Application d'un modèle stochastique à la construction d'un test adaptatif informatisé en mathématiques

Rapport final d'un projet de recherche effectué dans le cadre d'une bourse de formation-recherche du Ministère de l'Education Nationale et de la Formation Professionnelle

Jang SCHILTZ

Centre Universitaire du Luxembourg **Université Nancy II**
Département des Sciences Groupe d'Analyse Psychométrique des Conduites

Septembre 1999

Ce travail a été fait sous la direction de Paul Dickes, professeur à l'Université Nancy II, en collaboration avec Romain Martin, psychologue-chercheur de la cellule de recherche en évaluation à l'ISERP et Dominique Portante, directeur du SCRIPT.

TABLE DES MATIERES

Chapitre I

PRESENTATION DU PROJET DE RECHERCHE

Les épreuves standardisées actuelles développées pour le passage primaire post-primaire sont étalonnées d'après des modèles IRT (cf. P. Dickes et alt., 1994), mais n'épuisent pas toutes les possibilités méthodologiques de ces procédés. Il est possible d'en faire un instrument de recherche autrement puissant sous forme d'un test adaptatif informatisé. Le point de départ de notre projet est l'analyse et le traitement des données recueillies au cours des années précédentes par des procédés stochastiques dérivés du modèle de Rasch. En choisissant les algorithmes mathématiques adéquats et en les adaptant, nous nous proposons de construire un test adaptatif objectif, fiable, valide et en même temps flexible et adaptable à différents types de populations, pouvant servir à l'évaluation sommative et formative des acquisitions scolaires en mathématiques et contribuer efficacement à l'orientation des élèves.

1.1 Objectifs

Les épreuves standardisées constituent des instruments de diagnostic objectifs permettant une évaluation des savoirs et des savoirs-faire scolaires des élèves.

De telles épreuves ont été élaborées au Luxembourg pour le passage primaire post-primaire (cf. R. Martin, 1998). Elles sont basées sur des procédés modernes de la théorie des tests psychologiques, à savoir les modèles qui ont remplacé les procédés classiques basés sur

le score vrai, dans lesquels l'indice de difficulté et l'indice de discrimination des items dépendaient de la population d'étalonnage.

Le contenu de ces épreuves a été déterminé par les exigences curriculaires, alors que leur forme répondait aux exigences formelles de la méthodologie moderne. Il s'agissait donc d'un compromis entre les pédagogues et les théoriciens des tests, aboutissant à un instrument purement applicatif.

Il est cependant possible, en particulier grâce à l'application de modèles stochastiques spécifiques, d'en faire un instrument de recherche beaucoup plus puissant, exploitant mieux les possibilités de la théorie mathématique statistique.

Le but de notre projet de recherche est la construction d'un test adaptatif, à partir des données des épreuves standardisées, recueillies lors du passage primaire post-primaire au cours des années précédentes et conservées à l'Institut Supérieur d'Etudes et de Recherches Pédagogiques de Walferdange.

Un test adaptatif est un test sur mesure, adaptant la difficulté des items à la compétence du sujet. En général, au début, on présente un item de difficulté moyenne. Si le sujet réussit, on présente un item plus difficile. Lorsqu'il échoue, on présente un item plus facile. Ainsi, on tente toujours de proposer des items dont le niveau de difficulté est adapté au sujet testé. Dans un bon test adaptatif, le sujet réussit environ 50 pour-cent des items administrés, quel que soit son niveau de compétence. Les tests adaptatifs fournissent une discrimination bien meilleure que les tests collectifs classiques. En plus des avantages psychométriques, ils présentent encore des avantages psychologiques, puisqu'ils évitent au sujet la frustration ou l'ennui engendrés par des items trop difficiles ou trop faciles.

Ce test adaptatif sera administré par ordinateur (CAT : computerized adaptive test). La mémoire de l'ordinateur devra contenir une banque de données d'items obtenue par les procédés décrits ci-dessus ainsi que les algorithmes permettant d'estimer, après l'administration de chaque nouvel item, le niveau de compétence du sujet à partir des items déjà administrés et du vecteur des réponses correctes et fausses. Ensuite il choisira l'item présentant le pouvoir discriminatif le plus élevé au niveau de la compétence actuellement estimée du sujet.

1.2 Théories et procédés

La théorie générale sur laquelle reposera l'analyse des données et la construction du test adaptatif est la théorie de réponse à l'item. C'est une généralisation du modèle dichotomique logistique proposé par G. Rasch (cf. Rasch, 1960) et propagé par G. Fischer (cf. Fischer 1974). Ce modèle permet l'estimation des paramètres de difficulté des items sur une échelle relative, respectivement, après une logarithmisation des paramètres, sur une échelle absolue.

Une caractéristique très importante de ce modèle consiste dans son indépendance par rapport à la population de référence, tant en ce qui concerne la comparaison entre les items qu'entre les sujets. Chaque item est caractérisé par un seul paramètre, la "difficulté"; chaque sujet est caractérisé également par un seul paramètre, la "compétence".

Le domaine d'application du modèle logistique de Rasch peut être vérifié par voie empirique et généralisé graduellement : si le coefficient de difficulté de deux items d'un même test reste égal dans deux populations différentes, la validité du modèle peut être admise; lorsque ce coefficient change, l'item a une signification psychologique qualitativement différente dans les deux populations en question.

Lorsque la validité du modèle est établie pour la population en question, les items sont comparables, indépendamment de l'échantillon des individus qui ont répondu au test.

A partir de ce modèle, Rasch a développé sa théorie de la "spécificité objective". La maximalisation de la vraisemblance conditionnelle (conditional maximum likelihood) conduit à des équations de vraisemblance pour les paramètres des items, qui sont indépendants des paramètres des individus.

Un procédé très économique d'estimation des paramètres, mais dont le fondement mathématique théorique est complexe, a été proposé par G. Fischer : il s'agit du procédé d'estimation du "chi 2 minimal" (cf. Fischer, 1974).

E.B. Andersen a proposé en 1973 son "test du quotient de vraisemblance" pour contrôler la validité du modèle de Rasch pour les données recueillies dans un certain type de

population (cf. Andersen, 1973). Il existe d'autres procédés de contrôle du modèle. L'analyse mathématique des données nous guidera dans le choix du test le plus adéquat.

Le modèle de Rasch est généralisable en modèle plurifactoriel, appelé "modèle logistique linéaire" et en "modèle logistique polychotome" applicable à des réponses pluricatégorielles. Ces modèles permettent de classer les items par ordre de difficulté croissante, en relation avec les opérations logiques latentes impliquées, indépendamment des paramètres liés au matériel et aux personnes.

Le choix du modèle le plus approprié constitue une partie intégrante du projet de recherche.

1.3 Etapes de la recherche

Les données des épreuves standardisées des années précédentes sont analysées conformément aux procédés décrits ci-dessus. Il s'agit essentiellement des scores de compétence obtenus grâce à deux épreuves successives en mathématiques examinant les savoirs et savoirs-faire scolaires suivants :

Première épreuve :

1. opérations : ce score regroupe des exercices qui portent sur l'addition, la soustraction, la multiplication et la division;
2. fractions : ce score regroupe les exercices de calcul à l'aide de fractions;
3. système décimal et grandeurs : ces exercices touchent les transformations se rapportant au système décimal et au système métrique (longueurs, poids,…);
4. géométrie : ce score regroupe des exercices qui sont en rapport direct avec les caractéristiques de figures positionnées dans l'espace.

Deuxième épreuve :

1. fractions et système décimal : ces exercices combinent le calcul des fractions et les transformations d'échelles métriques;
2. problèmes : ces exercices demandent la résolution d'un problème factuel (Sachaufgabe) et nécessitent une approche par étapes;

3. géométrie : les exercices regroupés sous cette dimension sont en rapport direct avec les caractéristiques des corps et des figures.

La validité de la théorie de réponse à l'item est testée pour les données en question. La difficulté des items est évaluée au moyen du procédé logistique et les items non homogènes sont éliminés. Pour les items retenus, on essaie de représenter les indices de difficulté comme fonction simple des opérations latentes, définies par leur position dans l'axe des dimensions principales et dans la hiérarchie des niveaux, de sorte que la complexité psychologique d'une épreuve s'explique par sa structure logique.

Chaque item sera défini de manière univoque dans sa structure logique, par rapport à sa relation aux facteurs latents fondamentaux. Lorsque ces facteurs agissent de manière indépendante, les résultats sont facilement interprétables et un nombre pratiquement illimité d'items ayant une structure logique analogue peuvent être construits.

De nouvelles épreuves, de degré de complexité variable, peuvent être générées et leur degré de difficulté peut être prédit avec une exactitude suffisante, d'où la possibilité de créer à volonté des tests parallèles avec des items "déguisés".

Cette banque d'items permettra la construction d'un test adaptatif au moyen d'algorithmes mathématiques spécifiques à inclure dans le programme, définissant le point de départ, la sélection des items et la fin de l'épreuve pour un sujet en question.

1.4 Intérêt scientifique

Notre projet de recherche permettra l'élaboration d'un genre de test relativement nouveau et peu répandu, s'appliquant à la situation scolaire spécifiquement luxembourgeoise.

La méthodologie décrite ci-dessus peut aboutir à la construction d'épreuves objectives, fiables, flexibles, adaptables au niveau de compétence de différentes populations d'élèves, par exemple les élèves luxembourgeois et les élèves francophones défavorisés par la langue véhiculaire allemande, et néanmoins comparables entre elles, ce qui ouvrira des perspectives

intéressantes pour la recherche future. Comme la banque de données permettra de nouvelles combinaisons d'items en nombre quasiment illimité, la construction de tels tests adaptatifs est par ailleurs très économique.

1.5 Résultats escomptés

Les procédures d'orientation, instituées pour le passage primaire post-primaire, pourront tirer profit d'un instrument de mesure créé en accord avec des principes psychométriques ayant fait leurs preuves dans d'autres pays. La construction future d'épreuves standardisées analogues sera facilitée.

Des tests adaptatifs construits d'après les mêmes principes ne faciliteront pas seulement le passage primaire post-primaire mais pourront servir à l'évaluation continue des progrès des élèves et pourront être introduits également dans l'enseignement secondaire.

Il ne faut par ailleurs pas négliger l'effet important de telles épreuves sur la motivation des élèves : en effet ceux-ci sont beaucoup moins confrontés à l'expérience d'échec ou d'ennui, puisque les items sont sélectionnés par rapport à leur niveau de compétence actuel.

Il est dans l'intérêt de tous les membres de la communauté scolaire de disposer d'un instrument de mesure multifonctionnel pouvant fournir une contribution intéressante à l'orientation des élèves, puisqu'il permet une évaluation à la fois sommative et formative des processus d'apprentissage. Ainsi ce test pourra être mis à la disposition des enseignants luxembourgeois qui pourront l'utiliser dans leur pratique quotidienne. D'autre part, il pourra apporter une contribution importante aux cours d'appui.

Chapitre II
LA THEORIE DE REPONSE PAR ITEM

Le modèle de mesure utilisé pour évaluer les épreuves standardisées est celui de la théorie de réponse par item (IRT: *item response theory*) (cf. Fischer, 1974 ainsi que Weiss & Yoes, 1991 pour une présentation détaillée de ce modèle). Comme pour tout modèle mathématique, il existe certains postulats qui doivent être respectés.

2.1 Les postulats de l'IRT

L'IRT repose sur les postulats suivants:

- On suppose que la probabilité d'une réponse correcte est attribuable à la position du sujet sur un nombre spécifique *k* de traits latents de compétence requise pour répondre au type d'items en question. Ces traits latents peuvent être conceptualisés comme un espace de dimension *k*. Dans notre application, nous présumons que cet espace est unidimensionnel, ce qui revient à dire que nous nous intéressons à mesurer une seule variable (à savoir la compétence en mathématiques). On suppose que la probabilité d'une réponse correcte évolue comme une fonction logistique dont le point d'inflexion est projeté sur le trait latent de compétence, permettant ainsi de déterminer la difficulté de l'item.

- On suppose qu'il y a indépendance locale, c'est-à-dire que la probabilité d'une réponse correcte d'un sujet à un item donné ne dépend pas de ses réponses aux autres items du test. Ainsi un item administré au début du test ne doit pas influencer un item administré plus loin et les corrélations entre items doivent être

dues uniquement à l'influence du trait latent, ce qui implique que les items sont non corrélés pour des sujets à niveau égal de compétence.

2.2 La fonction de réponse de l'item

Le concept central de la théorie de réponse par item est celui des courbes caractéristiques des items (ICC: *item caracteristic curve*), également appelées fonctions de réponse des items (IRF: *item response function*), qui décrivent la probabilité de réussite à un item donné en fonction de la compétence des sujets. Pour des modèles dichotomiques, on utilise le plus souvent la fonction logistique.

Par convention, la réponse d'un sujet donné à un item dichotomique *j* est codée par

$$x_j = \begin{cases} 1, & \text{si la réponse est correcte} \\ 0, & \text{si la réponse est fausse} \end{cases}$$

Si l'on dénote la compétence du sujet par θ, la probabilité d'une réponse correcte à l'item *j* pour une compétence donnée θ est notée $P_j(\theta)$. La forme générale de la fonction de réponse de l'item est

$$P_j(\theta) = \frac{1}{1+e^{-a_j(\theta-b_j)}},$$

où a_j désigne la discrimination de l'item (*item slope*) et b_j sa difficulté (*item threshold*).

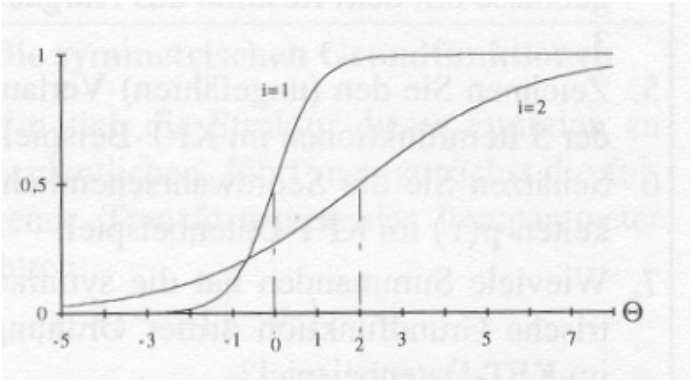


Figure 1: 2 IRF avec les paramètres a_1=0, b_1=2, respectivement a_2=2, b_2=0,5 (dans Rost, 1996).

L'opposé de l'exposant

$$z_j = a_j(\theta - b_j),$$

est appelé logit. Le logit peut aussi être écrit comme

$$z_j = a_j\theta + c_j,$$

où $c_j = -a_j b_j$ est appelé l'intercept (*item intercept*).

Si les discriminations sont nulles pour tous les items, on dit qu'on est en présence d'un modèle à un paramètre ou modèle de Rasch. Habituellement, les modèles logistiques sont représentés par la fonction $\Psi_j(\theta) = 1/1 + e^{-z_j}$.

Dans le cas d'items à choix multiple, un sujet qui ne connaît pas la bonne réponse peut néanmoins répondre correctement en choisissant au hasard parmi les alternatives proposées. La probabilité qu'un sujet avec une compétence égale à θ ne connaît pas la bonne réponse, mais répond correctement est $g_j\left[1-\Psi_j(\theta)\right]$, si g_j désigne la probabilité de deviner juste (pour des items à choix multiple, si le sujet choisit au hasard, g_j=1/A, où A désigne le nombre de réponses possibles).

La probabilité d'une réponse correcte à l'item numéro *j* pour un sujet avec une compétence égale à θ, est donc donnée par

$$\begin{aligned} P_j(\theta) &= g_j\left[1-\Psi_j(\theta)\right] + \Psi_j(\theta) \\ &= g_j + (1-g_j)\Psi_j(\theta). \end{aligned}$$

L'interprétation des paramètres qui caractérisent un item donné dans le modèle logistique est donnée par les figures 2 et 3.

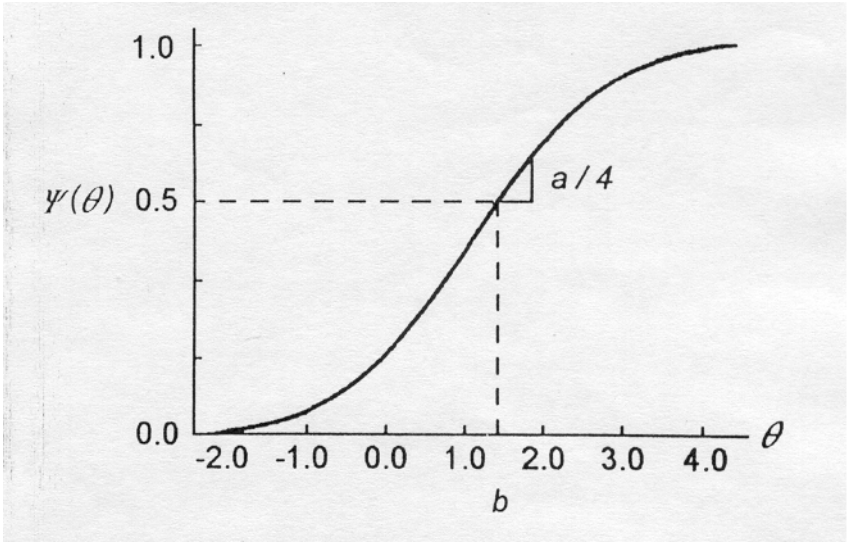


Figure 2: Interprétation des paramètres du modèle IRT à 2 paramètres (dans Zimokski et autres, 1996)

- La difficulté *b* de l'item détermine la position de la courbe par rapport à l'axe des compétences. Elle est égale à l'abscisse du point d'inflexion de la fonction de

réponse de l'item. C'est également le point où la pente de l'IRF est maximale. Donc, plus un item est difficile, plus sa courbe est décalée vers la droite.

- La discrimination *a* est proportionnelle à la pente de l'IRF. Plus cette pente est raide, plus l'item discrimine entre deux sujets qui ont des niveaux de compétence assez proches de la difficulté de l'item. Lorsque la pente est assez faible, l'item montre une discrimination faible, mais cela sur une bande assez large de compétences.
- Le paramètre de „guessing" *c*, également appelé niveau du score de pseudo-chance, indique la probabilité de donner la réponse exacte par hasard. Il est égal à l'ordonnée de l'asymptote inférieure de la fonction de réponse à l'item et correspond à la probabilité qu'un sujet de compétence très faible (tendant vers moins l'infini) donne la bonne réponse. Puisque dans les tests qu'on analyse ici, il n'y a presque pas d'items à choix multiple, on utilise un modèle à deux paramètres et on pose par conséquent c=0.

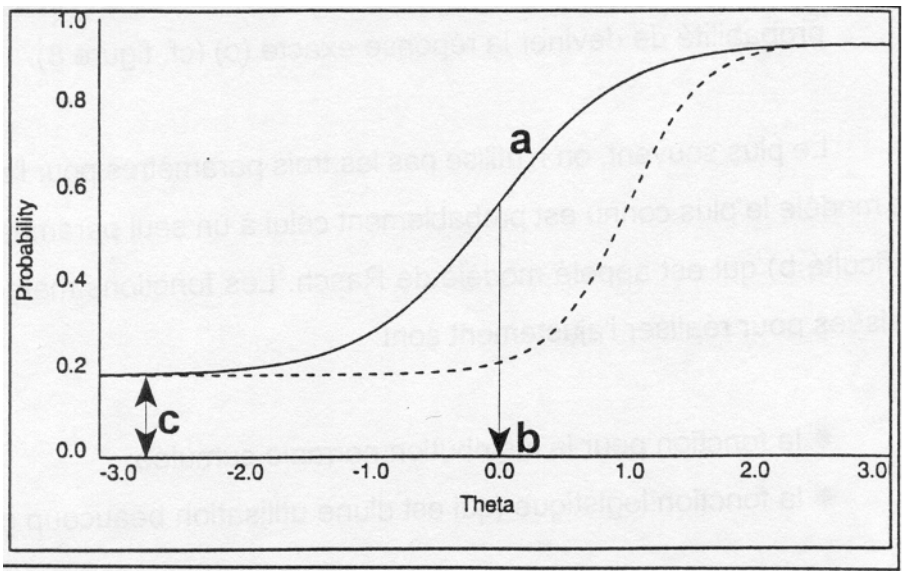


Figure 3: Interprétation des paramètres du modèle IRT à 2 paramètres (dans Assessment Systems Corporation, 1995).

Un des grands avantages de l'IRT est le fait que l'estimation des paramètres ne dépend pas de la population à partir de laquelle cette estimation est faite. Autrement dit, la distribution des compétences dans l'échantillon utilisé pour calibrer les items n'a aucun effet sur les estimations des paramètres des items. Ces paramètres sont invariants lorsqu'ils sont estimés dans des groupes ayant des niveaux de compétences différents.

Ceci veut dire qu'une échelle de mesure semblable peut être utilisée dans des populations différentes et que les sujets peuvent être testés avec des items différents, appropriés à leur niveau de compétence tout en préservant la comparabilité de leurs scores (Anastasi, 1982).

On obtient donc approximativement les mêmes fonctions de réponse des items, quelle que soit la distribution de la compétence dans l'échantillon utilisé pour calculer les paramètres de l'item. Par conséquent, une IRF donne la probabilité qu'un individu répondra correctement à un item à un niveau donné de compétence, sans que cette probabilité ne dépende du nombre de sujets situés à ce niveau de compétence. Cette propriété d'invariance des IRF est une caractéristique importante des modèles de théorie de réponse par item.

2.3 La fonction de réponse du test

Le postulat de l'indépendance locale implique que les IRF sont additifs, c'est-à-dire, qu'ils peuvent être additionnés à chaque niveau de compétence. En sommant pour tout item la probabilité d'une réponse correcte à chaque niveau de compétence et en divisant cette somme par le nombre d'items, on obtient une moyenne des IRF. La courbe ainsi obtenue est appelée fonction caractéristique du test ou encore fonction de réponse du test (TRF: *test response function*).

