

REVUE DE STATISTIQUE APPLIQUÉE

F. YATES

M. J. R. HEALY

S. LIPTON

L'analyse des expériences répétées sur calculatrice électronique

Revue de statistique appliquée, tome 7, n° 2 (1959), p. 47-75

http://www.numdam.org/item?id=RSA_1959__7_2_47_0

© Société française de statistique, 1959, tous droits réservés.

L'accès aux archives de la revue « Revue de statistique appliquée » (<http://www.sfds.asso.fr/publicat/rsa.htm>) implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques
<http://www.numdam.org/>

L'ANALYSE DES EXPÉRIENCES RÉPÉTÉES SUR CALCULATRICE ÉLECTRONIQUE

par F. YATES, M. J. R. HEALY et S. LIPTON

Parmi les techniques de l'analyse statistique, les corrélations et l'analyse de la variance constituent des méthodes très intéressantes, mais elles imposent à l'analyste un volume important de calculs fastidieux et sujets à erreurs.

L'emploi d'une calculatrice électronique lève cette lourde hypothèque et donne à ces types de travaux un caractère routinier et courant, dès que des programmes appropriés ont été rédigés. C'est ce que nous démontre pour un cas concret l'exposé qui suit, qui est une communication de MM. Yates, Healy et Lipton, de la station expérimentale anglaise d'Expérimentation agricole et biologique de Rothamsted, communication présentée à la Section Recherches de la British Royal Statistical Society le 1er mai 1957. Elle a paru dans le volume 19, n°2, 1957 - Série B du Journal de cette Association.

La traduction en a été assurée par M. Pepe, Administrateur à l'I.N.S.E.E.

La publication de cette traduction est faite avec l'accord de la British Royal Statistical Society, que nous remercions d'avoir bien voulu permettre cette présentation aux lecteurs français.

RESUME

Cette communication décrit les méthodes qui ont été développées pour l'analyse des expérimentations répétées sur la calculatrice électronique de Rothamsted. Leur principe général est de faire traiter à la machine l'ensemble de l'analyse, y compris le calcul des erreurs-type et des résidus, les résultats étant donnés sous une forme appropriée à sa transmission immédiate à l'expérimentateur. L'étude des parcelles manquantes et les calculs de covariance peuvent également être traités. Des calculs préliminaires sur les données brutes sont exécutés par la machine, en utilisant un programme qui est commun à tous les types de plans. Les données sont perforées sur bande de papier, une variable aléatoire à la fois, dans l'ordre où elles sont reçues de l'expérimentateur, c'est à dire sans suppression de caractère aléatoire.

Des programmes sont maintenant rédigés pour les types de plans les plus courants, tels que les blocs avec répétition au hasard des traitements, les carrés latins, les plans 3^3 (répétition unique) et 2^n . Comme exemple, un des programmes les plus complexes, celui pour les plans 2^n est décrit en détail. Ce programme

traite tous les types de plan 2ⁿ contenant jusqu'à 128 parcelles, y compris la "confusion" (1), l'estimation des erreurs provenant d'interactions d'ordre supérieur, les répétitions fractionnaires, les facteurs à 4 et 8 niveaux, et les carrés quasi-latins 8 x 8.

On donne un bref exposé du mode de fonctionnement des calculatrices électroniques.

I - INTRODUCTION

En 1954, nous avons eu la bonne fortune d'être dotés d'une calculatrice électronique au Service Statistique de Rothamsted. C'est la calculatrice Elliot 401 N.R. D.C. qui est un modèle prototype construit par Elliot Bros, comme partie du plan de développement de la National Research Development Corporation. Elle nous fut attribuée par l'Agricultural Research Council et par la National Research Development Corporation pour développer l'utilisation des calculatrices électroniques dans les recherches statistiques, particulièrement dans le domaine agricole et biologique.

