient(p : pointer);

implementation

uses repartition;

{$R *.DFM}
{*****
*
*
*          outils algebriques pour les tetras
*
*
*
*
*****}
procedure copie(source,cible : pointer);
var
    i : integer;
    s,c : ^geniteur_model;
begin
    s:=source;
    c:=cible;
    reallocmem(s,nteta*sizeof(cellule));
    reallocmem(c,nteta*sizeof(cellule));
    for i := 1 to nteta do
        begin
            with c^[i] do

```

```

begin
    valeur:=s^[i].valeur;
    modal:=s^[i].modal;
    modal_val:=s^[i].modal_val;
    place:=s^[i].place;
    utilite:=s^[i].utilite;
    test:=s^[i].test;
end;
end;
end;
//-----
procedure plus(cible,teta_1,teta_2 : pointer);
var
    i : integer;
    c, t1, t2 : ^geniteur_model;
begin
    c:=cible;
    t1:=teta_1;
    t2:=teta_2;
    reallocmem(c,nteta*sizeof(cellule));
    reallocmem(t1,nteta*sizeof(cellule));
    reallocmem(t2,nteta*sizeof(cellule));
    for i := 1 to nteta do c^[i].valeur:=t1^[i].valeur+t2^[i].valeur;
end;
//-----
procedure fois(lambda:double; cible : pointer);
var
    i : integer;
    c : ^geniteur_model;
begin
    c:=cible;
    reallocmem(c,nteta*sizeof(cellule));
    for i := 1 to nteta do c^[i].valeur:=c^[i].valeur*lambda;
end;
//-----
// retourne la norme euclidienne d'un teta
//-----
function norme(p : pointer) : double;
var
    i : integer;
    temp : double;
    x : ^geniteur_model;
begin
    temp:=0;
    x:=p;
    reallocmem(x,nteta*sizeof(cellule));
    for i := 1 to nteta do
        begin
            temp:=temp+sqr(x^[i].valeur);
        end;
    result:=sqrt(temp);
end;
{*****}
*
*      outils d'analyse fonctionnelle pour l'algo du gradient conjugué      *
*
*
*
*
*
*****}
//-----
//calculer l'utilite pour un mode donné
//-----
function utilite(p : pointer; u : integer) : double;
var
    i : integer;
    temp : double;
    x : ^geniteur_model;
begin
    x:=p;
    reallocmem(x,nteta*sizeof(cellule));
    temp:=0;
//pour le mode VP
    if (u=1) then
        begin
            temp:=temp+x^[1].valeur;
            for i := 2 to nteta_vp do
                begin
                    if x^[i].modal then
                        begin
                            if (x^[i].modal_val=Form3.echantillon.Fields[x^[i].place-1].Value) then
                                begin
                                    temp:=temp+x^[i].valeur;
                                end;
                            end
                        end
                    else
                        begin
                            temp:=temp+x^[i].valeur*Form3.echantillon.Fields[x^[i].place-1].Value;
                        end;
                    end;
                end;
            end;
        end;
//pour le mode TC
    if (u=2) then
        begin
            temp:=temp+x^[1+nteta_vp].valeur;
            for i := nteta_vp+2 to nteta_vp+nteta_tc do
                begin
                    if x^[i].modal then
                        begin

```