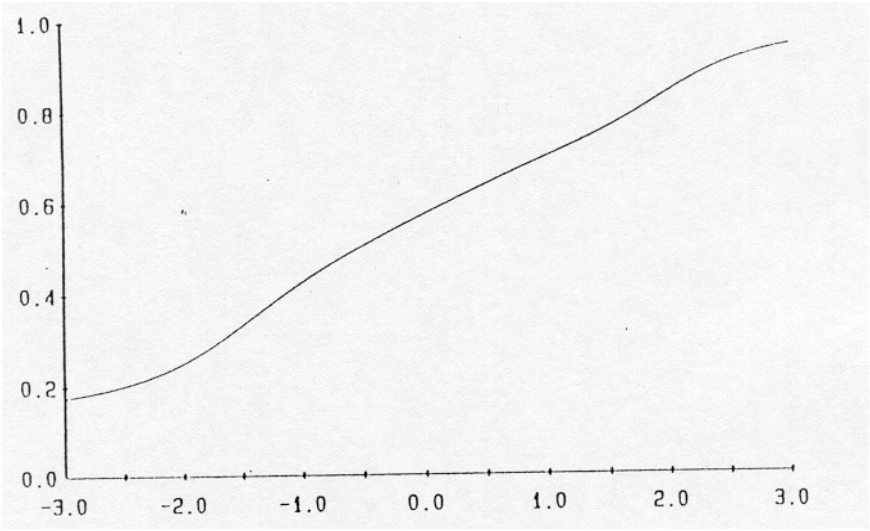


Figure 4: La TRF pour un test composé de six items d'après un modèle de Rasch à trois paramètres (dans Weiss & Yoes, 1991).

La TRF établit un lien entre le type de score utilisé dans la théorie classique des tests, c'est-à-dire le nombre de réponses correctes et le type de score utilisé en IRT, c'est-à-dire le niveau estimé de compétence pour un sujet donné.

2.4 La fonction d'information de l'item

Le concept d'information est aussi important que le concept de fonction de réponse de l'item. L'information renvoie à la précision de ce que l'on mesure; elle est inversement proportionnelle à l'erreur de mesure, donc plus l'erreur de mesure est petite, plus l'information est grande.

En IRT, il est possible de déterminer combien d'information chaque item fournit à chaque point du continuum de compétence. Plus on aura d'information à un certain niveau de compétence, plus précise sera l'évaluation des paramètres à ce niveau. Un des avantages de l'IRT est donc de pouvoir choisir les items fournissant un maximum d'information à des niveaux spécifiques de compétence. C'est en fait la pente de l'IRF qui indique le pouvoir de discrimination. Plus cette pente est raide, plus la probabilité de répondre correctement à un item varie en fonction de la compétence, à condition de se trouver dans un voisinage du point de pente maximale de l'item. En effet, l'information d'un item à un niveau de compétence θ est donnée par la formule suivante :

$$I(\theta) = a_j^2 P_j(\theta) \left[1 - \mathrm{P_j}(\theta)\right].$$

La fonction d'information de l'item ainsi obtenue (IIF: *item information function*) atteint son maximum au point où la pente de l'IRF est la plus élevée.

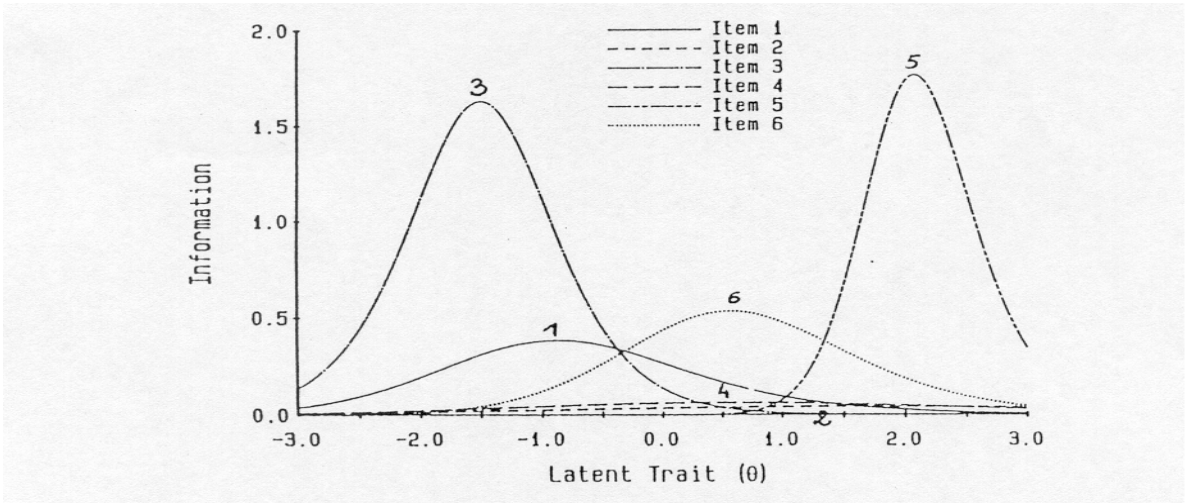


Figure 5: Les IIF d'un test comportant 6 items (dans Weiss & Yoes, 1991).

Il n'est d'ailleurs pas étonnant que la position du maximum de l'IIF dépende de la difficulté de l'item. En effet, si un item différencie bien des individus qui ont un niveau faible de compétence, cet item sera plus facile qu'un item qui différencie bien des individus de compétence élevée. L'indice de discrimination, quant à lui, a une influence sur la forme de

l'IIF qui implique un certain dilemme de largeur de bande de la fidélité: les items avec une fonction d'information à pic discriminent bien, mais sur une portion réduite du continuum latent. Si la fonction s'avère plus plate, on est en présence d'une faible discrimination, mais avec une largeur de bande élevée. Cet effet est connu sous le nom de paradoxe de la largeur de bande de la fidélité. Il implique que, lors de la construction d'un test il faut faire un compromis entre l'information totale que procure un item et l'étendue sur laquelle cette information est disponible.

2.5 La fonction d'information du test

Comme pour les fonctions de réponse de l'item, le postulat d'indépendance locale permet la sommation de l'information des différents items pour un niveau de compétence donné. On obtient ainsi la fonction d'information du test (TIF: *test information function*) qui indique combien d'information est fournie par le test à chaque niveau de compétence et qui représente donc un outil pour déterminer la qualité d'un test.

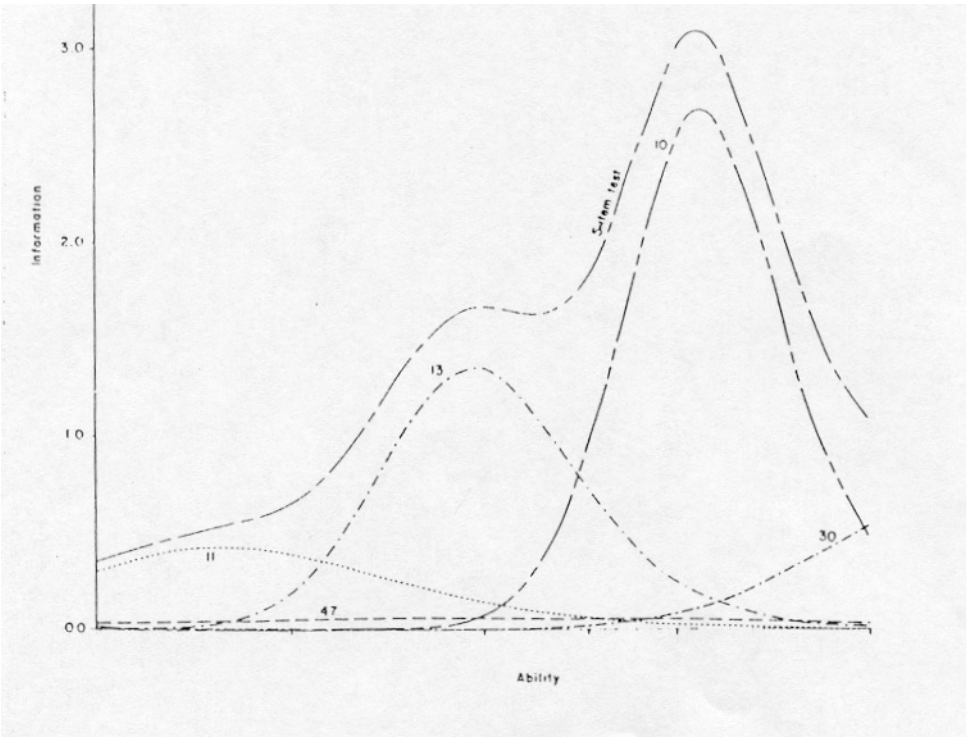


Figure 6: Les IIF d'un test composé de 5 items, ainsi que la TIF de ce test (dans Lord, 1980).

Alors que dans la théorie classique des tests, la contribution de chaque item à la fidélité ou à la validité du test dépend de tous les autres items qui composent le test, dans la théorie des

IRT, la contribution d'un item à l'efficacité d'un test ne dépend donc que de ses propriétés intrinsèques.

La TIF donne un moyen de connaître les niveaux du trait latent pour lesquels le test mesure le plus précisément et ceux pour lesquels il faut encore ajouter d'autres items. L'une des utilisations les plus importantes du concept d'information dans l'IRT réside dans le fait que les IIF et la TIF peuvent être développés avant que le test ne soit construit.

2.6 L'estimation des niveaux de compétence

Il y a deux méthodes courantes pour estimer les niveaux de compétence des sujets. Ces méthodes sont d'une part l'estimation du maximum de vraisemblance (*maximum likelihood estimation*) et d'autre part l'estimation modale bayesienne (*Bayesian modal estimation*).

Pour ces deux méthodes, on tient compte de toute l'information contenue dans la suite des réponses d'un sujet aux items du test. Le vecteur réponse pour chaque individu consiste en une suite de 1 et de 0, qui indiquent s'il a répondu correctement ou non aux différents items. L'indépendance locale implique que la probabilité pour un vecteur donné est obtenue par multiplication. Elle varie en fonction du niveau de compétence θ. Cette fonction est appelée la fonction de vraisemblance (LF: *likelihood function*).

Dans le cas d'un seul item, la fonction de vraisemblance correspond simplement à $P(\theta)$ si le sujet à répondu correctement à l'item, respectivement à 1- $P(\theta)$ dans le cas contraire. Avec deux items, la fonction de vraisemblance correspond au produit des deux probabilités qui reflètent les réponses du sujet. Dans le cas général, on multiplie tout simplement les probabilités correspondant aux réponses données aux différents items. Une fois que l'on a obtenu la fonction de vraisemblance pour une certaine séquence de réponses, il reste à déterminer quelle valeur de θ est la plus probable pour la séquence donnée.

2.6.1 L'estimation du maximum de vraisemblance

L'estimation du maximum de vraisemblance des réponses du sujet *i* est la valeur θ qui maximise la fonction L_i déterminée par

$$\log L_i(\theta) = \sum_j \left\{ x_{ij} \log P_j(\theta) + (1 - x_{ij}) \log\left[1 - P_j(\theta)\right] \right\},$$

où $P_j(\theta)$ désigne la réponse du sujet à l'item *j*.

Il faut alors résoudre l'équation de vraisemblance suivante

$$\frac{\partial \log L_i(\theta)}{\partial \theta} = \sum_j \frac{x_{ij} - P_j(\theta)}{P_j(\theta)\left[1 - P_j(\theta)\right]} \frac{\partial P_j(\theta)}{\partial \theta} = 0,$$

ce qui, en termes géométriques, revient à déterminer la projection sur l'axe des abscisses du maximum de la fonction de vraisemblance.

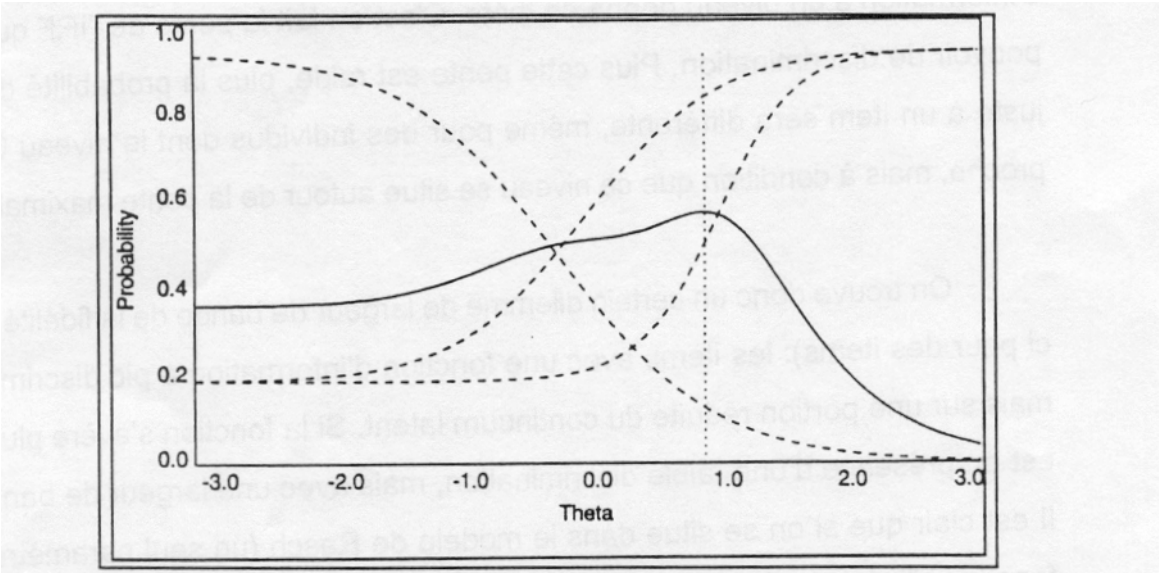


Figure 7: Représentation de trois IRF (en pointillé) avec deux items résolus correctement et un item à réponse fausse, ainsi qu'une courbe de vraisemblance du sujet (trait continu) pour lequel la compétence est déterminé par projection (dans Assessement Systems Corporation, 1995)

Un problème avec l'estimation du maximum de vraisemblance est que, si un sujet répond correctement à tous les items (vecteur parfait) ou incorrectement à tous les items (vecteur zéro), alors la fonction de vraisemblance n'admet pas un seul maximum identifiable d'une manière univoque. Ce problème est partiellement résolu par l'estimation modale bayesienne.

2.6.2 L'estimation modale bayesienne

L'idée de base de la méthode d'estimation modale bayesienne est que, si la distribution de θ est connue ou peut être estimée pour la population, alors cette information peut être utilisée pour faire des estimations plus précises de θ. On fait d'habitude l'hypothèse d'une distribution normale a priori sur le trait latent. La fonction de vraisemblance est alors pondérée (c'est-à-dire multipliée) par cette fonction de distribution a priori. On est donc en présence d'une nouvelle distribution appelée distribution postérieure. L'estimation de θ correspond à la valeur la plus élevée de cette fonction postérieure de vraisemblance, qui est donnée par

$$L(\theta|x_i) = \sum_j \left\{ x_{ij} \log P_j(\theta) + (1 - x_{ij}) \log\left[1 - P_j(\theta)\right] \right\} + \log g(\theta),$$

où $g(\theta)$ désigne la densité de la distribution a priori de θ. Dans ce cas, l'équation de vraisemblance à résoudre est

$$\sum_j \frac{x_{ij} - P_j(\theta)}{P_j(\theta)\left[1 - P_j(\theta)\right]} \frac{\partial P_j(\theta)}{\partial \theta} + \frac{\partial \log g(\theta)}{\partial \theta} = 0.$$

Le problème avec cette méthode d'estimation est que, si la distribution a priori ne correspond pas à la réalité, les estimations sont biaisées. L'avantage est qu'on peut faire des estimations pour des vecteurs parfaits ou zéros.

L'erreur standard de mesure commise en estimant θ par une des deux méthodes ci-dessus est donnée par

$$\text{S.E.}(\theta) = \sqrt{\frac{1}{I(\theta)}},$$

où I(θ) désigne la fonction d'information du test.

2.7 L'estimation des paramètres de l'IRF

Comme toutes les applications de l'IRT dépendent des paramètres des items, il est très important que ces paramètres soient estimés avec le plus de précision possible. Ce processus d'estimation de l'item est généralement appelé le calibrage. Il peut être effectué par des logiciels spécialisés (LOGIST, BILOG, ASCAL et BICAL par exemple). La sélection d'un logiciel pour l'estimation des paramètres des items dépend de la forme de l'IRF sélectionnée (modèle à 1, 2 ou 3 paramètres) et de la procédure d'estimation désirée (maximum de vraisemblance ou bayesienne).

Une caractéristique commune de tous les logiciels est qu'ils doivent tous définir arbitrairement le point zéro du continuum θ / b.

Dans les modèles à deux ou trois paramètres, le point zéro est défini comme la moyenne des compétences de l'échantillon à partir duquel les paramètres de l'item sont estimés. Dans le modèle à un paramètre, le point zéro est parfois défini comme la moyenne des difficultés des items. Ceci implique en fait une violation de la propriété d'invariance des paramètres.

2.8 Application de l'IRT à la création de tests

Le cadre théorique ainsi défini constitue un instrument très puissant pour la construction et le développement de tests. Les items dont on a estimé les paramètres constituent la banque d'items à partir de laquelle le test sera créé. Contrairement à la théorie

classique des tests, la théorie de réponse par items permet de déterminer les propriétés statistiques du test et d'estimer sa performance avant qu'il soit passé pour la première fois. On peut en effet calculer la fonction d'information du test et obtenir ainsi la précision avec laquelle le test mesure tout au long du continuum latent. Ceci permet de connaître l'erreur standard de mesure du test et de combler des lacunes éventuelles en introduisant d'une manière très précise des items à pouvoir discriminatif élevé, à l'endroit où l'on a constaté une erreur de mesure trop importante.

Dans cette conception méthodologie d'un test, les items ajoutés sont donc ceux qui contribuent à donner plus d'information dans le voisinage du continuum latent où cette information est nécessaire. Les items qui ne contribuent pas aux caractéristiques désirées peuvent être enlevés du test. La fonction d'information du test peut être examinée après addition ou élimination d'items particuliers, pour mesurer l'effet qui résulte de cette opération.

Ainsi, on peut construire des tests qui procurent un maximum d'information à un endroit précis du continuum latent, ceci par exemple lorsqu'il s'agit de distinguer entre des sujets qui réussissent ou qui échouent lors d'un examen. Un tel test est appelé test à pic (*peaked test*). Pour créer un tel test, on sélectionne des items dont les indices de difficulté sont proches du score limite. Leur indice de discrimination doit être aussi grand que possible, puisque l'information est proportionnelle à l'indice de discrimination, et dans le cas d'un modèle à trois paramètres, le paramètre de „guessing" est choisi aussi petit que possible; en effet, des grandes valeurs de celui-ci tendent à diminuer l'information.

La théorie de réponse par item rend également possible de concaténer deux tests différents, ce qu'on appelle mise en équation. La mise en équation est nécessaire lorsque l'on dispose de deux formes qui n'ont pas été explicitement faites pour être parallèles. Il faut alors pouvoir placer les scores correspondants sur la même échelle pour pouvoir comparer les sujets qui ont passé des tests différents.

Il y a substantiellement deux procédés de mise en équation : La mise en équation horizontale permet de mettre sur la même échelle des scores issus de tests ayant approximativement le même niveau de difficulté, par exemple des tests conçus pour le même niveau scolaire. La mise en équation verticale est utilisée pour les scores provenant de tests ayant des niveaux de difficulté différentes, par exemple de tests scolaires passés par des groupes d'âge différents.

Le problème rencontré provient du fait que les vrais paramètres des items sont inconnus et que, dans le processus d'estimation des paramètres, l'origine (le point zéro) de l'échelle des difficultés des items est arbitrairement fixé (ou bien, comme moyenne des estimations de la capacité des sujets qui servent à calibrer les items, ou bien, comme moyenne des difficultés de l'échantillon d'items utilisé). Ceci implique que les échelles obtenues au moyen d'estimations séparées, dans deux populations différentes, sont invariantes seulement modulo une transformation affine près. Pour faire le lien entre deux échelles, il faut donc être en mesure de déterminer cette transformation affine, ce qui nécessite que les deux procédés d'estimation aient quelque chose en commun, soit un certain nombre d'items, soit un certain nombre de sujets ayant participé à l'élaboration des deux échelles, soit un certain nombre de sujets ayant passé un troisième test mesurant le même trait latent et servant donc de référence.

Chapitre III

L'ANALYSE DES ITEMS

3.1 Provenance des items

> Les épreuves standardisées qui ont apparu dans la nouvelle procédure de passage primaire post-primaire sont organisées pour l'allemand, le français et les mathématiques. Ces épreuves standardisées ont été élaborées par trois groupes (un pour chaque matière scolaire) mixtes d'instituteurs, de professeurs (en provenance de l'enseignement secondaire technique et classique) et d'inspecteurs. Ces groupes ont été conseillés par des psychologues-chercheurs de la cellule de recherche en évaluation de l'ISERP. Les épreuves ont été administrées et corrigées par les institutrices et instituteurs concernés selon des consignes communes. Les réponses des élèves ont alors été envoyées à la cellule de recherche en évaluation de l'ISERP, qui a réalisé une analyse statistique comportant d'abord une vérification et une amélioration de la qualité psychométrique des instruments élaborés et ensuite un étalonnage des différents scores retenus (cf. Martin, 1998). Le traitement s'est fait sur des données anonymisées, les élèves pouvant être identifiés à l'aide d'un code dont seul l'instituteur possède la clef.

Ce sont ces données anonymisées des deux épreuves en mathématiques de l'année scolaire 1996/97 qu'on a mis à notre disposition, pour constituer la banque d'items à partir de laquelle on construira le logiciel. Malheureusement, pour des raisons psychométriques, on a seulement pu utiliser l'une des deux épreuves. En effet, même si elles ont été effectuées par la même population de sujets, le fait qu'elles se sont déroulées à un intervalle d'environ quatre mois et qu'il n'y a aucun item en commun empêche la possibilité d'arriver à des scores

comparables. Nous avons décidé de travailler avec l'épreuve de novembre 1996, puisque celle-là est constituée d'un nombre plus considérable d'items.

L'épreuve standardisée en mathématiques du mois de novembre 1996 est subdivisée en 24 questions fournissant un total de 81 items qui font l'objet de l'annexe.

Nous avons exclu tout de suite les six items de la question 16, parce que cette question peut être résolue de 6 façons différentes. Ainsi, par exemple, la solution

	Seite a (in cm)		Seite b (in cm)		Umfang (in cm)
Rechteck 1	a)	16	1	d)	34
Rechteck 2	b)	8	2	e)	20
Rechteck 3	c)	4	4	f)	16

est aussi juste que la solution

	Seite a (in cm)		Seite b (in cm)		Umfang (in cm)
Rechteck 1	a)	4	4	d)	16
Rechteck 2	b)	16	1	e)	32
Rechteck 3	c)	8	2	f)	20

Pourtant, un élève qui aurait trouvé les deux rectangles 4x4 et 8x2, mais pas le 16x1 aurait été codé

16a	16b	16c	16d	16e	16f
0	1	1	0	1	1

dans le premier cas et

16a	16b	16c	16d	16e	16f
1	0	1	1	0	1

dans le deuxième, ce qui rend incalibrable les items de la question 16.

3.2 Détermination du modèle

L'analyse des données proprement dite a comporté trois étapes. Dans une première étape, nous avons vérifié que les données peuvent être bien représentées par un modèle IRT. Ensuite, nous avons choisi parmi les différents modèles celui qui est le mieux adapté à la situation donnée, et finalement, nous avons déterminé de quels items fut constitué la banque d'items à partir de laquelle le logiciel de test adaptatif a été construit.

Les épreuves mathématiques contiennent des symboles univoques et toute notre population a appris les mathématiques par la même méthode. Ceci rend plausible la validité des modèles IRT pour les données en question.

Nous avons par contre dû écarter dès le début la question 9e, car elle ne vérifie pas le postulat de l'indépendance stochastique, qui est à la base de tous les modèles IRT. Il est en effet évident que la réponse à la question 9e dépend des réponses aux questions 9a à 9d. Un sujet qui n'a pas réussi à trouver un des termes de la somme ne devrait pas trouver le résultat correct pour la somme globale. Ce qui nous inquiète un peu, c'est qu'en regardant les résultats des corrigés, on s'aperçoit que cette impossibilité est pourtant arrivée assez fréquemment. Cela veut ou bien dire que certains élèves ont triché, ou bien que les instituteurs n'ont pas tous corrigé de la même façon (il semble qu'il y en a qui ont jugé la réponse correcte, même si l'élève n'est pas arrivé au résultat attendu, si celui ci a correctement additionné les sous-résultats faux).

Finalement, nous avons donc retenu 74 items pour la suite. Nous avons d'abord testé si nos données pourraient vérifier un modèle IRT à un, respectivement deux paramètres (le modèle à trois paramètres ne semble pas intéressant, puisqu'il n'y a que les questions 3, 4, 12 et 17 que les élèves pourraient trouver au hasard). Pour effectuer cette analyse, nous avons utilisé le logiciel BILOG-MG. Nous avons effectué le calibrage des items en utilisant la méthode du maximum de vraisemblance et l'estimation modale bayesienne. Dans les deux cas, nous avons trouvé exactement le même résultat.

Le calibrage pour un modèle de Rasch (modèle IRT à un paramètre) a donné le résultat suivant :

Item	Intercept Erreur	Difficulté **Erreur**	Chi 2 **Indice d'adéquation**	Degrés de **Liberté**
1a	1.586 0.081	-1.697 0.087	13.6 0.1352	9
1b	3.044 0.131	-3.258 0.140	1.3 0.9719	6
1c	1.873 0.089	-2.005 0.095	8.1 0.4262	8
1d	2.112	-2.260	9.5	8

	0.094	0.101	0.2998	
1e	-1.255 0.075	1.343 0.080	29.7 0.0005	9
1f	1.395 0.077	-1.493 0.083	5.8 0.7573	9
1g	2.551 0.109	-2.730 0.117	4.3 0.7504	7
1h	-0.528 0.069	0.565 0.074	7.8 0.5559	9
1i	2.379 0.105	-2.546 0.112	8.2 0.3153	7
1j	0.530 0.070	-0.568 0.074	10.7 0.2987	9
2a	1.041 0.074	-1.114 0.080	9.3 0.4101	9
2b	1.236 0.073	-1.323 0.078	39.6 .0.0000	9
2c	1.224 0.078	-1.310 0.084	14.4 0.1083	9
3a	0.706 0.070	-0.756 0.075	8.5 0.4816	9
3b	0.071 0.068	-0.076 0.073	18.4 0.0308	9
4a	0.645 0.072	-0.690 0.077	12.3 0.1945	9
4b	1.750 0.089	-1.873 0.095	10.1 0.2557	8
4c	1.063 0.075	-1.138 0.080	6.8 0.5568	8
4d	0.671 0.070	-0.718 0.075	14.8 0.0948	9
4e	0.958 0.073	-1.026 0.078	8.1 0.5218	9
4f	0.311 0.065	-0.333 0.069	49.7 0.0000	9
4g	0.996 0.079	-1.066 0.085	50.5 0.0000	8
4h	0.095 0.067	-0.101 0.072	21.7 0.0100	9
5a	-0.483 0.079	0.517 0.084	81.8 0.0000	9
5b	-1.237 0.087	1.324 0.093	59.7 0.0000	9
5c	-1.428 0.088	1.529 0.094	48.6 0.0000	9
5d	-1.182 0.086	1.265 0.092	68.3 0.0000	9
6a	1.183 0.076	-1.266 0.082	8.1 0.5281	9
6b	-0.059	0.063	33.9	9

	0.073	0.078	0.0001	
6c	0.383 0.070	-0.410 0.075	7.8 0.5603	9
6d	1.382 0.083	-1.479 0.089	24.3 0.0039	9
6e	-0.315 0.075	0.337 0.080	42.2 0.0000	9
6f	1.013 0.077	-1.084 0.082	16.7 0.0332	8
7a	0.456 0.072	-0.488 0.077	17.6 0.0399	9
7b	0.958 0.074	-1.025 0.079	8.9 0.4471	9
7c	1.439 0.083	-1.540 0.089	26.2 0.0005	7
8	0.570 0.073	-0.610 0.078	17.8 0.0377	9
9a	1.858 0.089	-1.988 0.095	18.2 0.0200	8
9b	0.243 0.076	-0.260 0.081	71.4 0.0000	9
9c	0.600 0.076	-0.642 0.081	51.2 0.0000	9
9d	0.051 0.071	-0.055 0.076	30.2 0.0005	9
10a	1.980 0.095	-2.119 0.102	30.7 0.0000	6
10b	0.753 0.066	-0.805 0.071	65.2 0.0000	9
11a	1.259 0.080	-1.348 0.086	13.4 0.1450	9
11b	0.969 0.078	-1.037 0.084	40.7 0.0000	8
12a	-0.020 0.061	0.021 0.065	137.5 0.0000	9
12b	-0.039 0.067	0.042 0.071	18.7 0.0278	9
12c	-0.236 0.064	0.253 0.068	64.4 0.0000	9
12d	-0.295 0.066	0.316 0.071	30.5 0.0004	9
12e	2.196 0.097	-2.350 0.104	8.4 0.3003	7
12f	1.858 0.087	-1.988 0.093	7.6 0.4704	8
13	0.446 0.072	-0.478 0.077	12.1 0.2053	9
14a	0.491 0.070	-0.525 0.074	9.7 0.3725	9
14b	0.210	-0.224	18.5	9

	0.066	0.071	0.0294	
15	0.655 0.074	-0.701 0.079	26.3 0.0019	9
17	-0.271 0.068	0.290 0.073	16.5 0.0571	9
18a	-1.673 0.092	1.791 0.099	38.6 0.0000	9
18b	-1.734 0.093	1.855 0.100	32.4 0.0002	9
19a	1.308 0.077	-1.400 0.082	11.3 0.2556	9
19b	-0.841 0.075	0.900 0.080	16.7 0.0542	9
20a	-1.018 0.076	1.090 0.082	11.9 0.2170	9
20b	-0.165 0.074	0.176 0.079	34.5 0.0001	9
20c	-1.607 0.086	1.720 0.093	6.0 0.7436	9
20d	-1.551 0.087	1.660 0.093	10.7 0.2934	9
21a	1.980 0.093	-2.119 0.099	11.1 0.1931	8
21b	1.296 0.081	-1.387 0.086	23.7 0.0050	9
22a	3.459 0.152	-3.702 0.163	20.1 0.0028	6
22b	1.538 0.079	-1.646 0.084	14.6 0.1004	9
22c	-1.111 0.078	1.189 0.083	15.9 0.0689	9
23a	-0.184 0.072	0.197 0.078	20.1 0.0171	9
23b	-0.078 0.071	0.083 0.076	13.7 0.1312	9
23c	0.325 0.071	-0.348 0.079	39.5 0.0000	9
24a	-1.733 0.088	1.855 0.094	7.6 0.5806	9
24b	-2.249 0.103	2.407 0.110	17.5 0.0253	8

Tableau 1: **Paramètres et erreurs standards de mesure pour le modèle de Rasch**

L'indice d'adéquation des items (*item fit index*) indique ici si l'item en question convient au modèle de Rasch. En fait, la probabilité pour qu'un item soit adéquat au modèle se comporte comme un chi 2 avec un certain degré de liberté, que le logiciel calcule à partir de la vraisemblance de la distribution des réponses. On peut supposer que les items qui ont

une probabilité supérieure à 5 pour-cent peuvent être retenus. On pourrait donc maintenir les 36 items ayant un indice d'adaptation supérieur à 0,05.

De plus, le logiciel BILOG-MG nous a permis de déterminer que le logarithme de la fonction de vraisemblance pour ce modèle est de –38952,7747, la discrimination commune pour tous les items étant de 0,934 avec une erreur standard de mesure de 0,01.

En faisant la même analyse pour le modèle IRT à 2 paramètres (également appelé modèle de Birnbaum), on trouve un logarithme de la fonction de vraisemblance de -8341,3670 et on obtient le tableau suivant:

Item	Intercept Erreur	Discrimination Erreur	Difficulté Erreur	Dispersion Erreur	Chi 2 Indice	Degrés de Liberté
1a	1.473 0.085	0.668 0.089	-2.206 0.275	1.497 0.199	14.3 0.0744	8
1b	2.901 0.148	0.705 0.136	-4.112 0.704	1.418 0.273	5.8 0.4501	6
1c	1.790 0.095	0.765 0.096	-2.339 0.263	1.307 0.163	2.2 0.9733	8
1d	1.977 0.100	0.671 0.097	-2.947 0.393	1.491 0.216	2.1 0.9783	8
1e	-1.146 0.078	0.578 0.073	1.983 0.258	1.729 0.219	11.7 0.2278	9
1f	1.287 0.080	0.650 0.087	-1.978 0.251	1.538 0.206	9.4 0.4023	9
1g	2.417 0.120	0.702 0.113	-3.444 0.499	1.425 0.230	6.4 0.6037	8
1h	-0.500 0.070	0.672 0.071	0.744 0.122	1.489 0.158	8.1 0.5203	9
1i	2.334 0.117	0.853 0.115	-2.734 0.321	1.172 0.158	7.0 0.4341	9
1j	0.500 0.070	0.798 0.082	-0.626 0.100	1.252 0.129	13.3 0.1469	9
2a	0.994 0.076	0.799 0.085	-1.244 0.140	1.251 0.129	5.2 0.8169	9
2b	1.078 0.073	0.411 0.068	-2.621 0.443	2.432 0.401	7.2 0.6222	9
2c	1.221 0.082	0.939 0.093	-1.301 0.127	1.065 0.106	6.9 0.6514	9
3a	0.657 0.071	0.736 0.078	-0.893 0.120	1.358 0.143	14.5 0.1062	9
3b	0.056 0.068	0.750 0.079	-0.075 0.090	1.333 0.140	11.3 0.2572	9
4a	0.639 0.074	0.937 0.088	-0.682 0.086	1.067 0.101	10.4 0.3217	9
4b	1.860	1.142	-1.629	0.876	4.2	8

	0.105	0.115	0.134	0.088	0.8362	
4c	1.020 0.078	0.814 0.089	-1.254 0.135	1.229 0.134	9.6 0.3864	9
4d	0.614 0.070	0.684 0.079	-0.898 0.130	1.462 0.168	18.1 0.0335	9
4e	0.894 0.074	0.724 0.082	-1.235 0.149	1.380 0.157	4.5 0.8763	9
4f	0.259 0.065	0.433 0.065	-0.597 0.169	2.310 0.344	16.6 0.0554	9
4g	1.190 0.090	1.487 0.115	-0.800 0.064	0.672 0.052	4.8 0.7834	8
4h	0.076 0.067	0.674 0.072	-0.113 0.099	1.484 0.159	17.7 0.0381	9
5a	-0.666 0.092	2.075 0.155	0.321 0.045	0.482 0.036	8.9 0.2598	7
5b	-1.706 0.122	1.984 0.142	0.860 0.057	0.504 0.036	6.8 0.4521	7
5c	-1.849 0.127	1.772 0.134	1.043 0.066	0.564 0.043	11.0 0.1398	7
5d	-1.651 0.121	2.029 0.150	0.814 0.056	0.493 0.036	14.1 0.0493	7
6a	1.160 0.080	0.884 0.086	-1.313 0.130	1.132 0.110	13.1 0.1580	9
6b	-0.064 0.075	1.307 0.099	0.049 0.057	0.765 0.058	17.5 0.0410	9
6c	0.376 0.071	0.938 0.087	-0.401 0.078	1.066 0.099	11.4 0.2474	9
6d	1.547 0.096	1.304 0.114	-1.187 0.091	0.767 0.067	7.4 0.3868	7
6e	-0.365 0.079	1.453 0.106	0.251 0.055	0.688 0.050	9.8 0.3632	9
6f	1.097 0.084	1.196 0.102	-0.917 0.080	0.836 0.071	7.2 0.5203	8
7a	0.479 0.074	1.111 0.096	-0.431 0.068	0.900 0.078	13.3 0.1494	9
7b	0.938 0.077	0.885 0.089	-1.060 0.113	1.130 0.134	6.7 0.6664	9
7c	1.580 0.101	1.242 0.124	-1.272 0.101	0.805 0.081	7.2 0.3014	6
8	0.611 0.075	1.166 0.096	-0.524 0.069	0.858 0.071	11.4 0.2466	9
9a	1.813 0.096	0.847 0.097	-2.140 0.219	1.180 0.134	15.8 0.0714	9
9b	0.323 0.080	1.680 0.122	-0.192 0.048	0.595 0.043	11.0 0.2753	9
9c	0.711 0.080	1.448 0.109	-0.491 0.058	0.691 0.052	17.9 0.0359	9
9d	0.052 0.073	1.131 0.092	-0.046 0.064	0.885 0.072	18.7 0.0276	9
10a	2.237	1.348	-1.659	0.742	12.0	6

	0.134	0.137	0.118	0.075	0.0610	
10b	0.644 0.067	0.394 0.064	-1.635 0.303	2.539 0.410	10.4 0.3213	9
11a	1.349 0.089	1.164 0.109	-1.159 0.098	0.859 0.080	9.3 0.3176	8
11b	1.128 0.088	1.411 0.115	-0.799 0.066	0.709 0.058	8.4 0.3964	8
12a	-0.032 0.064	0.236 0.049	0.135 0.270	4.235 0.882	12.4 0.1883	9
12b	-0.048 0.066	0.618 0.070	0.078 0.108	1.619 0.184	11.5 0.2402	9
12c	-0.218 0.065	0.353 0.058	0.619 0.206	2.835 0.466	4.8 0.8512	9
12d	-0.280 0.067	0.582 0.068	0.481 0.125	1.720 0.201	6.2 0.7183	9
12e	2.099 0.109	0.755 0.108	-2.781 0.346	1.325 0.189	7.1 0.5317	8
12f	1.733 0.092	0.670 0.095	-2.587 0.336	1.493 0.212	3.6 0.9372	9
13	0.467 0.073	1.102 0.091	-0.424 0.071	0.907 0.075	16.8 0.0516	9
14a	0.461 0.070	0.793 0.082	-0.581 0.096	1.260 0.131	9.0 0.4389	9
14b	0.179 0.066	0.611 0.073	-0.293 0.111	1.637 0.195	19.8 0.0191	9
15	0.720 0.079	1.236 0.097	-0.582 0.065	0.809 0.063	4.6 0.8692	9
17	-0.267 0.069	0.748 0.073	0.357 0.096	1.338 0.131	10.8 0.2916	9
18a	-2.015 0.134	1.543 0.130	1.306 0.084	0.648 0.055	4.8 0.6813	7
18b	-2.076 0.136	1.525 0.132	1.361 0.089	0.656 0.057	7.4 0.3918	7
19a	1.233 0.079	0.740 0.085	-1.667 0.1887	1.352 0.154	11.1 0.2702	9
19b	-0.862 0.079	1.003 0.088	0.860 0.095	0.997 0.087	13.9 0.1243	9
20a	-1.028 0.081	0.954 0.085	1.078 0.110	1.049 0.093	5.5 0.7923	9
20b	-0.185 0.076	1.360 0.105	0.136 0.057	0.735 0.057	8.2 0.5174	9
20c	-1.657 0.104	1.038 0.105	1.597 0.142	0.964 0.098	12.7 0.1770	9
20d	-1.699 0.114	1.236 0.115	1.374 0.105	0.809 0.075	6.6 0.5782	8
21a	1.990 0.106	0.952 0.110	-2.091 0.205	1.051 0.121	5.4 0.7209	8
21b	1.370 0.086	0.571 0.085	-1.219 0.107	0.889 0.081	13.9 0.1264	9
22a	3.146	0.994	-8.357	2.656	6.1	7

	0.154	0.091	2.428	0.773	0.5334	
22b	1.396 0.082	0.571 0.085	-2.444 0.344	1.751 0.261	6.2 0.7257	9
22c	-1.134 0.085	0.994 0.091	1.140 0.110	1.006 0.092	16.4 0.0590	9
23a	-0.199 0.074	1.165 0.091	0.171 0.063	0.859 0.067	7.9 0.5498	9
23b	-0.085 0.072	1.007 0.087	0.085 0.070	0.993 0.086	12.1 0.2062	9
23c	0.363 0.076	1.279 0.098	-0.284 0.059	0.782 0.060	14.2 0.1147	9
24a	-1.727 0.103	0.921 0.098	1.875 0.177	1.086 0.115	6.2 0.7174	9
24b	-2.385 0.150	1.142 0.131	2.088 0.178	0.875 0.100	12.6 0.0826	7

Tableau 2: **Paramètres et erreurs standards de mesure pour le modèle Birnbaum.**

Cette fois-ci, on pourrait donc retenir 67 items.

Nous avons utilisé un test du quotient de vraisemblance (*Likelihood ratio test*) (cf. Rost, 1995 p. 330) pour comparer les 2 options, à savoir les modèles IRT à 1 respectivement à 2 paramètres. Si l'on désigne la vraisemblance du modèle à 2 paramètres par L_0 et celle du modèle à 1 paramètre par L_1 alors, on a l'égalité suivante:

$$-2\ \log\left(\frac{L_0}{L_1}\right) \rightarrow \chi^2,\ \text{avec df} = n_p(L_1) - n_p(L_0).$$

Cela veut dire que, sous réserve d'un nombre assez important de sujets, l'opposé du double du quotient des vraisemblances des modèles respectifs, se comporte comme une loi du chi 2, qui admet comme degrés de liberté la différence du nombre de paramètres des deux modèles, donc dans notre cas, tout simplement le nombre d'items.

Nous trouvons un quotient logarithmé des vraisemblances de 1222,6760 pour 74 degrés de liberté, ce qui est une valeur qui se trouve loin de la frontière de la significativité. Nous pouvons donc conclure que le modèle de Rasch est aussi valable pour décrire nos données que le modèle de Birnbaum.

Cependant, les 36 items retenus ne sont pas suffisants pour construire un test adaptatif informatisé, capable de mesurer la compétence en mathématiques des élèves de 6ème année primaire, avec une assez grande précision à chaque niveau du continuum latent.

Nous avons donc dû retenir quand même le modèle IRT à 2 paramètres, avec, dans un premier temps, les 67 items ayant un bon indice d'adéquation.

3.3 Mise au point de la banque d'items

La prochaine étape consiste à rejeter les items pour lesquels la propriété d'invariance des paramètres n'est pas vérifiée.
A cet effet, nous partageons la population des élèves en 2 groupes, suivant la médiane du score obtenu à l'épreuve de mathématiques. Ainsi, on obtient un groupe de sujets „forts" en mathématiques et un groupe de sujets „faibles". Nous effectuons le calibrage des items séparément pour chaque groupe, ce qui donne les deux tableaux suivants.