```

        if (x^[i].modal_val=Form3.echantillon.Fields[x^[i].place-1].Value) then
        begin
            temp:=temp+x^[i].valeur;
        end;
    end
    else
    begin
        temp:=temp+x^[i].valeur*Form3.echantillon.Fields[x^[i].place-1].Value;
    end;
end;
end;
//pour le ML
if (u=3) then
begin
    for i := (nteta_tc+nteta_vp+1) to nteta do
    begin
        if x^[i].modal then
        begin
            if (x^[i].modal_val=Form3.echantillon.Fields[x^[i].place-1].Value) then
            begin
                temp:=temp+teta^[i].valeur;
            end;
        end
        else
        begin
            temp:=temp+x^[i].valeur*Form3.echantillon.Fields[x^[i].place-1].Value;
        end;
    end;
end;
    utilite:=temp;
end;
//-----
//calculer Pnj(teta,utilite)
//-----
function pnj(p : pointer; u : integer) : double;
var
    temp          : double;
    uvp, utc, uml : double;
begin
    temp:=0;
    uvp:=utilite(p, 1);
    utc:=utilite(p, 2);
    uml:=utilite(p, 3);
    case u of
        1 : temp := uvp;
        2 : temp := utc;
        3 : temp := uml;
    end;
//    exp(u_mode_choisi)
// renvoie -----
//    somme exp(u_modes)
    pnj:=exp(temp)/(exp(uvp)+exp(utc)+exp(uml));
end;
//-----
//calculer la vraisemblance de l'échantillon pour un teta donné.
//-----
function f(p : pointer) : double;
var
    i          : integer;
    u          : integer;
    vraisemblance : double;
    temp       : double;
begin
    Form3.echantillon.First;
    vraisemblance:=0;
    for i := 1 to Form3.echantillon.RecordCount do
    begin
        u:=Form3.echantillon.Fields[_place_mode-1].value;
        temp:=ln(pnj(p,u));
        vraisemblance:=vraisemblance+temp;
        Form3.echantillon.Next;
    end;
    result:=vraisemblance;
end;

//-----
// calculer le gradient de la vraisemblance pour un teta donné
//-----
procedure gradient(p : pointer);
var
    test : boolean;//=Vrai si i_teta est associe a l'utilite u de l'individu n
    n    : integer;//individu actif
    i_teta : integer;//compteur pour les coordonnees de teta.
    u     : integer;//utilite choisie par n
    p1,p2 : double;//les deux parties de la derivee partielle
    x     : ^geniteur_model;
    xink  : integer;
begin
    x:=p;
    reallocmem(x,nteta*sizeof(cellule));
    xink:=0;
// il faut initialiser gk
    for i_teta := 1 to nteta do gk^[i_teta].valeur:=0;
// calculer chaque composante du gradient ...
    for i_teta := 1 to nteta do
    begin
        Form3.echantillon.First;
// ... en parcourant la base

```

```

for n := 1 to Form3.echantillon.RecordCount do
begin
u:=Form3.echantillon.Fields[_place_mode-1].value;
//activer la variable test si le teta actif est ds l'utilite du mode
// que n a choisi
if u=x^[i_teta].utilite then test:=True else test:=False;
//xink
if x^[i_teta].modal then
begin
if x^[i_teta].modal_val=Form3.echantillon.Fields[x^[i_teta].place-1].Value
then xink:=1
else xink:=0;
end;
//calcul de p1
if test then
begin
if ((i_teta=1) or (i_teta=nteta_vp+1)) then p1:=1
else
if x^[i_teta].modal then p1:=xink
else p1:=Form3.echantillon.Fields[x^[i_teta].place-1].Value;
end
else p1:=0;
//calcul de p2
if ((i_teta=1) or (i_teta=nteta_vp+1))
then p2:=exp(utilite(x,x^[i_teta].utilite))
else
begin
if x^[i_teta].modal
then p2:=xink*exp(utilite(x,x^[i_teta].utilite))
else p2:=Form3.echantillon.Fields[x^[i_teta].place-1].Value
*exp(utilite(x,x^[i_teta].utilite));
end;
p2:=p2/(exp(utilite(x,1))+exp(utilite(x,2))+exp(utilite(x,3)));
gk^[i_teta].valeur:=gk^[i_teta].valeur+p1-p2;
Form3.echantillon.Next;
end;
end;
end;
//*****
//*****
//*****
procedure TForm4.Fermer1Click(Sender: TObject);
begin
Close;
end;
//-----
procedure TForm4.affiche(p : pointer);
var
i : integer;
x : ^geniteur_model;
begin
x:=p;
reallocmem(x,nteta*sizeof(cellule));
StringGrid1.RowCount:=nteta+3;
StringGrid1.Colcount:=7;
StringGrid1.Cells[1,0]:='valeur';
StringGrid1.Cells[2,0]:='n° du Champ';
StringGrid1.Cells[3,0]:='utilité';
StringGrid1.Cells[4,0]:='variable modale';
StringGrid1.Cells[5,0]:='valeur de la modalité';
StringGrid1.Cells[6,0]:='test';
for i := 1 to nteta do
begin
StringGrid1.Cells[0,i]:='teta '+inttostr(i);
StringGrid1.Cells[1,i]:=floattostr(x^[i].valeur);
if x^[i].place=0
then StringGrid1.Cells[2,i]:='constante'
else
StringGrid1.Cells[2,i]:=Form3.echantillon.Fields[x^[i].place-1].FullName;
case x^[i].utilite of
1 : StringGrid1.Cells[3,i]:='vp';
2 : StringGrid1.Cells[3,i]:='tc';
3 : StringGrid1.Cells[3,i]:='ml';
end;
if x^[i].modal then
begin
StringGrid1.Cells[4,i]:='oui';
StringGrid1.Cells[5,i]:=inttostr(x^[i].modal_val);
end
else StringGrid1.Cells[4,i]:='non';
StringGrid1.Cells[6,i]:=floattostr(x^[i].test);
end;
StringGrid1.Cells[1,nteta+1]:='vraisemblance';
StringGrid1.Cells[2,nteta+1]:='vraisemblance(0)';
StringGrid1.Cells[3,nteta+1]:='rho2';
StringGrid1.Cells[4,nteta+1]:='itérations';
StringGrid1.Cells[1,nteta+2]:=floattostr(1teta);
StringGrid1.Cells[2,nteta+2]:=floattostr(lzero);
StringGrid1.Cells[3,nteta+2]:=floattostr(1-1teta/lzero);
StringGrid1.Cells[4,nteta+2]:=floattostr(compteur);
end;
//-----
procedure TForm4.Sauver1Click(Sender: TObject);
var
fichier : textfile;
i,j : integer;
begin
if savedialog1.execute then

```