	Intercept Erreur	**Discrimination Erreur**	**Difficulté Erreur**	**Dispersion Erreur**	**Chi 2 Indice**	**Degrés de Liberté**
1a	2.025 0.141	0.436 0.094	-4.642 0.986	2.292 0.494	7.8 0.3526	7
1b	3.514 0.262	0.339 0.095	-10.369 2.992	2.951 0.830	4.0 0.6852	6
1c	2.457 0.164	0.477 0.094	-5.147 1.018	2.095 0.412	6.4 0.4921	7
1d	2.600 0.174	0.379 0.092	-6.862 1.683	2.640 0.640	10.9 0.1406	7
1e	-0.616 0.096	0.363 0.078	1.698 0.445	2.754 0.593	16.0 0.0666	9
1f	1.901 0.135	0.371 0.087	-5.125 1.187	2.696 0.632	2.5 0.9303	7
1g	2.919 0.199	0.421 0.105	-6.935 1.746	2.376 0.595	0.8 0.9967	7
1h	0.100 0.091	0.361 0.073	-0.277 0.256	2.769 0.557	6.0 0.7401	9
1i	2.996 0.203	0.546 0.113	-5.485 1.142	1.831 0.378	5.2 0.6339	7
1j	1.146 0.107	0.584 0.104	-1.963 0.356	1.713 0.306	9.3 0.2336	7
2a	1.732 0.125	0.538 0.104	-3.220 0.621	1.859 0.360	7.0 0.4285	7
2b	1.448	0.265	-5.473	3.779	6.0	8

	0.114	0.068	1.446	0.968	0.6490	
2c	1.990 0.135	0.634 0.109	-3.141 0.545	1.578 0.273	6.0 0.5389	7
3a	1.327 0.110	0.420 0.086	-3.156 0.661	2.379 0.484	9.0 0.3424	8
3b	0.688 0.097	0.496 0.101	-1.386 0.312	2.014 0.409	15.2 0.0542	8
4a	1.478 0.118	0.485 0.095	-3.049 0.591	2.062 0.404	2.5 0.9259	7
4b	2.793 0.186	0.704 0.114	-3.970 0.635	1.421 0.229	8.7 0.2756	7
4c	1.797 0.129	0.357 0.085	-5.039 1.195	2.804 0.667	4.7 0.7000	7
4d	1.181 0.109	0.500 0.092	-2.364 0.438	2.002 0.367	7.6 0.3673	7
4e	1.501 0.118	0.470 0.093	-3.197 0.628	2.129 0.420	4.7 0.7009	7
4f	0.582 0.095	0.328 0.071	-1.775 0.457	3.052 0.664	14.6 0.1031	9
4g	2.379 0.157	0.827 0.123	-2.876 0.418	1.209 0.180	18.2 0.0111	7
4h	0.663 0.097	0.335 0.072	-1.976 0.476	2.981 0.639	20.0 0.0104	8
5a	3.289 0.405	6.904 0.763	-0.476 0.028	0.145 0.016	9.2 0.0268	3
5b	0.527 0.174	7.132 0.693	-0.074 0.025	0.140 0.014	9.5 0.0501	4
5c	-0.378 0.212	7.845 1.212	0.048 0.024	0.127 0.020	7.9 0.0466	3
5d	0.755 0.264	10.599 1.364	-0.071 0.028	0.094 0.012	1.4 0.7059	3
6a	1.966 0.134	0.487 0.087	-4.038 0.751	2.054 0.368	3.9 0.7952	7
6b	0.973 0.107	0.771 0.126	-1.263 0.211	1.297 0.213	13.2 0.0665	7
6c	1.170 0.110	0.547 0.107	-2.140 0.411	1.829 0.359	3.4 0.8502	7
6d	2.539 0.167	0.929 0.131	-2.732 0.370	1.076 0.152	8.1 0.3248	7
6e	0.808 0.104	0.823 0.128	-0.982 0.165	1.215 0.189	4.9 0.6706	7
6f	1.949 0.133	0.720 0.109	-2.706 0.409	1.389 0.210	4.0 0.7832	7
7a	1.400 0.117	0.623 0.115	-2.247 0.399	1.605 0.296	1.8 0.9671	7
7b	1.788 0.129	0.456 0.098	-3.919 0.831	2.192 0.473	5.3 0.6256	7
7c	2.658 0.181	0.588 0.111	-4.522 0.826	1.701 0.320	10.9 0.1439	7
8	1.619	0.699	-2.315	1.430	10.9	7

	0.123	0.129	0.408	0.263	0.1422	
9a	2.759 0.185	0.370 0.086	-7.448 1.802	2.699 0.627	19.4 0.0071	7
9b	1.650 0.132	0.990 0.166	-1.667 0.248	1.011 0.170	9.1 0.2473	7
9c	1.843 0.130	0.748 0.139	-2.464 0.442	1.337 0.249	17.2 0.0160	7
9d	0.930 0.101	0.680 0.106	-1.368 0.238	1.470 0.229	26.7 0.0009	8
10a	3.580 0.266	0.572 0.109	-6.259 1.214	1.748 0.333	12.1 0.0966	7
10b	0.932 0.100	0.349 0.075	-2.668 0.611	2.864 0.612	9.8 0.3683	9
11a	2.301 0.155	0.606 0.109	-3.795 0.669	1.649 0.296	13.6 0.0590	7
11b	2.222 0.149	0.694 0.117	-3.204 0.531	1.442 0.244	10.2 0.1764	7
12a	0.084 0.090	0.205 0.052	-0.408 0.451	4.878 1.231	18.4 0.0306	9
12b	0.486 0.095	0.337 0.072	-1.443 0.393	2.969 0.638	13.2 0.1057	8
12c	0.108 0.090	0.197 0.050	-0.549 0.478	5.082 1.284	7.6 0.5762	9
12d	0.222 0.091	0.318 0.069	-0.696 0.317	3.144 0.684	4.1 0.9073	9
12e	2.928 0.202	0.289 0.076	-10.123 2.723	3.457 0.908	9.7 0.2076	7
12f	2.245 0.152	0.409 0.091	-5.489 1.226	2.445 0.547	6.6 0.4731	7
13	1.350 0.111	0.699 0.109	-1.932 0.312	1.431 0.223	7.7 0.3628	7
14a	1.151 0.108	0.463 0.094	-2.488 0.507	2.162 0.438	6.4 0.6092	8
14b	0.727 0.099	0.399 0.085	-1.824 0.420	2.507 0.534	11.2 0.1909	8
15	1.772 0.129	0.723 0.105	-2.449 0.349	1.382 0.200	10.8 0.1486	7
17	0.421 0.094	0.373 0.077	-1.128 0.325	2.678 0.554	11.5 0.2447	9
18a	-0.660 0.100	0.679 0.116	0.971 0.221	1.472 0.252	12.7 0.0791	7
18b	-0.740 0.101	0.703 0.123	1.052 0.233	1.423 0.248	17.5 0.0146	7
19a	1.907 0.132	0.395 0.088	-4.831 1.090	2.533 0.564	12.0 0.0992	7
19b	-0.013 0.095	0.674 0.111	0.020 0.142	1.843 0.244	13.6 0.0934	8
20a	-0.224 0.095	0.558 0.099	0.401 0.191	1.791 0.319	15.2 0.0862	9
20b	0.899	0.907	-0.991	1.103	10.0	7

	0.111	0.144	0.154	0.175	0.1855	
20c	-0.759 0.101	0.635 0.122	1.194 0.286	1.574 0.302	8.5 0.3841	8
20d	-0.583 0.097	0.615 0.110	0.948 0.233	1.626 0.291	6.8 0.5621	8
21a	2.812 0.188	0.558 0.111	-5.042 1.011	1.793 0.358	5.2 0.6401	7
21b	2.269 0.146	0.709 0.127	-3.202 0.581	1.411 0.253	11.2 0.1313	7
22a	3.312 0.243	0.287 0.103	-11.521 4.074	3.479 1.241	5.3 0.3864	5
22b	1.968 0.137	0.252 0.069	-7.807 2.148	3.967 1.087	3.6 0.8205	7
22c	-0.252 0.094	0.527 0.104	0.478 0.209	1.898 0.374	5.2 0.7387	8
23a	0.764 0.099	0.552 0.102	-1.385 0.281	1.812 0.333	3.8 0.8758	8
23b	0.771 0.098	0.487 0.102	-1.581 0.355	2.051 0.428	11.2 0.2622	9
23c	1.497 0.118	0.450 0.101	-3.328 0.738	2.224 0.501	1.3 0.9880	7
24a	-0.894 0.101	0.465 0.093	1.915 0.433	2.150 0.429	6.4 0.6039	8
24b	-1.253 0.110	0.456 0.096	2.750 0.601	2.195 0.460	4.8 0.6891	7

Tableau 3: Paramètres et erreurs standards de mesure de la moitié des élèves forts en mathématiques.

Item	**Intercept Erreur**	**Discrimination Erreur**	**Difficulté Erreur**	**Dispersion Erreur**	**Chi 2 Indice**	**Degrés de Liberté**
1a	1.088 0.108	0.436 0.094	-2.493 0.498	2.292 0.494	7.8 0.3526	7
1b	2.397 0.160	0.339 0.095	-7.074 1.882	2.951 0.830	4.0 0.6852	6
1c	1.318 0.116	0.477 0.094	-2.761 0.495	2.095 0.412	6.4 0.4921	7
1d	1.523 0.121	0.379 0.092	-4.019 0.903	2.640 0.640	10.9 0.1406	7
1e	-1.549 0.124	0.363 0.078	4.265 1.047	2.754 0.593	16.0 0.0666	9
1f	0.838 0.102	0.371 0.087	-2.260 0.504	2.696 0.632	2.5 0.9303	7
1g	2.020 0.145	0.421 0.105	-4.799 1.093	2.376 0.595	0.8 0.9967	7
1h	-0.956 0.105	0.361 0.073	2.647 0.654	2.769 0.557	6.0 0.7401	9
1i	1.849 0.137	0.546 0.113	-3.385 0.625	1.831 0.378	5.2 0.6339	7
1j	0.079	0.584	-0.136	1.713	9.3	7

	0.097	0.104	0.161	0.306	0.2336	
2a	0.493 0.100	0.538 0.104	-0.918 0.206	1.859 0.360	7.0 0.4285	7
2b	0.819 0.099	0.265 0.068	-3.096 0.776	3.779 0968	6.0 0.6490	8
2c	0.686 0.103	0.634 0.109	-1.083 0.197	1.578 0.273	6.0 0.5389	7
3a	0.161 0.095	0.420 0.086	-0.383 0.216	2.379 0.484	9.0 0.3424	8
3b	-0.379 0.097	0.496 0.101	0.764 0.273	2.014 0.409	15.2 0.0542	8
4a	-0.008 0.096	0.485 0.095	0.017 0.198	2.062 0.404	2.5 0.9259	7
4b	1.165 0.116	0.704 0.114	-1.656 0.239	1.421 0.229	8.7 0.2756	7
4c	0.401 0.096	0.357 0.085	-1.124 0.309	2.804 0.667	4.7 0.7000	7
4d	0.245 0.096	0.500 0.092	-0.490 0.184	2.002 0.367	7.6 0.3673	7
4e	0.468 0.098	0.470 0.093	-0.997 0.232	2.129 0.420	4.7 0.7009	7
4f	0.064 0.093	0.328 0.071	-0.196 0.276	3.052 0.664	14.6 0.1031	9
4g	0.258 0.099	0.827 0.123	-0.311 0.115	1.209 0.180	18.2 0.0111	7
4h	-0.381 0.096	0.335 0.072	1.135 0.427	2.981 0.639	20.0 0.0104	8
5a	-2.582 0.230	6.904 0.763	0.374 0.036	0.145 0.016	9.2 0.0268	3
5b	-3.954 0.395	7.132 0.693	0.554 0.038	0.140 0.014	9.5 0.0501	4
5c	-4.441 0.665	7.845 1.212	0.566 0.037	0.127 0.020	7.9 0.0466	3
5d	-5.237 0.780	10.599 1.364	0.494 0.035	0.094 0.012	1.4 0.7059	3
6a	0.560 0.100	0.487 0.087	-1.150 0.236	2.054 0.368	3.9 0.7952	7
6b	-0.808 0.104	0.771 0.126	1.048 0.231	1.297 0.213	13.2 0.0665	7
6c	-0.210 0.097	0.547 0.107	0.385 0.211	1.829 0.359	3.4 0.8502	7
6d	0.844 0.110	0.929 0.131	-0.908 0.134	1.076 0.152	8.1 0.3248	7
6e	-1.209 0.114	0.823 0.128	1.470 0.277	1.215 0.189	4.9 0.6706	7
6f	0.426 0.101	0.720 0.109	-0.592 0.138	1.389 0.210	4.0 0.7832	7
7a	-0.218 0.097	0.623 0.115	0.349 0.183	1.605 0.296	1.8 0.9671	7
7b	0.297	0.456	-0.650	2.192	5.3	7

	0.097	0.098	0.211	0.473	0.6256	
7c	0.713 0.102	0.588 0.111	-1.213 0.231	1.701 0.320	10.9 0.1439	7
8	-0.130 0.097	0.699 0.129	0.186 0.151	1.430 0.263	10.9 0.1422	7
9a	1.127 0.109	0.370 0.086	-3.042 0.661	2.699 0.627	19.4 0.0071	7
9b	-0.669 0.103	0.990 0.166	0.676 0.161	1.011 0.170	9.1 0.2473	7
9c	-0.189 0.097	0.748 0.139	0.253 0.147	1.337 0.249	17.2 0.0160	7
9d	-0.569 0.100	0.680 0.106	0.836 0.203	1.470 0.229	26.7 0.009	8
10a	1.215 0.117	0.572 0.109	-2.125 0.358	1.748 0.333	12.1 0.0966	7
10b	0.498 0.096	0.349 0.075	-1.426 0.344	2.864 0.612	9.8 0.3683	9
11a	0.588 0.100	0.606 0.109	-0.970 0.193	1.649 0.296	13.6 0.0590	7
11b	0.221 0.098	0.694 0.117	-0.319 0.135	1.442 0.244	10.2 0.1764	7
12a	-0.067 0.091	0.205 0.052	0.329 0.469	4.878 1.231	18.4 0.0306	9
12b	-0.449 0.096	0.337 0.072	1.333 0.455	2.969 0.638	13.2 0.1057	8
12c	-0.466 0.094	0.197 0.050	2.369 0.831	5.082 1.284	7.6 0.5762	9
12d	-0.652 0.098	0.318 0.069	2.048 0.594	3.144 0.684	4.1 0.9073	9
12e	1.464 0.116	0.289 0.076	-5.060 1.263	3.457 0.908	9.7 0.2076	7
12f	1.350 0.114	0.409 0.091	-3.300 0.683	2.445 0.547	6.6 0.4731	7
13	-0.165 0.097	0.699 0.109	0.236 0.150	1.431 0.223	7.7 0.3628	7
14a	-0.048 0.095	0.463 0.094	0.104 0.213	2.162 0.438	6.4 0.6092	8
14b	-0.209 0.095	0.399 0.085	0.524 0.293	2.507 0.534	11.2 0.1909	8
15	-0.064 0.098	0.723 0.105	0.089 0.139	1.382 0.200	10.8 0.1486	7
17	-0.810 0.102	0.373 0.077	2.168 0.578	2.678 0.554	11.5 0.2447	9
18a	-2.909 0.214	0.679 0.116	4.282 0.808	1.472 0.252	12.7 0.0791	7
18b	-2.908 0.214	0.703 0.123	4.137 0.788	1.423 0.248	17.5 0.0146	7
19a	0.723 0.101	0.395 0.088	-1.831 0.399	2.564 0.564	12.0 0.0992	7
19b	-1.428	0.674	2.118	1.483	13.6	8

	0.121	0.111	0.412	0.244	0.0934	
20a	-1.578 0.126	0.558 0.099	2.825 0.591	1.791 0.319	15.2 0.0862	9
20b	-0.915 0.108	0.907 0.144	1.009 0.214	1.103 0.175	10.0 0.1855	7
20c	-2.263 0.163	0.635 0.122	3.563 0.737	1.574 0.302	8.5 0.3841	8
20d	-2.560 0.184	0.615 0.110	4.164 0.802	1.626 0.291	6.8 0.5621	8
21a	1.383 0.122	0.558 0.111	-2.480 0.434	1.793 0.358	5.2 0.6401	7
21b	0.706 0.106	0.703 0.127	-0.996 0.182	1.411 0.253	11.2 0.1313	7
22a	3.055 0.201	0.287 0.103	-10.626 3.777	3.479 1.241	5.3 0.3864	5
22b	0.944 0.101	0.252 0.069	-3.746 0.994	3.967 1.087	3.6 0.8205	7
22c	-1.799 0.136	0.527 0.104	3.415 0.752	1.898 0.374	5.2 0.7387	8
23a	-0.948 0.106	0.552 0.102	1.718 0.396	1.812 0.333	3.8 0.8758	8
23b	-0.753 0.102	0.487 0.102	1.544 0.410	2.051 0.428	11.2 0.2622	9
23c	-0.599 0.100	0.450 0.101	1.332 0.417	2.224 0.501	1.3 0.9980	7
24a	-2.363 0.168	0.465 0.093	5.081 1.105	2.150 0.429	6.4 0.6039	8
24b	-3.521 0.282	0.456 0.096	7.729 1.736	2.195 0.460	4.8 0.6891	7

Tableau 3: Paramètres et erreurs standards de mesure de la moitié des élèves faibles en mathématiques.

Le tableau 4 montre bien, qu'effectivement, pour le premier groupe les items sont en moyenne plus faciles que pour le deuxième.

Paramètre	**Moyenne**	**Ecart type**
Discrimination	0.934	1.772
Log (discrimination)	-0.557	0.723
Groupe 1:		
Difficulté	-2.710	2.812
Groupe 2:		
Difficulté	-0.128	2.779

Les distributions différentes pour les deux groupes empêchent cependant de comparer les résultats sans ajustement. Pour pouvoir comparer correctement les difficultés et éliminer les items qui ont une difficulté différente suivant les groupes, le logiciel BILOG-MG permet de

déterminer la difficulté en conservant la même distribution pour les deux groupes. On obtient le résultat suivant :

Item	Groupe 1 Erreur	Groupe 2 Erreur	Différence Erreur
1a	-4.642 0.986	-5.075 0.498	-0.433 1.105
1b	-10.369 2.992	-9.656 1.892	0.713 3.535
1c	-5.147 1.018	-5.343 0.495	-0.196 1.132
1d	-6.862 1.683	-6.601 0.903	0.261 1.910
1e	1.698 0.445	1.684 1.047	-0.014 1.138
1f	-5.125 1.187	-4.842 0.504	0.283 1.290
1g	-6.935 1.746	-7.381 1.093	-0.446 2.059
1h	-0.277 0.256	0.065 0.654	0.342 0.702
1i	-5.485 1.142	-5.967 0.625	-0.482 1.301
1j	-1.963 0.356	-2.718 0.161	-0.755 0.391
2a	-3.220 0.621	-3.499 0.206	-0.279 0.654
2b	-5.473 1.446	-5.678 0.776	-0.205 1.641
2c	-3.141 0.545	-3.665 0.197	-0.524 0.580
3a	-3.156 0.661	-2.965 0.216	0.191 0.695
3b	-1.386 0.312	-1.818 0.273	-0.432 0.414
4a	-3.049 0.591	-2.565 0.198	0.484 0.624
4b	-3.970 0.635	-4.238 0.239	-0.268 0.679
4c	-5.039 1.195	-3.705 0.309	1.334 1.235
4d	-2.364 0.438	-3.072 0.184	-0.708 0.475
4e	-3.197 0.628	-3.579 0.232	-0.382 0.669
4f	-1.775 0.457	-2.778 0.276	-1.002 0.534
4g	-2.876 0.418	-2.893 0.115	-0.017 0.433

4h	-1.976 0.476	-1.447 0.427	0.530 0.640
5a	-0.476 0.028	-2.208 0.036	-1.731 0.046
5b	-0.074 0.025	-2.027 0.038	-1.953 0.045
5c	0.048 0.024	-2.016 0.037	-2.064 0.044
5d	-0.071 0.028	-2.088 0.035	-2.016 0.045
6a	-4.038 0.751	-3.732 0.236	0.306 0.787
6b	-1.263 0.211	-1.534 0.231	-0.271 0.313
6c	-2.140 0.411	-2.197 0.211	-0.057 0.463
6d	-2.732 0.370	-3.490 0.134	-0.757 0.393
6e	-0.982 0.165	-1.112 0.277	-0.130 0.322
6f	-2.706 0.409	-3.174 0.138	-0.467 0.431
7a	-2.247 0.399	-2.233 0.183	0.014 0.439
7b	-3.919 0.831	-3.232 0.211	0.687 0.857
7c	-4.522 0.826	-3.795 0.231	0.726 0.857
8	-2.315 0.408	-2.396 0.151	-0.080 0.435
9a	-7.448 1.802	-5.624 0.661	1.824 1.919
9b	-1.667 0.248	-1.905 0.161	-0.238 0.296
9c	-2.464 0.442	-2.329 0.147	0.136 0.466
9d	-1.368 0.238	-1.745 0.203	-0.377 0.313
10a	-6.259 1.214	-4.707 0.358	1.552 1.265
10b	-2.668 0.611	-4.007 0.344	-1.339 0.702
11a	-3.795 0.669	-3.551 0.193	0.243 0.697
11b	-3.204 0.531	-2.900 0.135	0.303 0.548
12a	-0.408 0.451	-2.253 0.469	-1.845 0.651
12b	-1.443 0.393	-1.248 0.455	0.194 0.601

12c	-0.549 0.478	-0.212 0.831	0.337 0.959
12d	-0.696 0.317	-0.533 0.594	0.163 0.674
12e	-10.123 2.723	-7.642 1.263	2.481 3.002
12f	-5.489 1.226	-5.882 0.683	-0.393 1.403
13	-1.932 0.312	-2.346 0.150	-0.414 0.346
14a	-2.488 0.507	-2.478 0.213	0.011 0.550
14b	-1.824 0.420	-2.058 0.293	-0.234 0.512
15	-2.449 0.349	-2.493 0.139	-0.044 0.376
17	-1.128 0.325	-0.414 0.578	0.713 0.663
18a	0.971 0.221	1.700 0.0808	0.730 0.838
18b	1.052 0.233	1.555 0.788	0.503 0.822
19a	-4.831 1.090	-4.413 0.399	0.418 1.161
19b	0.020 0.142	-0.464 0.412	-0.484 0.436
20a	0.401 0.191	0.244 0.591	-0.158 0.621
20b	-0.991 0.154	-1.573 0.214	-0.581 0.263
20c	1.194 0.286	0.981 0.737	-0.213 0.791
20d	0.948 0.233	1.583 0.802	0.635 0.835
21a	-5.042 1.011	-5.061 0.434	-0.019 1.100
21b	-3.202 0.581	-3.578 0.182	-0.375 0.609
22a	-11.521 4.074	-13.208 3.777	-1.687 5.556
22b	-7.807 2.148	-6.328 0.994	1.479 2.366
22c	0.478 0.209	0.833 0.752	0.355 0.780
23a	-1.385 0.281	-0.863 0.396	0.522 0.485
23b	-1.581 0.355	-1.037 0.410	0.544 0.543
23c	-3.328 0.738	-1.250 0.417	2.078 0.848

24a	1.915 0.433	2.499 1.105	0.583 1.187
24b	2.750 0.601	5.147 1.736	2.397 1.837

Tableau 5: Valeurs ajustées des difficultés des items pour les 2 groupes et différence entre les deux groupes, avec erreurs standards de mesure.

Nous avons rejeté les items pour lesquels la différence de l'indice de difficulté entre les deux groupes était en valeur absolue plus grande que l'écart-type multiplié par 1,96. Cela correspond à un seuil de significativité de 5%.

Les items que nous avons dû ainsi éliminer pour cause de violation de la propriété d'invariance sont les quatre items de la question 5, ainsi que la question 23c. La banque de données dont nous disposons pour construire le test adapté informatisé est donc constitué finalement de 63 items.

La fonction d'information de cette banque d'items est représentée sur la figure 8. On voit bien que nous ne disposons pas d'assez d'items pour mesurer de manière suffisamment exacte la capacité en mathématiques, sauf pour des élèves à capacité moyenne (en fait, une fidélité conventionelle de 0,9 ne peut être acquise que pour des capacités entre –1,2 et 0,2). Il manque surtout des items un peu plus difficiles.

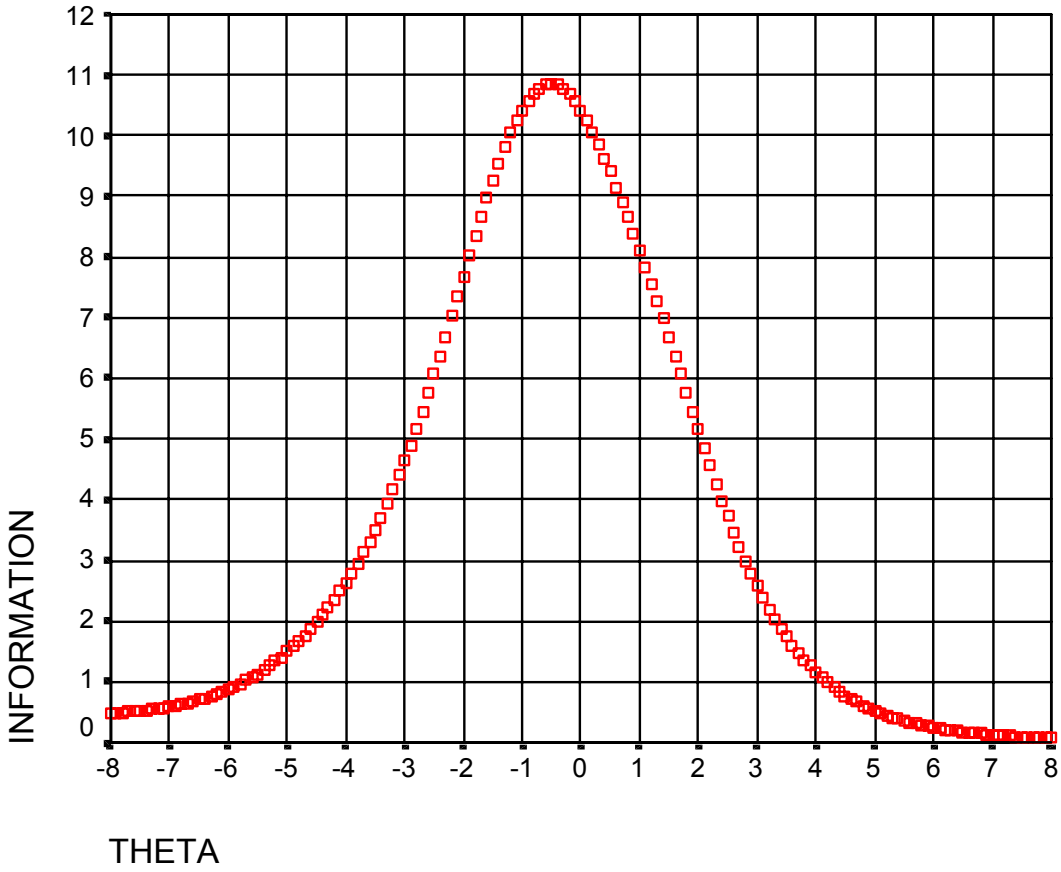


Figure 8 : Fonction d'information de notre banque d'items

Chapitre IV

LES TESTS SUR MESURE ADMINISTRES PAR ORDINATEUR

4.1 Le testing sur mesure

Administrer les mêmes items à tous les sujets, comme on le fait dans les tests collectifs classiques ne fournit pas une discrimination optimale puisque cela ne permet que deux types de tests :

- Les tests à pic, qui présentent des items à difficulté homogène, généralement concentrés autour d'un niveau moyen de difficulté. Ces items sont seulement adaptés à des individus à compétence moyenne. Cela suffit certes pour la majorité de la population, mais un tel test ne permet pas de réaliser des discriminations précises pour des individus à compétence faible ou élevée.
- Les tests rectangulaires, qui sont construits de façon à avoir un nombre égal d'items à chaque niveau de difficulté. Ils permettent de mesurer toutes les aptitudes avec la même précision, mais cette précision est assez faible, puisque, pour chaque sujet, la plupart des items sont non adaptés.

Les tests classiques impliquent donc le dilemme de la bande de fidélité. Ils permettent, soit de mesurer tous les individus avec une même précision assez faible, soit de mesurer certains individus avec une précision élevée et d'autres avec une précision très faible.

Il est clair que ce problème peut être résolu en effectuant un test sur mesure, c'est-à-dire en posant des items différents aux différents sujets, car cela permet d'adapter le niveau de difficulté des items à la compétence de l'individu qui est testé. Un tel test procure des mesures

d'une précision élevée, égale à tous les niveaux du trait latent, à condition de disposer d'une banque d'item assez grande.

En plus de ces avantages psychométriques évidents, cela implique également que la probabilité de répondre correctement reste toujours dans un voisinage de 50%. Ceci présente un avantage psychologique de poids, puisqu'on évite au sujet la frustration en face d'items trop difficiles ou trop faciles.

4.2 Le testing sur mesure par ordinateur

La complexité des modèles de testing sur mesure implique qu'en pratique un tel test ne peut être administré que par ordinateur.

Les tests sur mesure administrés par ordinateur (CAT – *computerized adaptive testing*) ont été proposés par Lord (1971), Owen (1975) et Weiss (1976). Ils permettent de mesurer plus précisément la capacité des sujets, tout en lui proposant moins d'items à résoudre. Depuis, il y a eu des progrès significatifs dans le développement et l'implimentation des CAT's, en partie grâce au développement rapide de la technologie informatique (cf. Wainer 1990). Ces procédures de tests informatisés utilisent la théorie de réponse par item pour assembler les items, tester les individus et calculer leur performance.

L'ordinateur remplit plusieurs fonctions dans ce processus.

- Il sélectionne un item.
- Il présente l'item au sujet.
- Il enrégistre la réponse, l'évalue et la comptabilise.
- Il sélectionne l'item suivant en tenant compte de l'information obtenue à partir de toutes les réponses précédentes.
- Il met fin à ce processus continu lorsqu'un critère de terminaison est atteint.

4.3 L'algorithme du test

L'algorithme typique d'un test adaptatif administré par ordinateur comporte les trois étapes suivantes (voir figure 8 ci-dessous).

1. L'initialisation : Donner une estimation initiale de la compétence ; celle-ci va spécifier quel item doit être administré en premier.
2. Le corps de l'algorithme : Estimer la compétence après chaque réponse à un item. Administrer ensuite celui des items restants, qui procure l'information maximale pour cette compétence estimée.
3. La terminaison : Arrêter le test lorsque la précision de l'estimation de la compétence est adéquate ou lorsqu'un certain nombre d'items a été administré.

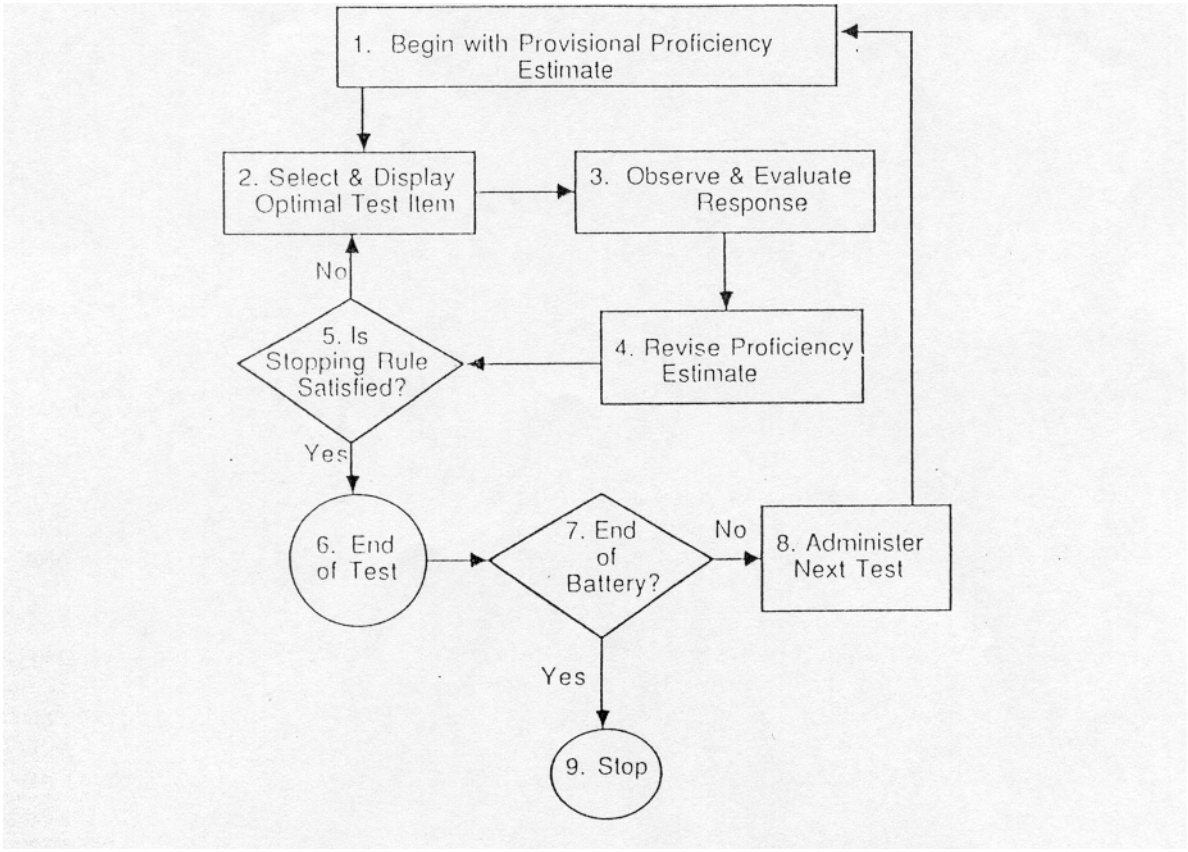


Figure 9 : Organigramme donnant la structure d'un test sur mesure (dans Wainer, 1990).

4.4 Le premier item administré

Pour le premier item administré, on ne peut bien sûr pas baser le principe de sélection sur des réponses à des items antérieurs. Lorsque l'on ne connaît rien de la capacité de l'individu testé, on choisit généralement la moyenne de la population comme estimation initiale de θ. Par conséquent, l'ordinateur commence par administrer au sujet un item de difficulté moyenne.

Si l'on dispose d'informations concernant le niveau scolaire, l'âge, …, on peut utiliser comme estimation initiale de la compétence la moyenne d'un groupe de sujets qui présentent les mêmes caractéristiques. Cela permet donc de choisir un premier item qui est, a priori, le plus approprié au niveau de compétence du sujet.

Si le sujet a déjà accompli un test différent, mesurant le même trait latent, respectivement un trait latent ayant une corrélation positive avec celui en question, on peut exploiter des relations éventuellement établies entre ces deux tests. Par exemple, un sujet, dont la performance à un test de vocabulaire est élevée, devrait avoir également un score élevé dans un test basé sur la compréhension d'un test. Obtenir une bonne estimation initiale à partir de l'estimation finale du test précédent est un problème statistique facile à résoudre.

Cependant, les propriétés psychométriques de l'IRT impliquent que, de toute façon, un mauvais choix du premier item n'aura que peu d'incidences sur le résultat final, à moins que le test soit vraiment très court.

Pour notre logiciel, nous avons choisi la question 12.b) comme item initial. En effet, comme elle présente une difficulté moyenne (0,078) et une discrimination pas trop forte (0,618), elle permet de donner une estimation acceptable dans une large bande du trait latent, ce qui revient à dire qu'elle permet de donner une première estimation de la capacité de tous les sujets, quel que soit leur niveau de capacité.

4.5 Les stratégies de sélection des items

Les stratégies de branchement variable (*variable-branching strategies*) basées sur l'IRT permettent une utilisation optimale des tests sur mesure.

Il y a actuellement deux stratégies fondamentales (plus de nombreuses variantes) de sélection des items basées sur la théorie de réponse par item, qui sont généralement utilisées dans les tests sur mesure administrés par ordinateur : la stratégie dite d'information maximale et la stratégie bayesienne.

4.5.1 La stratégie d'information maximale

Cette stratégie de sélection des items consiste à sélectionner les items qui procurent le maximum d'information à un niveau estimé du trait.

Après avoir estimé la compétence du sujet, en tenant compte de ses réponses à toutes les questions précédentes, on évalue la fonction d'information des différents items à ce niveau de compétence, et on choisit comme item suivant celui qui procure l'information maximale parmi les items non encore utilisés. Lorsque le sujet a répondu à ce nouvel item, on réestime θ et on répète la procédure. A la fin du test, la dernière estimation de θ sera le score obtenu par le sujet. De tels scores sont tous sur la même échelle pour tous les individus, même si on leur a administré des sets d'items totalement différents.

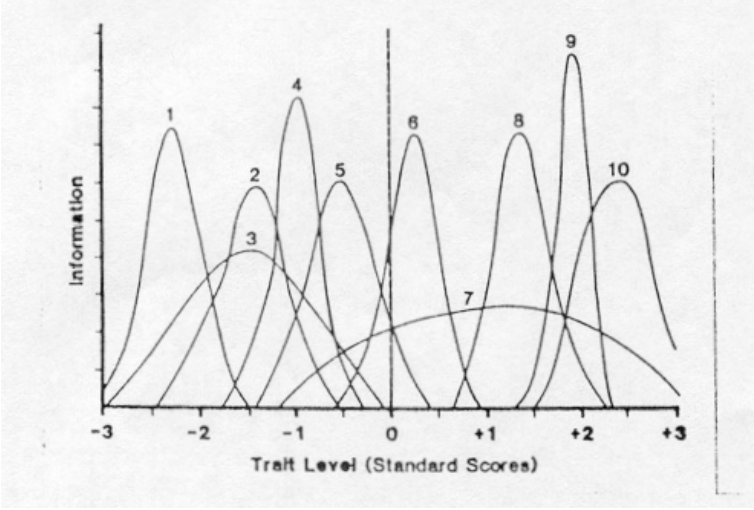


Figure 10 : Exemple hypothétique des IIF de 10 items (dans Weiss, 1985)

Clarifions ce procédé par un exemple. La figure 9 représente les fonctions d'information d'un test composé de 10 items, numérotés de 1 à 10. La ligne verticale représente le niveau de compétence initial estimé à 0,0 pour l'individu testé. La stratégie de l'information maximale consiste alors à déterminer l'item pour lequel l'information au niveau 0.0 est maximal, ce qui, dans notre exemple, est l'item numéro 6. Cet item est alors administré au sujet, sa réponse est noté et une nouvelle estimation pour sa compétence est effectuée.

La théorie de réponse par item implique qu'une réponse correcte à un item augmente le niveau de compétence estimé tandis qu'un réponse incorrecte le fait diminuer.

Supposons que le sujet ait répondu correctement à l'item numéro 6. L'estimation de sa compétence sera donc supérieure à l'estimation initiale. Supposons qu'elle est de 1,5. Le prochain item administré sera celui qui procure l'information maximale au niveau 1,5, donc l'item numéro 8. Supposons que, cette fois-ci le sujets fait une réponse erronée, ce qui fait chuter l'estimation de sa capacité et supposons que la nouvelle estimation est de 0,5. L'item qui fournit le plus d'information à ce niveau parmi les items restants est l'item numéro 7 (l'item numéro 6 procure certes plus d'information au niveau 0,5, mais il a déjà été administré). On administre donc l'item numéro 7 au sujet, on suppose qu'il répond correctement et que la nouvelle estimation de compétence est de 1,0. A ce niveau, tous les items non encore administrés fournissent une information nulle. On arrête alors le test, comme aucun des items restants ne fournit la possibilité de différencier entre les niveaux du trait latent au voisinage de la dernière estimation de la compétence du sujet, et on lui attribue un score de 1,0.

En plus de fournir une estimation du niveau de compétence, la théorie de réponse par item permet de calculer l'erreur de mesure associée à cette estimation. Comme l'erreur de mesure est inversement proportionnelle à l'information, la procédure de sélection des items dite d'information maximale fournit donc une estimation de la capacité, avec une erreur de mesure minimale, donc aussi précise que possible. Si l'on dispose d'un nombre suffisant d'items de haute qualité, il devient possible de mesurer la capacité de tous les individus avec une précision égale, déterminable à volonté.

Il est clair que ceci ne peut pas être le cas pour l 'exemple de test qu'on a donné ci-dessus. Répondre à seulement 3 questions ne suffit d'habitude pas à donner une estimation de la compétence avec une erreur de mesure assez petite, à l'exception du cas où ces 3 items procureraient des informations extraordinairement élevées.

En pratique, pour pouvoir construire un test de bonne qualité, il faut disposer d'une banque d'items assez grande pour pouvoir effectuer des mesures précises à chaque endroit du trait latent de compétence.

4.5.2 La table d'information

Un ordinateur qui administre un CAT doit estimer l'information que procure chacun des items de la banque au niveau actuel de θ, à chaque fois qu'il sélectionne l'item suivant. Ainsi, il risque de devoir effectuer les mêmes calculs, à chaque fois qu'un nouveau sujet est testé. Pour éviter cela, beaucoup de logiciels utilisent une table d'information (*Info table*) placée dans la mémoire de l'ordinateur. Cette table contient la liste des items, ordonnés en fontion de la quantité d'information qu'ils procurent aux différents niveaux de la compétence. La figure 11 montre une table d'information correspondant à un test hypothétique, le "GCAT Mathematical Knowledge test". De tels tables sont en principe conçues pour être lues uniquement par un ordinateur et comportent pour chaque niveau de compétence les numéros des items, qui en ordre décroissant, maximisent l'information à ce niveau donné. La colonne "Maximum Information" contient l'information maximale du test au niveau donné de compétence.

θ	Items in order of preference from left to right, at each value of θ															Information
−2.25	955	50	948	915	136	57	64	331	296	362	938	35	366	106	349	5.2
−2.12	948	955	50	915	136	57	296	331	64	35	362	106	938	366	349	5.6
−2.00	948	955	50	136	915	35	296	331	106	57	64	366	362	349	938	5.8
−1.88	948	955	50	35	136	296	106	331	915	349	64	57	366	362	938	5.9
−1.75	948	955	35	50	106	296	136	349	331	915	366	64	57	377	362	5.8
−1.62	948	35	106	349	955	296	50	331	136	366	915	377	64	57	321	5.6
−1.50	948	35	349	106	296	955	331	136	50	377	321	366	64	915	68	5.5
−1.38	35	349	948	106	296	321	377	331	955	280	136	68	366	50	103	5.5
−1.25	349	35	948	106	321	377	280	296	68	331	103	366	136	955	50	5.4
−1.12	349	321	35	280	106	377	103	948	68	296	341	146	331	366	144	5.5
−1.00	321	349	280	35	103	377	341	68	146	106	144	296	948	331	42	5.6
−0.88	321	280	103	341	349	144	146	35	377	68	106	42	296	948	67	5.9
−0.75	144	321	341	280	103	146	349	42	377	35	68	106	67	1	296	6.3
−0.62	144	341	321	146	103	280	42	349	377	68	35	1	67	8	106	6.5
−0.50	144	341	146	42	321	103	280	1	377	68	349	67	8	128	35	6.8
−0.38	144	42	341	146	321	103	280	1	128	8	328	377	67	107	68	7.1
−0.25	144	42	341	146	321	103	128	280	1	88	328	8	49	107	67	7.5
−0.12	144	42	341	146	88	128	321	328	49	103	1	376	8	107	280	7.8
0.00	144	42	88	146	128	376	341	49	328	8	1	107	321	103	383	8.0
0.12	88	42	376	144	128	49	328	146	383	341	365	8	107	1	85	8.3
0.25	88	376	42	365	383	49	128	144	328	85	335	146	107	8	341	8.9
0.38	376	88	365	383	85	49	42	128	126	335	328	144	107	146	241	9.4
0.50	126	365	376	85	346	383	215	88	335	49	241	128	362	328	42	10.1
0.62	126	365	376	85	346	383	215	88	335	49	241	128	262	328	42	11.1
0.75	126	365	346	215	376	85	383	88	241	335	49	262	355	399	128	11.6
0.88	346	126	215	365	85	376	383	241	355	335	88	262	228	399	49	11.8
1.00	346	215	126	365	85	355	376	228	347	241	262	209	383	399	335	11.7
1.12	346	215	126	347	228	355	365	85	209	376	241	399	262	160	251	11.3
1.25	346	347	215	228	355	126	160	209	365	251	85	399	262	376	241	10.9
1.38	347	160	346	228	355	215	237	209	251	126	365	399	263	378	262	10.8
1.50	160	347	237	228	346	355	251	209	215	378	263	399	126	262	365	10.7
1.62	160	237	347	228	251	355	209	346	215	378	263	186	399	262	126	10.4
1.75	160	237	347	228	251	355	209	186	346	378	263	399	215	262	241	10.0
1.88	160	237	347	251	228	186	355	209	378	263	346	399	215	262	241	8.9
2.00	160	237	186	251	347	228	355	378	209	263	399	346	262	215	115	7.7
2.12	160	237	186	251	347	228	355	378	263	209	399	346	262	115	215	6.3
2.25	160	186	237	251	347	228	378	355	263	209	399	262	115	346	241	5.0

Figure 11 : Exemple d'une table d'information (dans Wainer, 1990).