```

begin
  AssignFile(fichier,savedialog1.FileName);
  Rewrite(fichier);
  for i := 1 to Stringgrid1.ColCount+1 do
  begin
    for j := 1 to Stringgrid1.RowCount do write(fichier,stringgrid1.cells[j-1,i-1]+' ');
    writeln(fichier,'');
  end;
  closefile(fichier);
end;
end;
//-----
end.

```

Unité du modèle désagrégé de répartition modale

```

unit repartition;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Db, DBTables, Grids, DBGrids, Menus, StdCtrls, DbiProcs, DbiTypes, DbiErrs,
  matrix, repartition_utils, ExtCtrls, CheckLst;

type
  TForm3 = class(TForm)
    MainMenu: TMainMenu;
    Fichiers1: TMenuItem;
    Ouvrir1: TMenuItem;
    N1: TMenuItem;
    Quitter1: TMenuItem;
    echantillon: TTable;
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
    OpenDialog1: TOpenDialog;
    db: TDatabase;
    ComboBox1: TComboBox;
    Bevel1: TBevel;
    Label1: TLabel;
    Edit1: TEdit;
    Label2: TLabel;
    Edit2: TEdit;
    Bevel2: TBevel;
    Bevel3: TBevel;
    Bevel4: TBevel;
    Bevel5: TBevel;
    Bevel6: TBevel;
    Bevel7: TBevel;

    mList: TListBox;
    vplist: TListBox;
    tcList: TListBox;

    vp: TRadioButton;
    tc: TRadioButton;
    ml: TRadioButton;

    Button1: TButton;
    Button2: TButton;
    CheckListBox1: TCheckListBox;
    Parametres1: TMenuItem;
    Estimation1: TMenuItem;
    parametrespardfaut1: TMenuItem;
    Afficherteta1: TMenuItem;
    procedure Ouvrir1Click(Sender: TObject);
    procedure ComboBox1Change(Sender: TObject);
    procedure Quitter1Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Estimation1Click(Sender: TObject);
    procedure parametrespardfaut1Click(Sender: TObject);
    procedure Afficherteta1Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form3: TForm3;

  function maximisation : double;

implementation

{$R *.DFM}

//*****
//-----
//-----

```

```

//ouverture de la base et choix de la table.
//-----
//-----
procedure fDbiOpenTableList(hTmpDb: hDBidb; TableList: TStrings);