Pour notre logiciel, nous avons cependant choisi de ne pas utiliser cette méthode, et de faire répéter le calcul à chaque fois par l'ordinateur, ce qui, avec la capacité actuelle des ordinateurs, ne pose plus aucun problème. Ce procédé présente le grand avantage qu'on n'a pas besoin de construire et d'entrer dans la mémoire de l'ordinateur un nouveau tableau, toutes les fois qu'on veut ajouter ou retirer des items. Ainsi, le logiciel reste plus facilement adaptable à la création d'autres tests.

4.5.3 La stratégie de sélection bayesienne d'Owen

La stratégie de sélection bayesienne permet d'utiliser l'information antérieure sur le sujet plus complètement que la stratégie de sélection d'information maximale. Dans cette approche,

on maximise la précision postérieure, c'est-à-dire, on recherche l'item qui, lorsqu'il sera administré, réduira au maximum la variance future de l'estimation de θ.

Les équations établies par Owen (Owen 1975) sont basées sur une distribution normale de la population et sur une approximation par la loi normale de la contribution de chaque item à l'estimation de la compétence.

L'avantage de cette stratégie est que la programmation n'est pas aussi fastidieuse. Cette méthode a été très utilisée il y a une vingtaine d'années, en raison de la puissance limitée des ordinateur.

Le grand désavantage de cette méthode est que les estimations de θ varient en fonction de l'ordre dans lequel les items sont administrés. Ceci résulte de l'approximation par la loi normale de la contribution non-normale de chaque item à la précision postérieure. Pour cette raison, et grâce à la puissance grandissante des ordinateurs, la stratégie de sélection bayesienne est aujourd'hui nettement moins utilisée.

En fait, les programmes de CAT sont de nos jours souvent hybrides et reposent sur une estimation bayesienne de la compétence, mais sur une stratégie d'information maximale pour la sélection des items. C'est cette méthode que nous utilisons également pour notre logiciel.

4.6 La règle de terminaison

Avec chaque item supplémentaire administré, l'estimation de la compétence représente mieux le niveau réel du trait latent, comme l'erreur standard de mesure diminue avec chaque item.

Un test sur mesure peut donc se terminer quand une précision déterminée est atteinte, mais aussi, après un nombre fixé d'items, ou lorsqu'un temps déterminé s'est écoulé. En pratique, lorsqu'on veut atteindre un degré de précision déterminé, on impose souvent simultanément un nombre maximal d'items administrés car, dans certains cas, la banque d'items peut être épuisée avant que la précision voulue ne soit atteinte.

Tester tous les sujets avec le même degré de précision présente l'avantage d'obtenir des scores estimés qui se conforment au présupposé d'une erreur de mesure égale pour tous

les sujets de la théorie classique des tests. De plus, lorsque l'on fait des analyses statistiques qui prennent l'erreur de mesure en compte, les résultats sont plus faciles à manipuler.

En fait, les critères de terminaison varient en fonction des applications du test, et il y en existe encore d'autres. Si le test est par exemple administré pour classer les individus, un critère de terminaison reposant sur une précision de mesure égale n'a pas beaucoup de sens. Ce qui est important dans ce cas, c'est la classification. Ainsi, on impose une règle de fin différente, basée sur la probabilité d'erreur de classification (Weiss & Vale, 1987).

Pour notre logiciel, nous avons choisi un critère de terminaison basé sur une erreur de mesure déterminée, doublé d'un critère destiné à empêcher l'administration d'items qui ne contribuent plus à une augmentation significative de l'information totale. Ce deuxième critère a été nécessaire, puisque la banque d'items dont on dispose ne contient pas assez d'items discriminatifs pour des capacités significativement plus faibles respectivement plus fortes que la moyenne.

Chapitre V

LE LOGICIEL

5.1 Choix de la plate-forme de programmation

Le test doit être administré par un logiciel adapté, pour présenter, enregistrer, contrôler, traiter, comptabiliser, emmagasiner la réponse du sujet, puis calculer et afficher son score.

Le logiciel habituellement utilisé est le MicroCAT™ (cf. Assessment Systems Corporation, 1995)), un programme qui accomplit chacune des étapes énoncées précédemment. Le désavantage de ce programme est cependant qu'il permet de générer uniquement des items à choix multiples, les possibilités graphiques du logiciel étant inexistantes. Puisque notre test de mathématiques contient outre de nombreux items à réponse libre aussi des items géométriques, qui nécessitent des possibilités graphiques importantes de la part du programme, nous avons décidé d'utiliser plutôt la plate-forme de programmation multimédia Quest Net+ for Windows™ de Allen Communication, qui présente des possibilités graphiques beaucoup plus importantes. Pour une introduction rapide à ce logiciel, le lecteur pourra consulter Allen Communication (1997).

Dans la suite, nous donnons une description détaillée de l'algorithme utilisé pour notre logiciel. Le langage de programmation utilisé est le Quest C, une version un peu modifiée du langage C.

Comme nous l'avons précisé dans le chapitre 4, le programme commence par administrer la première question au sujet. Puis il enregistre sa réponse, détermine si celle-ci est correcte ou non, calcule une estimation de sa capacité, choisit le prochain item en fonction

de cette estimation, administre cet item, et ainsi de suite, jusqu'à ce que le critère de terminaison soit vérifié. Alors le programme affiche la capacité en mathématiques du sujet et s'arrête. Examinons ces étapes une à une.

5.2 L'initialisation du programme

Au début du programme, nous définissons les variables globales dont nous avons besoin pour la suite. La première ligne définit une variable "constante" appelée "NITEMS", qui est égale au nombre des items, que comporte la banque d'items, à savoir 63. Nous l'utilisons à chaque fois qu'intervient un tableau dont la dimension dépend de ce nombre. Ceci permet d'adapter facilement notre logiciel à d'autres tests, respectivement d'ajouter ou d'enlever certains items. Pour cela, il suffit de remplacer le nombre 63 dans la définition de la variable "NITEMS" par le nouveau nombre d'items et bien sûr d'ajouter respectivement d'enlever les modules contenant les graphiques des items en question.

Le début du programme est donc le suivant :

```
#define NITEMS 63
float inftot=0.0, theta=0.0;
float par [NITEMS] [2]={{0.668,-2.206}, {0.705,-4.112}, {0.765,-2.339}, {0.671,-2.947},
                        {0.578,1.983}, {0.650,-1.978}, {0.702,-3.444}, {0.672,0.744},
                        {0.853,-2.734}, {0.798,-0.626}, {0.799,-1.244}, {0.411,-2.621},
                        {0.939,-1.301}, {0.736,-0.893}, {0.75,-0.075}, {0.937,-0.682},
                        {1.142,-1.629}, {0.814,-1.254}, {0.724,-1.235}, {0.433,-0.597},
                        {1.487,-0.8}, {0.884,-1.313}, {0.938,-0.401}, {1.304,-1.187},
                        {1.453,0.251}, {1.196,-0.917}, {1.111,-0.431}, {0.885,-1.06},
                        {1.242,-1.272}, {1.166,-0.524}, {0.847,-2.14}, {1.68,-0.192},
                        {1.348,-1.659}, {0.394,-1.659}, {1.164,-1.159}, {1.411,-0.799},
                        {0.236,0.135}, {0.618,0.078}, {0.353,0.619}, {0.582,0.481},
                        {0.755,-2.781}, {0.67,-2.587}, {1.102,-0.424}, {0.793,-0.581},
                        {1.236,-0.582}, {0.748,0.357}, {1.543,1.306}, {1.525,1.361},
                        {0.74,-1.667}, {1.003,0.86}, {0.954,1.078}, {1.36,0.136},
                        {1.038,1.597}, {1.236,1.374}, {0.952,-2.091}, {0.571,-1.219},
```

```
                {0.994,-8.357}, {0.571,-2.444}, {0.994,1.14}, {1.165,0.171},
                {1.007,0.085}, {0.921,1.875}, {1.142,2.088}};
float info[NITEMS], corr[NITEMS] ;
int ens [NITEMS];
int nom;
char fname[20];
```

Le tableau appelée "par" , à 63 lignes et deux colonnes contient les paramètres des items qu'on a établi préalablement à l'aide du logiciel BILOG.

La variable "theta" contient les estimations successives de la capacité en mathématiques du sujet, ou, autrement dit, du score obtenu lors du test. Elle est initialisée à zéro, puisqu'on ne dispose pas d'information préalable sur le sujet en question. La variable "inftot" contient la totalité de l'information dont on dispose à tout moment du test. Elle est évidemment également initialisée à zéro. Les autres variables sont des variables techniques dont nous allons expliquer l'utilité au fur et à mesure qu'elles seront utilisées dans le programme.

Après avoir affiché l'écran de départ, qui explique aux candidats comment utiliser le logiciel, le logiciel branche sur le module "début", qui contient le programme suivant

```
nom=1;
ens[0]=38;
strcpy (fname, "q38");
BranchToFrame(fname);
```

La variable "nom" indique combien d'items ont déjà été administrés, plus précisément, nom=1 indique que le logiciel va administrer le premier item. Le tableau "ens", à 1 ligne et 63 colonnes, contient le numéro des items qui ont déjà été administrés. Ens[0]=38 veut dire que le premier élément du tableau (en C, le j-ième élément d'un tableau porte l'indice j-1) est le nombre 38. En effet, l'item numéro 38 correspond à la question 12b, qui est la première posée. Les deux dernières lignes sont nécessaires pour faire un branchement sur le module "q38", qui contient l'item 38.

5.3 Le corps principal du programme

5.3.1 Les modules des items

Le logiciel comporte les modules “q1” à “q63”, qui contiennent chacun un item. Chacun de ces modules propose la question correspondante au sujet, enregistre sa réponse, détermine si la réponse est correcte ou non et branche finalement sur le module “Good job”, si la réponse est correcte, respectivement sur le module “Oops”, si la réponse est erronée.

Quest Net+ for Windows™ permet de générer aussi bien des questions à choix multiples que des questions à réponse libre. Il permet même des questions à graphiques interactifs, ce qui est très utile pour des items de géométrie, par exemple, si le sujet doit construire des droites parallèles à d’autres.

Le module “Good job“ ne contient que les 2 lignes de programmation suivantes

corr[nom-1]=1.0;
Branch to... frame (Algorithmes)

Il sert à retenir que la réponse à la question en cours est correcte et a brancher ensuite sur le module “Algorithmes”. En effet le tableau “corr”, à 1 ligne et 63 colonnes, sert a retenir si les réponses du candidat sont justes ou fausses. Si la réponse au j-ième item est correcte, le module “Good job” inscrit comme j-ième élément du tableau “corr” le nombre 1, si la réponse est incorrecte, le module “Oops” inscrit j-ième élément du tableau “corr“ le nombre 0.

Puis, on branche sur le module “Algorithmes” qui contient tous les algorithmes nécessaires à l’estimation de θ, à la détermination du prochain item et à la terminaison du test.

5.3.2 L’estimation de θ

Puisque nous utilisons la méthode d'estimation modale bayesienne, la k-ième estimation de θ est donnée d'après le paragraphe 2.6.2, par la valeur la plus élevée de la fonction postérieure de vraisemblance L, définie par

$$L(\theta) = \frac{\exp\left[\sum_{i=1}^{k-1} a_{j_i}(b_{j_i} - \theta)(1 - u_{j_i})\right]}{\prod_{i=1}^{k-1}\left[1 + \exp\left[-a_{j_i}(\theta - b_{j_i})\right]\right]} g(\theta).$$

où la fonction g désigne la distribution de θ. Or, nous supposons que la compétence de la population totale est normalement distribuée, ce qui présente l'avantage que nous disposons alors de procédés statistiques simples pour comparer les différents scores des sujets. La fonction g est alors définie par

$$g(x) = \frac{1}{\sqrt{2\pi}} \frac{e^{-x^2}}{2}.$$

Puisque chercher le maximum d'une fonction revient à chercher le maximum du logarithme de cette fonction, il suffit alors de calculer le logarithme de la fonction L, de le dériver et de déterminer le point pour lequel cette dérivée s'annulle. Donc il faut résoudre l'équation

$$-\sum_{i=1}^{k-1} a_{j_i}(1 - u_{j_i}) + \sum_{i=1}^{k-1} \frac{a_{j_i} \exp\left[a_{j_i}(b_{j_i} - \theta)\right]}{1 + \exp\left[a_{j_i}(b_{j_i} - \theta)\right]} - 2\theta = 0,$$

ce qui revient à résoudre l'équation,

$$f(\theta) = 0,$$

où

$$f(\theta) = \sum_{i=1}^{k-1} \frac{a_{j_i} \exp\left[a_{j_i} b_{j_i}\right] \exp\left[-a_{j_i}\theta\right]}{1 + \exp\left[a_{j_i} b_{j_i}\right] \exp\left[-a_{j_i}\theta\right]} - \sum_{i=1}^{k-1} a_{j_i}(1 - u_{j_i}) - 2\theta.$$

L'algorithme numérique le plus approprié pour résoudre cette équation nous semble être l'algorithme de dichotomie. En effet d'après la modélisation du problème, nous savons que la solution de l'équation est unique (puisque le score obtenu dans le test est déterminé de façon unique par la théorie de réponse par item), et en plus, nous savons qu'elle ne se trouve pas trop éloignée de l'estimation précédente (en fait la distribution normale du score implique que plus de 99% de la population auront un score entre –4 et +4). L'algorithme de dichotomie semble donc bien adapté. De plus, la fonction f n'est pas contractante, ce qui exclut l'utilisation des autres algorithmes classiques comme la méthode de Newton ou de Lagrange.

Le fonction “funct(x)” permet de calculer f(x), tout en tenant compte des items administrés jusque là et des réponses que le sujet a données à ces items.

```
int i;
float funct (float x)
    {float res,a,b;
  int itemn;
  res=-2.0*x;
  for (i=1 ; i<=nom ; i++)
     {itemn=ens[i-1];
    a=par [itemn-1][0];
    b=par [itemn-1][1];
     res+=(a*exp(a*b)*exp(-a*x))/(1.0+exp(a*b)*exp(-a*x))-(a*(1.0-corr[i-1]));}
       return res;}
```

En effet, pour tout i entre 1 et le nombre actuel d’items administrés, la variable “itemn” contient le numéro du i-ième item administré, ce qui permet de pouvoir trouver les paramètres des items dans le tableau “par”.

L’expression 1.0-corr[i-1] correspond bien à u(i) puisque cette expression vaut 0, si le sujet a répondu correctement au i-ième item, respectivement 1, si le sujet a mal répondu au i-ième item.

L’algorithme de dichotomie “algo(x)” ne comporte pas deux variables correspondant aux extrémités de l’intervalle dans lequel on cherche la solution, comme c’est le cas dans l’algorithme de dichotomie classique, mais seulement une seule, qui correspond à la dernière estimation de la capacité et qui détermine le milieu de l’intervalle dans lequel on cherche la solution. De plus, nous savons que notre équation a exactement une solution réelle. Ceci nous permet d’agrandir tout simplement l’intervalle de recherche, si l’intervalle de départ [θ-2, θ+2] ne contient pas la solution. Ceci nous conduit à l’algorithme suivant

```
float algo (float x_0)
{float res, a, b;
  a=x_0-2.0;
```

```
    b=x_0+2.0;
    do {
      if (funct(a)*funct(b)<0)
      {if (funct(a)*funct((a+b)/2.0)<0)  b=(a+b)/2.0;
      else a=(a+b)/2.0;}
      else a=a-1.0, b=b+1.0;
          }
    while (b-a>0.005);
    res=a;
  return res;}
```

La nouvelle estimation de θ est alors tout simplement obtenue en appliquant l'algorithme de dichotomie à l'ancienne estimation :

```
theta=algo(theta);
```

5.3.3 Détermination de l'item suivant

La clé de la détermination de l'item suivant est constituée par les fonctions d'information des différents items au niveau actuel de l'estimation de la capacité. Il nous faut donc une fonction à deux variables I(x,n), qui calcule l'information fournie par le n-ième item au niveau de capacité x.

```
float I (float x, int n)
{float res,a,b;
  a=par [n-1][0];
  b=par [n-1][1];
  res=(a*a*exp(-a*x+a*b))/((1.0+exp(-a*x+a*b))*(1.0+exp(-a*x+a*b)));
return res;}
```

L'item suivant est alors celui des items non encore administrés qui fournit la plus grande information au niveau θ actuel. Pour le déterminer, le programme calcule d'abord l'information de tous les items au niveau θ et la place dans le tableau à une ligne et 63 colonnes "info".

```
for (i=1 ; i<=63 ; i++)
{info[i-1]=I(theta, i);}
```

Ensuite, il remplace l'information des items déjà administrés par 0, pour les éliminer pour la suite.

```
for (i=1 ; i<=nom ; i++)
{info[ens[i-1]-1]=0.0;}
```

Le prochain item administré sera alors celui pour lequel l'information au niveau actuel de θ est maximale. L'algorithme servant à déterminer ce maximum est le suivant :

```
float sup;
int item=1;
sup=info[0];
for (i=2; i<=63 ; i++)
{if (info[i-1]>sup) {sup=info[i-1];
                    item=i;}}
```

On commence par attribuer l'information du premier item à la variable "sup" et son numéro à la variable "item". Puis on compare un élément après l'autre du tableau d'information avec le contenu de la variable "sup", et à chaque fois qu'on trouve une information plus grande, on attribue cette information-là à la variable "sup" et le numéro de l'item correspondant à la variable "item". Ensuite, le contenu de la variable "item" contient le numéro du prochain item à administrer, qui est introduit alors dans le tableau "ens" de la liste de tous les items déjà traités. De plus, la variable "nom", qui indique le nombre de questions déjà posées, est incrémenté d'une unité.

```
ens[nom]=item;
nom++;
```

Finalement, le logiciel branche vers le module contenant l'item déterminé par le procédé ci-dessus, à l'aide du programme suivant, qui est nécessaire à cause des limitations du langage Quest C.

```
strcpy (fname, "q");
char nstr [5];
itoa (item, nstr, 10);
strcat (fname, nstr);
BranchToFrame(fname);
```

5.4 Le critère de fin du programme

Le test sera fini dés que l'information totale fournie par tous les items administrés dépasse la valeur 10. Ceci implique que l'erreur standard de mesure est de 0,3162, ce qui correspond à une fidélité conventionnelle de 0,9.

La fonction à une variable Inf(x) calcule, après l'administration de chaque item, l'information totale du test fournie jusque là au niveau x de la dernière estimation de θ, en utilisant la fonction I définie ci-dessus.

```
float Inf (float x)
{float res=0.0;
  int itemn;
  for (i=1 ; i<=nom ; i++)
  {itemn=ens[i-1];
  res+=I(theta, itemn);}
return res;}
```

Puis, cette information totale est attribuée à la variable "inftot" et, si l'information totale est supérieure à 10, le logiciel branche sur le module "résultat", qui arrête le test et affiche le score obtenu (égal à la dernière estimation de la capacité θ). Remarquons que la variable "inftot" n'est pas vraiment nécessaire. Nous aurons très bien pu la laisser de côté et prendre

comme critère de fin tout simplement « if (Inf(theta)>10.0) », mais nous avons considéré qu'en introduisant cette variable, le programme devient plus lisible.

```
        inftot=Inf(theta);
        if (inftot>10.0)
  {strcpy (fname, "résultat");
  BranchToFrame(fname);}
```

Mais ce critère de terminaison n'est pas suffisant. Puisque notre banque d'items n'est pas assez complète pour disposer d'items en quantité et qualité suffisante sur toute la bande du continuum latent, il faut ajouter un deuxième critère, servant à couvrir le cas où tous les items pouvant apporter une information significative à un niveau donné ont déjà été administrés. Nous avons donc ajouté un deuxième critère, qui stipule que, dès que l'information maximale fournie par le dernier item (qui se trouve dans la variable "sup") tombe en dessous de 0,25, le test s'arrête. En fait, le logiciel branche sur le module "erreur" qui arrête le test, affiche le score obtenu, ainsi qu'une remarque que la mesure d'erreur peut être non négligeable, en raison d'un nombre non suffisant d'items disponibles à ce niveau du trait latent.

```
  if (sup<0.005)
  {strcpy (fname, "erreur");
  BranchToFrame(fname);}
```

5.5 Les modules des items

Les items proprement dits sont placés dans les modules "q1" à "q62". La structure du programme est conçue de façon à ce que chaque item doit se trouver dans un seul module et que ces modules doivent être appelé "q1", "q2", … et ainsi de suite jusqu'au dernier. Si on veut retirer un item de la banque d'items, il faut donc supprimer ce module et penser à renuméroter les modules restants pour qu'il n'y ait pas de trou. Si on veut ajouter un item à la banque, il suffit de mettre le module correspondant derrière les autres et le nommer en conséquence.