var
  hCursor : hDBICur;
  ListDesc : TBLBaseDesc;
begin
  Check(DbiOpenTableList(hTmpDb, False, False, '*.mdb', hCursor));
  TableList.Clear;
  while (DbiGetNextRecord(hCursor, dbiNOLOCK, @ListDesc, nil) = dbiErr_None) do
    TableList.Add(ListDesc.szName);
end;
//-----
procedure TForm3.Ouvrir1Click(Sender: TObject);
begin
  if OpenDialog1.Execute then
  begin
    //definition de l'Alias pour la base access
    db.DatabaseName:=caption;
    db.drivename:='MSACCESS';
    db.params.clear;
    db.params.add('LANGDRIVER=Access General');
    db.params.add('DATABASE NAME='+OpenDialog1.FileName);
    db.params.add('OPEN MODE=READ/WRITE');
    db.params.add('SYSTEM DATABASE=');
    db.params.add('PASSWORD=');
    db.Connected:=True;
    echantillon.databasename:=caption;
    ComboBox1.Text:='Choisir une table';
    fDbiOpenTableList(db.handle, ComboBox1.Items);
  end;
end;
//-----
procedure TForm3.ComboBox1Change(Sender: TObject);
var
  i : integer;
begin
  echantillon.Active:=False;
  echantillon.TableName:=ComboBox1.Text;
  echantillon.Active:=True;
  Edit1.Text:=IntToStr(echantillon.RecordCount);
  Edit2.Text:=IntToStr(echantillon.FieldCount);
  CheckListBox1.Items.Clear;
  for i := 1 to echantillon.FieldCount do
    CheckListBox1.Items.Add(echantillon.Fields[i-1].FullName);
end;
//-----
procedure TForm3.Quitter1Click(Sender: TObject);
begin
  Edit1.Text:='';
  Edit2.Text:='';
  ComboBox1.text:='';
  CheckListBox1.Items.clear;
  ComboBox1.items.Clear;
  tclist.items.clear;
  mllist.items.clear;
  vplist.items.clear;
  echantillon.Active:=False;
  db.Connected:=False;
  Application.Terminate;
end;
//-----
procedure TForm3.Button1Click(Sender: TObject);
begin
  if (vp.Checked and (checkboxlistbox1.itemindex<>-1))then
  begin
    vplist.Items.add(CheckListBox1.Items.Strings[CheckListBox1.itemindex]);
  end;
  if (tc.Checked and (checkboxlistbox1.itemindex<>-1))then
  begin
    tclist.Items.add(CheckListBox1.Items.Strings[CheckListBox1.itemindex]);
  end;
  if (ml.Checked and (checkboxlistbox1.itemindex<>-1))then
  begin
    mllist.Items.add(CheckListBox1.Items.Strings[CheckListBox1.itemindex]);
  end;
end;
//-----
procedure TForm3.Button2Click(Sender: TObject);
begin
  if ((vp.Checked) and (vplist.itemindex<>-1)) then
  begin
    vplist.Items.Delete(vplist.ItemIndex);
  end;
  if ((tc.Checked) and (tclist.itemindex<>-1)) then
  begin
    tclist.Items.Delete(tclist.ItemIndex);
  end;
  if ((ml.Checked) and (mllist.itemindex<>-1)) then
  begin
    mllist.Items.Delete(mllist.ItemIndex);
  end;
end;
{*****
*
*
*
*
*****}

```