Un des grands avantages de notre logiciel par rapport aux logiciels qu'on peut trouver sur le marché, comme le MicroCAT™, est sa grande flexibilité quant aux types d'items qu'il permet de gérer. Nous allons présenter les différents types d'items que peut contenir la banque d'items.

5.5.1 Les items à choix multiple

Les items à choix multiples sont les plus faciles à construire. Les modules correspondants comportent trois parties. D'abord, il faut afficher la question. En tant que plate-forme programmatrice multimedia, Quest permet de le faire de nombreuses façons ; on peut tout simplement faire afficher un texte, on peut montrer un graphique, voire même un petit film ou un fichier sonore, et on peut même combiner plusieurs de ces moyens.

Puis, on crée autant de boutons interactifs qu'il y a de réponses prévues et on fait afficher sur chacun de ces boutons le texte d'une de ces réponses. Quest permet alors à l'ordinateur de déterminer sur quel bouton le sujet clique avec la souris.

Finalement, il faut prévoir un branchement sur le module "Good job" dans le cas d'une bonne réponse, respectivement un branchement sur le module "Oops" pour chacune des fausses réponses existantes.

La manière la plus simple de créer un nouvel item à choix multiple est de copier le contenu de l'un des modules à choix multiples déjà existant et de changer simplement le texte (éventuellement ajouter des graphiques ou des boutons supplémentaires).

5.5.2 Les items à réponse libre

La construction d'un item à réponse libre ne diffère pas beaucoup de celle d'un item à choix multiple. On remplace simplement les boutons par une boîte de texte dans laquelle le sujet peut taper librement sa réponse, puis la valider en tapant sur la touche "Entrée". Une telle boîte de texte permet de définir ce qu'en Quest on appelle des événements par exemple l'événement "correct" dans lequel on précise la bonne réponse à la question. Quest offre la possibilité de définir un ou plusieurs mots et même des phrases entières comme étant la bonne solution. On peut aussi définir un nombre comme bonne réponse, ou bien tout un intervalle de réels, si on veut laisser une petite marge d'erreur, par exemple pour des items, où il faut mesurer un segment.

La troisième phase consiste alors à faire un branchement vers le module “Good job”, dans le cas, où la réponse correspond à ce qu’on a défini par l'événement “correct”, respectivement vers le module “Oops”, pour toute autre réponse.

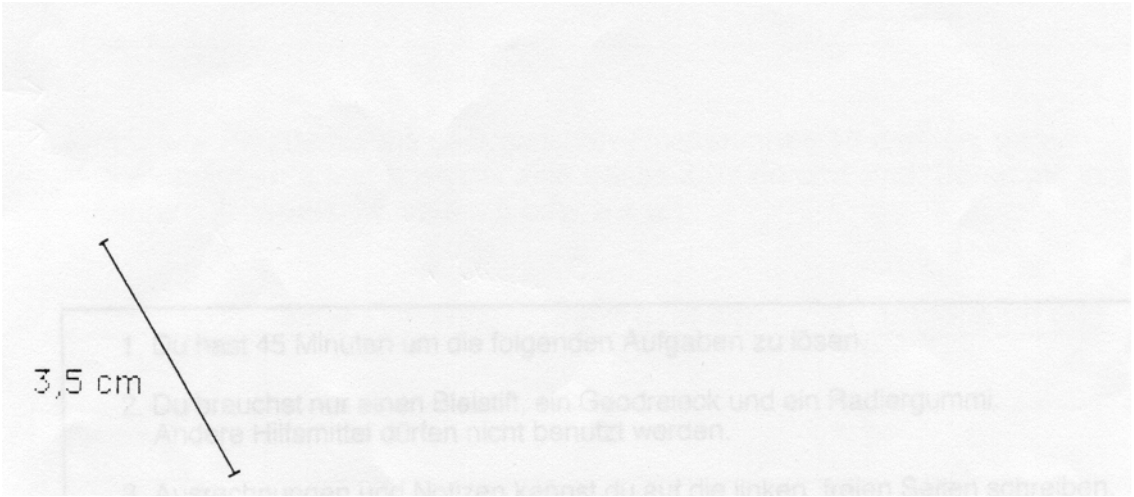
Pour les items à réponse libre, la manière la plus simple de créer un nouvel est également de copier le contenu de l’un des modules à réponse libre déjà existant et de changer simplement le texte.

5.5.3 Les items graphiques

La possibilité de créer des items qui consistent à demander au sujet de compléter un dessin en ajoutant des droites (ou d’autres figures géométriques simples) est un des grands avantages de notre logiciel. De tels items sont néanmoins nettement plus fastidieux à programmer que les items à choix multiple ou à réponse libre. La méthode la plus simple consiste à créer un ou plusieurs points mobiles que le sujet peut déplacer sur son écran en cliquant dessus avec le bouton gauche de sa souris et en déplaçant la souris en gardant le bouton pressé. Pour dessiner une droite par exemple, on a besoin de deux points mobiles que le sujet promène à travers l’écran. L’ordinateur construit alors la droite qui passe à tout instant par ces deux points. Pour montrer comment construire en détail un tel item, nous allons examiner par la suite l’item numéro 33 de notre banque d’item, qui correspond à la question 13 de l’épreuve standardisée en mathématiques de novembre 1966.

Rappelons d’abord la question :

13. Hier ist die Breite eines Rechteckes:



Vervollständige das Rechteck, wenn du weißt, daß seine Länge doppelt so groß wie die Breite ist.

On commence, comme pour tout item à faire afficher la question. Pour cela, nous avons entré le dessin ci-dessus au scanner et nous l'avons sauvegardé comme *bitmap*. Quest permet de traiter facilement de tels fichiers graphiques. De plus, nous avons fait afficher en haut et en bas du dessin les deux textes qui comportent la question.

```
Graphic File (q43.bmp)
Text
Text
```

Afin de permettre au sujet de compléter le rectangle, nous avons besoin quatre fois d'un fichier graphique qui comporte un gros point noir et qu'on utilise comme points extrémités du rectangle à construire.

```
Graphic File (cercle.wmf) "point_1"
Graphic File (cercle.wmf) "point_2"
Graphic File (cercle.wmf) "point_3"
Graphic File (cercle.wmf) "point_4"
```

En fait, les deux extrémités situées aux bouts du côté donné du rectangle sont des points fixes, qui ne sont utilisés que comme extrémités des deux côtés du rectangle qui s'y attachent et qu'on cache pour qu'ils ne sont pas visible sur l'écran. Les deux autres points sont déclarés points mobiles.

```
point_3.Hide();
point_4.Hide();
Drag Animation (point_1) "dp1"
Drag Animation (point_2) "dp2"
```

Le sujet peut les déplacer librement sur l'écran (en fait, on a réduit la zone de déplacement à la partie à droite du côté existant, pour forcer le sujet de construire son rectangle de ce côté-là). L'idée est alors de faire construire par l'ordinateur le rectangle qui comporte le côté donné en avance et dont les deux autres points extrémités sont à chaque instant les deux points mobiles. L'ordinateur doit donc "dessiner" les trois côtés restants. De plus, on veut que le sujet puisse changer son rectangle, jusqu'à ce qu'il en soit satisfait. L'ordinateur doit donc être capable de mettre à jour le rectangle (qui en fait peut très bien être un quadrilatère quelconque, puisque rien n'oblige le sujet de placer ses deux points de façon à ce que deux côtés opposés de sa figure soient parallèles) à chaque fois que le sujet déplace un des deux points mobiles.

Pour effectuer cela, on a besoin a priori de trois segments de droites qui sont à chaque instant placés de façon à ce que, ensemble avec le côté fixe, ils forment un quadrilatère.

Malheureusement en Quest, une fois créée, une droite doit toujours rester penchée du même coté (elle doit rester ou bien tout le temps montante ou bien tout le temps descendante). C'est pourquoi, on a besoin d'avoir chaque segment de droite en deux exemplaires. En tout, il faut donc créer six droites, trois qui montent et trois qui descendent. Pour commencer, on rend invisible l'ensemble des droites.

```
Line "droite1"
Line "droite2"
Line "droite3"
Line "droite4"
Line "droite5"
Line "droite6"
droite1.Hide();
droite2.Hide();
droite3.Hide();
droite4.Hide();
droite5.Hide();
droite6.Hide();
```

De plus, on a besoin de quatre paires de variables qui contiennent les coordonnées des quatre extrémités du quadrilatère. On les a notés (X1, Y1), (X2, Y2), (X3, Y3) respectivement (X4, Y4). Puisque les points 3 et 4 sont des points fixes, les deux dernières paires sont en fait des constantes qu'on peut fixer une fois pour toutes.

```
int X1, Y1, X2, Y2, X3, Y3, X4, Y4;
X3=point_3.GetX();
Y3=point_3.GetY();
X4=point_4.GetX();
Y4=point_4.GetY();
```

Quand l'item est proposé au sujet, celui-ci voit donc sur l'écran le texte avec la question, le dessin qu'on a scanné et qui montre le côté connu du rectangle, ainsi que les deux points mobiles qui se trouvent au début quelque part tout à droite de l'écran. Le sujet déplace maintenant un des deux points, disons le point numéro 1. Dés qu'il relâche le bouton de sa souris, l'ordinateur place les nouvelles coordonnées des points numéro 1 et 2 dans les variables correspondantes. Puis, il doit construire le quadrilatère qui correspond à la situation sur l'écran. Remarquons que le point en haut du côté donné est le point numéro 3 et celui en bas, le point numéro 4. Si le point numéro 1 se trouve en dessus du point numéro 2, il faut alors "dessiner" une droite entre le point numéro 3 et le point numéro 3, une droite entre le point numéro 1 et le point numéro 2, et une droite entre le point numéro 2 et le point numéro 4. Si par contre le sujet a placé le point numéro 1 plus bas que le point numéro 2, il faut

“dessiner“ une droite entre le point numéro 3 et le point numéro 2, une deuxième entre le point numéro 2 et le point numéro 1, et une troisième entre le point numéro 1 et le point numéro 4. Pour construire ces droites, il faut tenir compte du fait, s’il faut utiliser des droites montantes ou des droites descendantes. Si par exemple, dans le premier des cas ci-dessus, le point numéro 1 est placé plus haut que le point numéro 3 et que le point numéro 2 est placé plus bas que le point numéro 4, il faut prendre une droite montante comme droite entre le point numéro 3 et le point numéro 4, une droite descendante entre le point numéro 4 et le point numéro 2. La droite entre le point numéro 1 et le point numéro deux peut être ou bien montante ou bien descendante, cela dépend duquel des deux point se situe le plus à gauche sur l’écran. Il y a en tout 16 situations différentes qui peuvent se présenter, ce qui rend un peu longue l’algorithme nécessaire.

```
Watch for... "dp1" Dropped then...

      X1=point_1.GetX();
Y1=point_1.GetY();
X2=point_2.GetX();
Y2=point_2.GetY();

{if ((Y1<=Y3)&(Y2<=Y4)&(Y1<=Y2)&(X1<=X2))
  { droite1.SetRect(X3+8, Y1+6,X1+8,Y3+6);
    droite1.Show();
    droite3.SetRect(X4+8, Y2+6, X2+8, Y4+6);
    droite3.Show();
    droite2.SetRect(X1+8,Y1+6,X2+8,Y2+6);
    droite2.Show();
  }
}
{if ((Y1<=Y3)&(Y2<=Y4)&(Y1<=Y2)&(X2<=X1))
  { droite1.SetRect(X3+8, Y1+6,X1+8,Y3+6);
    droite1.Show();
    droite3.SetRect(X4+8, Y2+6, X2+8, Y4+6);
    droite3.Show();
    droite5.SetRect(X2+8,Y1+6,X1+8,Y2+6);
    droite5.Show();
  }
}
{if ((Y2<=Y3)&(Y1<=Y4)&(Y2<=Y1)&(X2<=X1))
  { droite1.SetRect(X3+8, Y2+6,X2+8,Y3+6);
    droite1.Show();
    droite3.SetRect(X4+8, Y1+6, X1+8, Y4+6);
    droite3.Show();
    droite2.SetRect(X2+8,Y2+6,X1+8,Y1+6);
    droite2.Show();
  }
}
```

```
{if ((Y2<=Y3)&(Y1<=Y4)&(Y2<=Y1)&(X1<=X2))
  { droite1.SetRect(X3+8, Y2+6,X2+8,Y3+6);
     droite1.Show();
    droite3.SetRect(X4+8, Y1+6, X1+8, Y4+6);
    droite3.Show();
    droite5.SetRect(X1+8,Y2+6,X2+8,Y1+6);
    droite5.Show();
  }
}
{if ((Y1<=Y3)&(Y4<=Y2)&(Y1<=Y2)&(X1<=X2))
  { droite1.SetRect(X3+8, Y1+6,X1+8,Y3+6);
     droite1.Show();
    droite2.SetRect(X4+8, Y4+6, X2+8, Y2+6);
    droite2.Show();
    droite4.SetRect(X1+8,Y1+6,X2+8,Y2+6);
    droite4.Show();
  }
}
{if ((Y1<=Y3)&(Y4<=Y2)&(Y1<=Y2)&(X2<=X1))
  { droite1.SetRect(X3+8, Y1+6,X1+8,Y3+6);
     droite1.Show();
    droite2.SetRect(X4+8, Y4+6, X2+8, Y2+6);
    droite2.Show();
    droite3.SetRect(X2+8,Y1+6,X1+8,Y2+6);
    droite3.Show();
  }
}
{if ((Y2<=Y3)&(Y4<=Y1)&(Y2<=Y1)&(X2<=X1))
  { droite1.SetRect(X3+8, Y2+6,X2+8,Y3+6);
     droite1.Show();
    droite2.SetRect(X4+8, Y4+6, X1+8, Y1+6);
    droite2.Show();
    droite4.SetRect(X2+8,Y2+6,X1+8,Y1+6);
    droite4.Show();
  }
}
{if ((Y2<=Y3)&(Y4<=Y1)&(Y2<=Y1)&(X1<=X2))
  { droite1.SetRect(X3+8, Y2+6,X2+8,Y3+6);
     droite1.Show();
    droite2.SetRect(X4+8, Y4+6, X1+8, Y1+6);
    droite2.Show();
    droite3.SetRect(X1+8,Y2+6,X2+8,Y1+6);
    droite3.Show();
  }
}
{if ((Y3<=Y1)&(Y4<=Y2)&(Y1<=Y2)&(X2<=X1))
  { droite2.SetRect(X3+8, Y3+6,X1+8,Y1+6);
     droite2.Show();
    droite4.SetRect(X4+8, Y4+6, X2+8, Y2+6);
    droite4.Show();
    droite1.SetRect(X2+8,Y1+6,X1+8,Y2+6);
```

```
    droite1.Show();
  }
}
{if ((Y3<=Y1)&(Y4<=Y2)&(Y1<=Y2)&(X1<=X2))
  { droite2.SetRect(X3+8, Y3+6,X1+8,Y1+6);
    droite2.Show();
    droite4.SetRect(X4+8, Y4+6, X2+8, Y2+6);
    droite4.Show();
    droite6.SetRect(X1+8,Y1+6,X2+8,Y2+6);
    droite6.Show();
  }
}
{if ((Y3<=Y2)&(Y4<=Y1)&(Y2<=Y1)&(X1<=X2))
  { droite2.SetRect(X3+8, Y3+6,X2+8,Y2+6);
    droite2.Show();
    droite4.SetRect(X4+8, Y4+6, X1+8, Y1+6);
    droite4.Show();
    droite1.SetRect(X1+8,Y2+6,X2+8,Y1+6);
    droite1.Show();
  }
}
{if ((Y3<=Y2)&(Y4<=Y1)&(Y2<=Y1)&(X2<=X1))
  { droite2.SetRect(X3+8, Y3+6,X2+8,Y2+6);
    droite2.Show();
    droite4.SetRect(X4+8, Y4+6, X1+8, Y1+6);
    droite4.Show();
    droite6.SetRect(X2+8,Y2+6,X1+8,Y1+6);
    droite6.Show();
  }
}

{if ((Y3<=Y1)&(Y2<=Y4)&(Y1<=Y2)&(X2<=X1))
  { droite2.SetRect(X3+8, Y3+6,X1+8,Y1+6);
    droite2.Show();
    droite1.SetRect(X4+8, Y2+6, X2+8, Y4+6);
    droite1.Show();
    droite3.SetRect(X2+8,Y1+6,X1+8,Y2+6);
    droite3.Show();
  }
}
{if ((Y3<=Y1)&(Y2<=Y4)&(Y1<=Y2)&(X1<=X2))
  { droite2.SetRect(X3+8, Y3+6,X1+8,Y1+6);
    droite2.Show();
    droite1.SetRect(X4+8, Y2+6, X2+8, Y4+6);
    droite1.Show();
    droite4.SetRect(X1+8,Y1+6,X2+8,Y2+6);
    droite4.Show();
  }
}
{if ((Y3<=Y2)&(Y1<=Y4)&(Y2<=Y1)&(X1<=X2))
  { droite2.SetRect(X3+8, Y3+6,X2+8,Y2+6);
```

```
      droite2.Show();
      droite1.SetRect(X4+8, Y1+6, X1+8, Y4+6);
      droite1.Show();
      droite3.SetRect(X1+8,Y2+6,X2+8,Y1+6);
      droite3.Show();
    }
  }
  {if ((Y3<=Y2)&(Y1<=Y4)&(Y2<=Y1)&(X2<=X1))
    { droite2.SetRect(X3+8, Y3+6,X2+8,Y2+6);
      droite2.Show();
      droite1.SetRect(X4+8, Y1+6, X1+8, Y4+6);
      droite1.Show();
      droite4.SetRect(X2+8,Y2+6,X1+8,Y1+6);
      droite4.Show();
    }
  }
```

Après, il faut encore répéter les mêmes lignes de programmation pour le cas, où le sujet bouge le point numéro 2, ce qu'on a omis ici, pour ne pas trop allonger ce paragraphe.

Finalement, on a encore chois de rendre invisible l'ensemble des droites, pendant que le sujet est en train de bouger un des deux point, puisque cela évite une certaine confusion sur l'écran.

```
  Watch for... "dp1" Dragging then...
  droite1.Hide();
  droite2.Hide();
  droite3.Hide();
  droite4.Hide();
  droite5.Hide();
  droite6.Hide();

  Watch for... "dp2" Dragging then...
  droite1.Hide();
  droite2.Hide();
  droite3.Hide();
  droite4.Hide();
  droite5.Hide();
  droite6.Hide();
```

Quand le sujet estime qu'il a correctement construit son rectangle, il appuie sur la touche "Echap". L'ordinateur doit alors vérifier s'il a répondu correctement ou non à la question. En fait, on estime que la réponse est correcte, si les points numéro 1 et 2 sont placés dans un voisinage assez proche de l'endroit où ils devraient être en laissant au sujet la liberté de placer ou bien le point numéro 1 ou bien le point numéro 2 en haut. Une réponse correcte entraîne alors, comme pour tout item, un branchement vers le module "Good job", une réponse incorrecte par contre un branchement vers le module "Oops".

Watch for... Key Press (Echap) then...

```
  if (((322<X1)&(X1<332)&(322<X2)&(X2<332)&(196<Y1)&(Y1<206)&(313<Y2)&
  (Y2<323))
  |((322<X2)&(X2<332)&(322<X1)&(X1<332)&(196<Y2)&(Y2<206)&(313<Y1)&
(Y1<323)))
   {strcpy (fname, "Good job");
  BranchToFrame(fname);}
  else
  {strcpy (fname, "Oops");
  BranchToFrame(fname);}
```

5.6 Installation et mise en marche du logiciel

Notre logiciel nécessite un PC avec Windows 95 ou 98. De plus, il faut que la plate-forme de programmation Quest Net+ for Windows™ de Allen Communication soit installé dessus.

On a besoin des trois disquette intitulées “Test adaptatif informatisé en mathématiques”, volume 1 à 3, qui contiennent les fichiers “cat.arj”, “cat.a01”, respectivement “cat.a02” et “arj.exe“. Les trois premiers contiennent notre programme sous forme comprimée, le quatrième est le logiciel nécessaire pour les décompresser. I l faut alors copier le contenu de ces disquettes dans le répertoire principal de Quest (généralement c’est le répertoire C:\QUEST). Cela étant fait, on passe sous MS-DOS et on décompresse les fichiers.

(Pour copier les fichiers sous MS-DOS, on se place dans le répertoire principal du disque dur, en tapant “cd ..”. Puis, on passe dans le répertoire principal de Quest en tapant “cd quest“. Ensuite, on met une à une les trois disquettes fournies dans le lecteur et on tape “copy a : *.*”, afin de copier leur contenu dans le répertoire principal de Quest.)

On décompresse alors les fichiers en tapant “arj x –vaa cat.arj”. L’ordinateur pose les quatre questions suivantes auxquelles il faut à chaque fois répondre par “y” :

TEST\TEST.QT, Create this directory ? [YNAQ]
TEST\GRAPHICS\CERCLE.WMF, Create this directory ? [YNAQ]
OK to process next volume / diskette (1) ? [YNAQ]
OK to process next volume / diskette (2) ? [YNAQ].