```

*          estimation des parametres du modele          *
*
*
*
*****}
function maximisation : double;
var
  epsilon, delta : double;
  a, b, c, d, e : double;
  fc, fd, fe : double;
begin
  epsilon:=0.001;
  a:=-0.1; b:=0.1;c:=0;
  delta:=abs(a-b);
  while delta>epsilon do
  begin
    c:=(a+b)/2;
    d:=(a+c)/2;
    e:=(c+b)/2;
    copie(mk, teta_temp);
    fois(c, teta_temp);
    plus(teta_temp, teta, teta_temp);
    fc:=f(teta_temp);
    copie(mk, teta_temp);
    fois(d, teta_temp);
    plus(teta_temp, teta, teta_temp);
    fd:=f(teta_temp);
    copie(mk, teta_temp);
    fois(e, teta_temp);
    plus(teta_temp, teta, teta_temp);
    fe:=f(teta_temp);
    if (fd>fc) then b:=c
    else if (fc>fe) then a:=c
    else
      begin
        a:=d;
        b:=e;
      end;
    delta:=abs(a-b);
  end;
  maximisation:=c;
end;
//-----
procedure TForm3.Estimation1Click(Sender: TObject);
var
  i,j,k,l : integer;
  rep : string;
  lambda : double;
  beta : double;
  norme_gk, norme_teta : double;
begin
  //formatage de la variable *modalites*
  reallocmem(modalites,CheckBox1.Items.Count*sizeof(taille_modalite));
  //pour les champ a modalite on cherche le nombre de modalites differentes
  //manuellement
  if MessageDlg('Voulez vous entrer les renseignements sur les modalites manuellement ? (conseillé)',
    mtConfirmation, [mbYes, mbNo], 0) = mrYes then
  begin
    for i := 1 to CheckBox1.Items.Count do
    begin
      if CheckBox1.Checked[i-1] then
      begin
        rep:=InputBox('Entrez le nombre de modalités :',
          'pour la variable correspondant au champ : '+echantillon.Fields[i-1].FullName,'1');
        modalites^[i]:=strtoint(rep);
      end
      else modalites^[i]:=1;
    end;
  end
  //ou brutalement
  else
  begin
    for i := 1 to CheckBox1.Items.Count do modalites^[i]:=1;
    echantillon.First;
    for j := 1 to echantillon.RecordCount do
    begin
      for i := 1 to CheckBox1.Items.Count do
      begin
        if CheckBox1.Checked[i-1] then
        begin
          if echantillon.fields[i-1].Value>modalites^[i] then
          begin
            modalites^[i]:=echantillon.fields[i-1].Value;
          end;
        end;
      end;
    end;
    echantillon.Next;
  end;
  end;
  //formatage de la variable *teta*
  nteta_vp:=1;
  nteta_tc:=1;
  nteta_ml:=0;
  for i := 1 to vplist.Items.Count do
  begin
    for j := 1 to CheckBox1.Items.Count do
    begin
      if CheckBox1.Items.strings[j-1]=vplist.Items.strings[i-1] then

```

```

begin
  if CheckListBox1.Checked[j-1]
    then nteta_vp:=nteta_vp+modalites[j]
    else inc(nteta_vp);
  end;
end;
end;
for i := 1 to tclist.Items.Count do
begin
  for j := 1 to CheckListBox1.Items.Count do
  begin
    if CheckListBox1.Items.strings[j-1]=tclist.Items.strings[i-1] then
    begin
      if CheckListBox1.Checked[j-1]
        then nteta_tc:=nteta_tc+modalites[j]
        else inc(nteta_tc);
      end;
    end;
  end;
  for i := 1 to mllist.Items.Count do
  begin
    for j := 1 to CheckListBox1.Items.Count do
    begin
      if CheckListBox1.Items.strings[j-1]=mllist.Items.strings[i-1] then
      begin
        if CheckListBox1.Checked[j-1]
          then nteta_ml:=nteta_ml+modalites[j]
          else inc(nteta_ml);
        end;
      end;
    end;
  end;
  nteta:=nteta_vp+nteta_tc+nteta_ml;
  reallocmem(teta,nteta*sizeof(cellule));
//on initialise la variable teta
teta^[1].place:=0;
teta^[1].utilite:=1;
teta^[1].valeur:=1;
teta^[1].modal:=False;
teta^[1].test:=0;
k:=2;
for i := 1 to vplist.Items.Count do
begin
  for j := 1 to CheckListBox1.Items.Count do
  begin
    if CheckListBox1.Items.strings[j-1]=vplist.Items.strings[i-1] then
    begin
      if CheckListBox1.Checked[j-1] then
      begin
        for l := 1 to modalites^[j] do
        begin
          teta^[k].place:=j;
          teta^[k].utilite:=1;
          teta^[k].valeur:=1;
          teta^[k].modal:=True;
          teta^[k].modal_val:=1;
          teta^[k].test:=0;
          inc(k);
        end;
      end
      else
      begin
        teta^[k].place:=j;
        teta^[k].utilite:=1;
        teta^[k].valeur:=1;
        teta^[k].modal:=False;
        teta^[k].test:=0;
        inc(k);
      end;
    end;
  end;
end;
teta^[k].place:=0;
teta^[k].utilite:=2;
teta^[k].valeur:=1;
teta^[k].modal:=False;
teta^[k].test:=0;
inc(k);
for i := 1 to tclist.Items.Count do
begin
  for j := 1 to CheckListBox1.Items.Count do
  begin
    if CheckListBox1.Items.strings[j-1]=tclist.Items.strings[i-1] then
    begin
      if CheckListBox1.Checked[j-1] then
      begin
        for l := 1 to modalites^[j] do
        begin
          teta^[k].place:=j;
          teta^[k].utilite:=2;
          teta^[k].valeur:=1;
          teta^[k].modal:=True;
          teta^[k].modal_val:=1;
          teta^[k].test:=0;
          inc(k);
        end;
      end
      else
      begin

```