Maintenant le répertoire QUEST contient un sous-répertoire TEST, dans lequel se trouve notre logiciel. Il est composé des deux fichiers principaux “test.qt” et “test0000.qm”, ainsi que d’un sous-répertoire GRAPHICS dans lequel sont placés tous les fichiers graphiques que les différents items utilisent. Avec cela, le logiciel est prêt à l’emploi. On peut maintenant effacer les quatre fichiers qu’on a copié des disquettes fournies.

Pour lancer le test adaptatif, il faut d'abord lancer le logiciel Quest Net+, puis lancer l'application "test.qt" comme n'importe quelle application de Quest. Cela se fait en appuyant simultanément sur les touches "Ctrl" et "O", puis en doublecliquant avec le bouton gauche de la souris dans le menu "ouverture" sur le sous-répertoire "test", ensuite sur le fichier "test.qt".

5.7 Conclusion

Le but de notre projet, construire un test adaptatif informatisé pouvant servir à l'évaluation sommative et formative des acquisitions scolaires en mathématiques et contribuer efficacement à l'orientation des élèves, en se basant sur l'analyse et le traitement des données recueillies au cours des années lors des épreuves standardisées en mathématiques effectuées lors du passage primaire post-primaire, a été atteint.

Nous avons conçu un logiciel qui pourra de plus servir comme prototype permettant de développer d'autres tests adaptatifs administrés par ordinateur. Comme le souligne R. Martin (Martin 1998), c'est grâce à cette perspective de travail que les épreuves standardisées peuvent devenir quelque chose de plus qu'un simple instrument applicatif et qu'elles peuvent engendrer un instrument de recherche très puissant. En effet, on peut arriver avec une telle méthodologie à la construction d'épreuves flexibles, fiables, adaptées au niveau de compétence des élèves et néanmoins comparables. Les élèves pourront en profiter pour documenter leurs progrès d'apprentissage d'une manière assez informelle, mais néanmoins fiable et les enseignants pourront utiliser ces évaluations, afin de pouvoir mieux guider les élèves dans leur processus d'apprentissage.

Une condition nécessaire pour cela sera de disposer d'une banque d'item beaucoup plus importante. Cela pourrait être acquise en incorporant dans les futurs tests standardisés en mathématiques à chaque fois quelques items de la banque de données. Ceci impliquerait que les scores des futurs test pourraient être mises sur la même échelle que celle utilisé pour notre programme et deviendraient ainsi comparables. De cette façon on disposerait d'une banque d'items qui grandirait d'année en année ce qui faciliterait également l'analyse de la structure latente des compétences entrant en action lors de la résolution des épreuves en mathématiques.

Nous n'avons en effet pas réussi à trouver une telle taxonomie qui s'applique aux résultats de l'épreuve standardisée qui a servi de base à notre travail. Nous avons en effet essayé d'extraire une structure par des méthodes d'échelonnement multi-dimensionnel, mais ni la

structuration utilisée au Luxembourg qui comporte une dimension à quatre facettes (les opérations, les fractions, le système décimal et les grandeurs, ainsi que la géométrie), ainsi qu'une deuxième dimension à trois facettes, la reproduction (exercices qui demandent à l'élève de reproduire une technique, une connaissance ou un raisonnement vus en classe), la réorganisation (exercices qui demandent à l 'élève d'adapter ses connaissances et savoir-faire à des exercices similaires), et le transfert (exercices qui demandent à l'élève de transférer ce qu'il a appris à des situations nouvelles et/ou d'élaborer un raisonnement), ni la structure établie au Boston College lors d'une grande étude internationale appelée Third International Mathematics and Science Study et qui comporte à son tour deux dimensions (la première composée de 6 facettes, à savoir, les nombres entiers (*whole numbers*), les fractions (*fractions*), la mesure (*measurement*), la représentation des données (*data representation*), la géométrie (*geometry*) et l'étude des structures (*patterns*) et la deuxième composée de 4 facettes, les connaissances acquises (*knowing*), la reproduction de raisonnements vus en classe (*routine procedure*), la réorganisation (*complex procedures*), et la solution de problèmes (*solving problems*)), n'étaient adaptables à nos données. Ceci n'est pas très surprenant puisque le nombre faible d'items dont nous avons disposé a impliqué que de nombreuses classes d'items sont restés vides où ne contiennent qu'un seul élément, ce qui ne suffit bien sûr pas pour une analyse.

L'élaboration de cette structure latente des compétences en mathématiques reste donc un problème ouvert, intéressant pour une recherche future.

Annexe

L'EPREUVE STANDARDISEE EN MATHEMATIQUES DE NOVEMBRE 1966

Les élèves ont effectué l'épreuve en deux parties. Pour chaque partie ils ont eu les indications suivantes pour répondre aux questions :

1. Du hast 45 Minuten um die folgenden Aufgaben zu lösen.

2. Du brauchst nur einen Bleistift, ein Geodreieck und ein Radiergummi.
   Andere Hilfsmittel dürfen nicht benutzt werden.

3. Ausrechnungen und Notitzen kannst du auf die linken, freien Seiten schreiben.

4. Schreibe bitte sehr leserlich.

5. Wenn du eine Aufgabe nicht lösen kannst, so halte dich nicht zu lange dabei auf, sondern lasse sie aus und versuche die folgenden Aufgaben zu lösen.

6. Halte dich genau an die Anweisungen. Zum Beispiel:
   Beim Kopfrechnen dürfen keine Tafelrechnungen gemacht werden.

Ci-dessous nous rappelons les 24 questions de l'épreuve.

1. Berechne im Kopf.

a) 51+37+53+149=

b) 53+527=

c) 90:5=

d) 1999-430=

e) 10·7·0·20=

f) 400·12=

g) (8+40):4=

h) 21+9:3=

i) 609+……=700

j) ……-429=2999

……...

2. Mache als Tafelrechnung am freigelassenen Platz.

a) **17321-2735-87**	b) **27·483**	c) **5208:14**

……...

3. Mache einen Überschlag! Kreuze jeweils die Zahl an, welche dem Resultat der angeschriebenen Rechnung am nächsten kommt.

a) **72·31**

O 210 O 2100 O 750 O 21000 O Keine Ahnung

b) **8023:98**

O 8000 O 80 O 8 O 800 O Keine Ahnung

……...

4. Setze das richtige Zeichen (>, <, =).

a)	1,91……1,191	c)	$\frac{7}{12}$ …… $\frac{7}{11}$	e)	$\frac{7}{12}$ …… $\frac{5}{8}$	g)	$\frac{3}{4}$ h……45min
b)	0,01……0,010	d)	$\frac{7}{12}$ …… $\frac{49}{100}$	f)	$\frac{70}{120}$ …… $\frac{10}{16}$	h)	50g…… $\frac{1}{2}$ kg

……...

5. Schreibe folgende Brüche als Dezimalzahl.

a)		b)		c)		d)	
Bruch	Dezimalzahl	Bruch	Dezimalzahl	Bruch	Dezimalzahl	Bruch	Dezimalzahl
$\frac{1}{2}$ =	……..	$\frac{3}{4}$ =	……...	$\frac{1}{20}$ =	……..	$\frac{2}{5}$ =	…….

……..……

6. Ergänze.

a)	2 m = …… dm	c)	3,2 kg = …… g	e)	2500 kg = 2,5 ……
b)	3,2 …… = 3200 m	d)	1 min 12 s = …… s	f)	3 cm 2 mm = …… mm

…….

7. Hier sind fünf Zahlen:

22738 76372 21670 13655 35742

a) Welche von ihnen ist teilbar durch 4? ……

b) Welche von ihnen ist eilbar durch 3 ?……

c) Welche von ihnen ist gleichzeitig teilbar durch 2 **<u>und</u>** durch 5 ? ……

…….

8. Die Zahl 6352 ist nicht teilbar durch 9. Wieviel muβ man mindestens zu der Zahl 6352 hinzuzählen damit sie durch 9 teilbar wird?

Deine Antwort: ……

9. In der Samenhandlung hängt folgende Preisliste aus:

Kunstdünger pro Tüte: 275 F
25 Erdbeerpflanzen: 200 F
20 g Gemüsesamen: 35 F
25 Kohlpflanzen: 87 F
Zwiebeln pro Kilogramm: 80 F

Der Lehrling hat für einen Kunden folgende Rechnung aufgestellt. Vervollständige sie!

Rechnung Nr: 3587 vom 10. April 1996 für Frau Irene Müller

	Artikel		**Betrag**
a)	3 Tüten Kunstdünger	3·275=	
b)	0,5 kg Zwiebeln		
c)	100 Erdbeerpflanzen		
d)	80 g Blumensamen		
e)			**Total:** F

……...

10. Hier sind zwei Brüche : $\frac{4}{5}$ und $\frac{5}{4}$

a) Welcher der beiden Brüche ist größer als 1 ? ……

b) Welcher der beiden Brüche liegt auf dem Zahlenstrahl am nächsten bei 1 ? ……

……...

11. Hier sind zwei Zahlenfolgen. Suche jeweils die folgende Zahl.

a) 569 548 527 506 ……

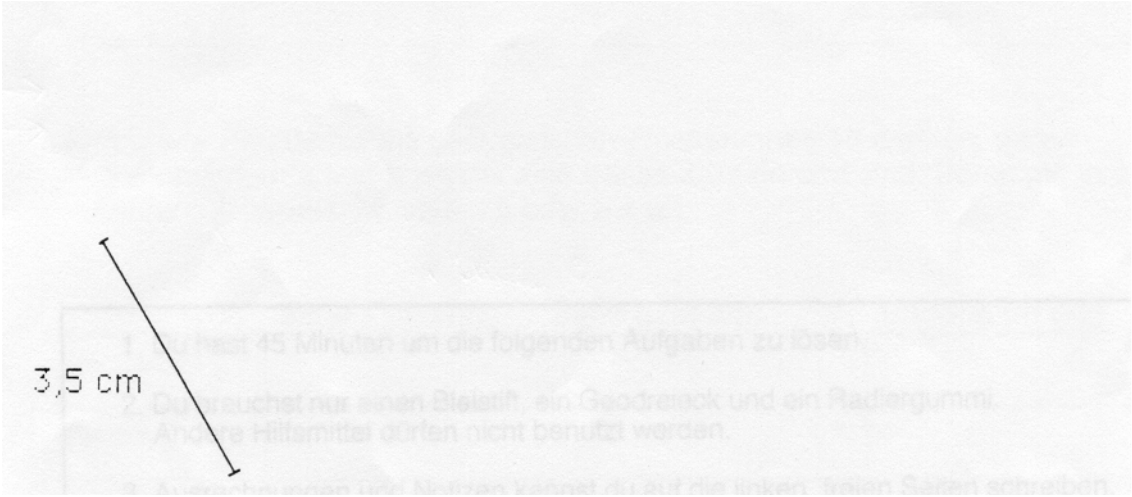
b) 14 8 16 10 20 14 28 ……

12. Sind die folgenden Sätze wahr oder falsch? Kreuze an!

a)	Alle Primzahlen sind ungerade	O wahr	O falsch
b)	57 ist eine Primzahl	O wahr	O falsch
c)	Das Produkt von zwei Primzahlen ist nie eine Primzahl	O wahr	O falsch
d)	Die Diagonalen eines Rechteckes stehen senkrecht zueinander	O wahr	O falsch
e)	Die gegenüberliegenden Seiten eines Rechtecks sind parallel	O wahr	O falsch
f)	Ein Würfel hat 6 quadratische Seitenflächen	O wahr	O falsch

……...

13. Hier ist die Breite eines Rechteckes:



Vervollständige das Rechteck, wenn du weißt, daß seine Länge doppelt so groß wie die Breite ist.

14. Wenn du weißt, daß **72 · 28 = 2016** ergibt, ergänze folgende Rechenaufgaben, ohne zusätzliche schriftliche Berechnungen.

a) **36 · 28** = …… b) **144 · 14** =……

…….

15. Du darfst die Ziffern 7 2 0 9 und das Komma jeweils einmal benutzen.
Welches ist die kleinste Dezimalzahl, die du mit diesen Zeichen bilden kannst?

Deine Antwort: ……

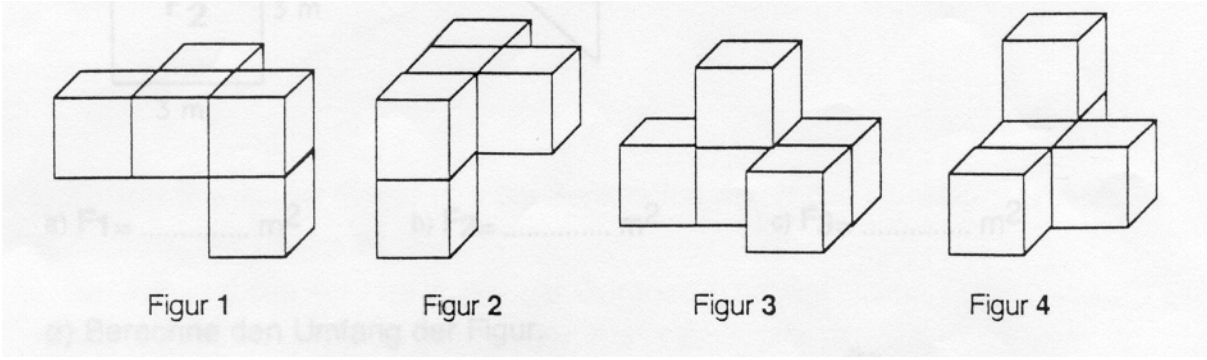
…….

16. Gib alle Rechtecke mit dem gleichen Flächeninhalt 16 cm^2 an, deren Seitenlängen a und b jeweils eine ganze Zahl (in cm) sind. Berechne ihren Umfang.
(Hinweis: Nimm a > b oder a = b.)

		Seite a (in cm)	Seite b (in cm)		Umfang (in cm)
Rechteck 1	a)			d)	
Rechteck 2	b)			e)	
Rechteck 3	c)			f)	

…….

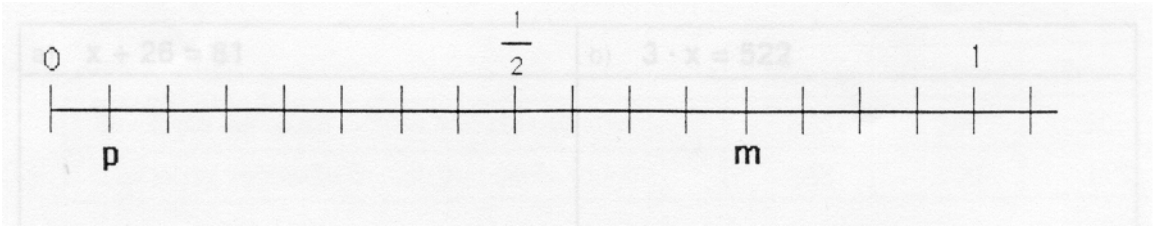
17. Hier sind jedesmal 5 Würfel aus verschiedenen Blickwinkeln gesehen. Welche Figur unterscheidet sich von den anderen?



Umkreise die richtige Antwort.

Keine Figur **1** Figur **2** Figur **3** Figur **4**

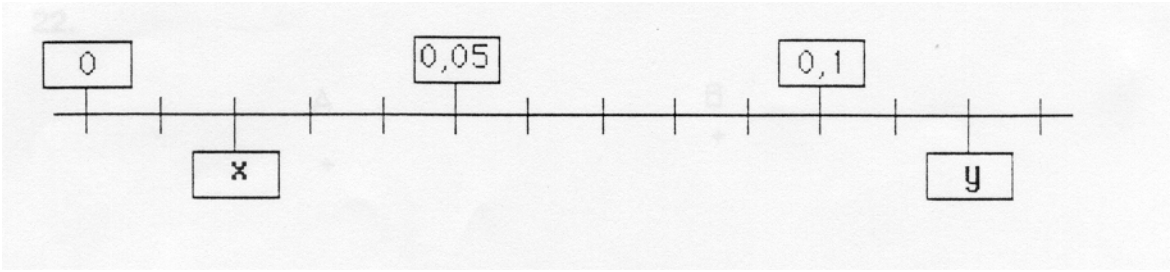
18. Bestimme die Brüche, welche durch die Buchstaben m und p auf dem Zahlenstrahl dargestellt sind.



a) **m** = …… b) **p** = ……

……

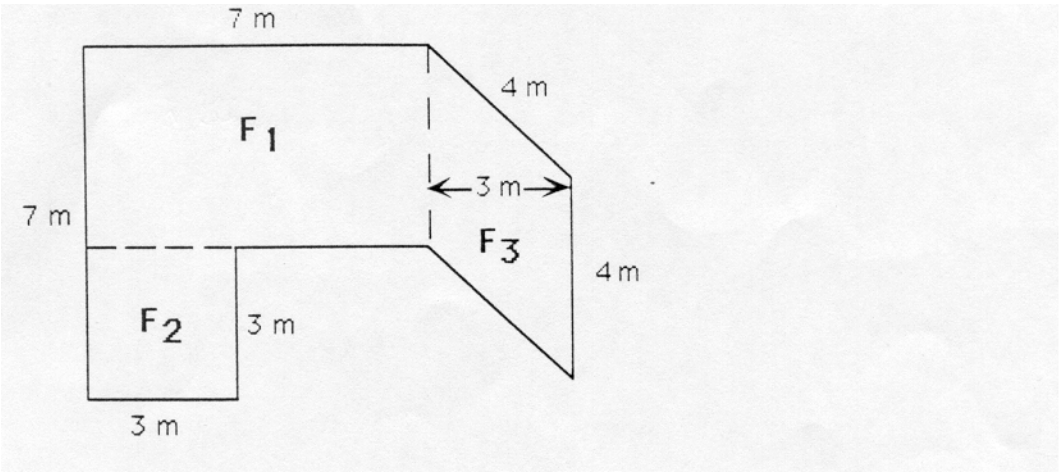
19. Bestimme die Dezimalzahlen, welche durch die Buchstaben x und y auf dem Zahlenstrahl dargestellt sind?



a) **x** = …… b) **y** = ……

……

20. Berechne die Flächen F_1, F_2 und F_3.



a) F_1 = ... m^2 b) F_2 = ... m^2 c) F_3 = ... m^2

d) Berechne den Umfang der Figur

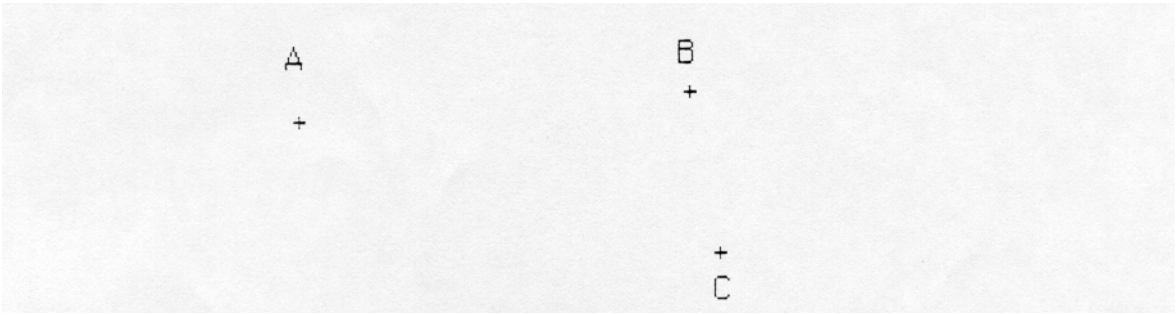
Der Umfang der Figur beträgt …… m.

21. Löse folgende Gleichungen in den vorgesehenen Kästen.

a) **x + 26 = 81**	b) **3 · x = 522**

……

22.



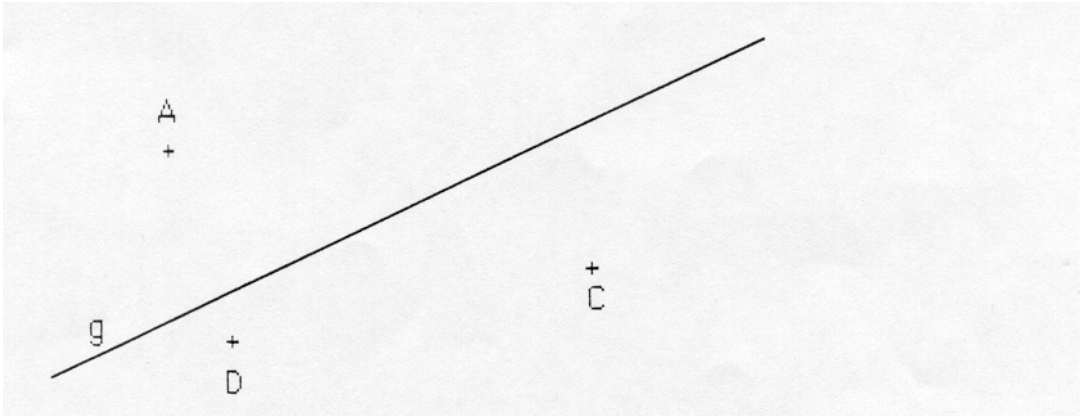
a) Zeichne die Strecke [AB].

b) Miß und stelle fest: Die Strecke [AB] ist …… cm lang.

c) Zeichne die Halbgerade [CA)

……

23.



a) Konstruire zu der Geraden **g** die Senkrechte durch den Punkt **C**.

b) Zeichne, miß und stelle fest:

Der Abstand des Punktes **A** von der Geraden **g** beträgt …… cm.

c) Zeichne die Parallele zu **g** durch den Punkt **D**.

24. Diese Eintrittsscheine wurden während drei Tagen an der Kinokasse verkauft:

	Montag	Dienstag	Mittwoch
Der erste verkaufte Schein trug die Nummer	120	387	501
Der letzte verkaufte Schein trug die Nummer	386	500	713

a) Wieviele Eintrittsscheine wurden am Montag verkauft? ……

b) Wieviele der Eintrittsscheine, die am Dienstag verkauft wurden, trugen wenigstens eine Ziffer 4? ……

……...