```

        teta^[k].place:=j;
        teta^[k].utilite:=2;
        teta^[k].valeur:=1;
        teta^[k].modal:=False;
        teta^[k].test:=0;
        inc(k);
    end;
end;
end;
for i := 1 to mlist.Items.Count do
begin
    for j := 1 to CheckListBox1.Items.Count do
    begin
        if CheckListBox1.Items.strings[j-1]=mlist.Items.strings[i-1] then
        begin
            if CheckListBox1.Checked[j-1] then
            begin
                for l := 1 to modalites^[j] do
                begin
                    teta^[k].place:=j;
                    teta^[k].utilite:=3;
                    teta^[k].valeur:=1;
                    teta^[k].modal:=True;
                    teta^[k].modal_val:=1;
                    teta^[k].test:=0;
                    inc(k);
                end;
            end
            else
            begin
                teta^[k].place:=j;
                teta^[k].utilite:=3;
                teta^[k].valeur:=1;
                teta^[k].modal:=False;
                teta^[k].test:=0;
                inc(k);
            end;
        end;
    end;
end;
// formatage & initialisation des autres variables...
reallocmem(gk,nteta*sizeof(cellule));
reallocmem(mk,nteta*sizeof(cellule));
reallocmem(teta_temp,nteta*sizeof(cellule));
copie(teta,gk);
copie(teta,mk);
copie(teta,teta_temp);
// Algo de Fletcher-Reeves
dbgrid1.DataSource:=nil;
gradient(teta);//stocké dans gk
copie(gk,mk);
compteur:=0;
repeat
begin
    inc(compteur);
    norme_gk:=norme(gk);//on garde la norme de gk;
    norme_teta:=norme(teta);
    lambda:=maximisation;
    fois(lambda,mk);
    plus(teta,teta,mk);
    gradient(teta);
    beta:=sqr(norme(gk))/sqr(norme_gk);
    fois(beta,mk);
    plus(mk,gk,mk);
end;
until (abs(norme(teta)-norme_teta)<0.001);
lteta:=f(teta);
copie(teta,teta_temp);
for i := 2 to nteta do if (i<>nteta_vp+1) then teta_temp[i].valeur:=0;
lzero:=f(teta_temp);
dbgrid1.DataSource:=datasource1;
teta_view.Show;
teta_view.affiche(teta);
end;
//-----
procedure TForm3.parametrespardfaut1Click(Sender: TObject);
var
    rep : string;
begin
    rep:=InputBox('Entrez la n° du champ','correspondant au mode choisit','1');
    _place_mode:=strtoint(rep);
end;
//-----
procedure TForm3.Afficherteta1Click(Sender: TObject);
begin
    teta_view.Show;
    teta_view.affiche(teta);
end;
//-----
end.

```

Bibliographie

- [1] J. de D. Ortùzar et L. G. Willumsen. *Modeling Transport*. WILEY, deuxième édition.
- [2] Dossiers du CERTU. *Comportements de déplacements en milieu urbain : les modèles de choix discret*. Certu, Juin 1998.
- [3] A. G. Wilson. *Entropy in urban and regional modelling*. Pion, London, 1970.
- [4] Anech Pajouh. *Estimation des matrices origine–destination sur les comptages et la théorie de l’information*. Rapport INRETS n°7, Juin 1986.
- [5] Document de travail de l’INSEE. *L’économétrie et l’étude des comportements*. Document n°9606.