On peut, en gros, diviser nos calculs statistiques en trois types :

1/ calculs répétitifs simples, qui doivent être exécutés sur une grande quantité de données. Des exemples sont fournis par le calcul, pour chacune des nombreuses unités, de fonctions simples, tels que rapports, produits, transformation en racines carrées, en logarithmes, etc., et la formation de totaux, de moyennes, de sommes de carrés et de produits.

2/ Processus analytiques qui impliquent une longue suite d'opérations, avec seulement des calculs relativement simples à chaque stade, tels que l'analyse d'expériences répétées, et l'ajustement des lignes de probits (2).

3/ Processus analytiques qui nécessitent un lourd travail numérique, tels que l'inversion des matrices, le calcul de valeurs et de vecteurs propres, la résolution des équations différentielles et autres se présentant en génétique et en biologie générale.

Dans leur phase initiale de développement, les calculatrices électroniques furent surtout utilisées pour des problèmes du type 3. Avec le développement de capacités de mémoire plus grandes et avec l'amélioration des dispositifs d'entrée et de sortie, leur emploi dans des problèmes des types 1 et 2 est devenu de plus en plus praticable.

Nous avons jusqu'ici utilisé notre calculatrice pour les classes principales suivantes de problèmes statistiques (des références à des descriptions déjà publiées de travaux dans les diverses classes autres que la première, sont données entre parenthèses, des exemples d'applications à des problèmes de plusieurs de ces classes sont également donnés par Lipton, 1957).

(1) Traduction du mot anglais "confounding", par lequel, dans des expériences factorielles, on limite la taille des blocs en sacrifiant une partie des comparaisons indépendantes correspondant aux interactions d'ordre supérieur.

(2) Note du traducteur : Lignes d'estimation de nuages de points, obtenues en portant en abscisse les logarithmes des doses de stimulus, et en ordonnée les variables normales réduites (augmentées de 5) correspondant aux proportions de sujets ayant réagi à chaque dose.

- 1/ Analyse des expériences répétées.
- 2/ Analyse de probits.
- 3/ Recherches sur l'échantillonnage et les erreurs expérimentales (Church et Lipton, 1956).
- 4/ Travaux à variables aléatoires multiples (Ashton, Healy et Lipton, 1957).
- 5/ Problèmes de génétique.
- 6/ Problèmes d'entomologie.
- 7/ Adaptation de modèles mathématiques à des systèmes biologiques (Patterson, 1956).
- 8/ Analyse d'enquêtes.
- 9/ Construction de tables statistiques (Fisher et Healy 1956 ; Fisher et Yates, 1957 ; Foster et Rees, 1957).
- 10/ Recherches expérimentales par échantillon aléatoire (méthodes de Monte Carlo) dans les problèmes de statistique mathématique.

La calculatrice a en outre été utilisée à la résolution de divers problèmes dans d'autres domaines (non statistiques). (Le Couteur et Lipton, 1955 ; et Shofield et Rees, 1956).

Dans la présente communication, nous nous proposons de décrire les méthodes qui ont été développées pour l'analyse courante d'expériences répétées. Comme indiqué précédemment, c'est seulement l'un des multiples emplois que nous avons trouvés pour notre calculatrice et, en lui-même, il ne serait pas suffisant pour justifier l'installation d'une calculatrice électronique, car de telles analyses peuvent être, et sont, très efficacement exécutées sur des machines de bureau. Toutefois, quand on dispose d'une calculatrice électronique et qu'il y a une lourde charge d'analyse expérimentale, notre expérience a montré qu'on pouvait l'employer très efficacement à cette fin. Le sujet est en outre d'intérêt général, en ce qu'il illustre la manière dont on peut instruire la calculatrice électronique à traiter des suites d'opérations longues et compliquées.

Nous disposons maintenant de programmes pour les plans suivants (1) :

- 1/ Blocs avec répartition au hasard des traitements, jusqu'à 126 parcelles (Healy).
- 2/ Carrés latins jusqu'à 10 x 10 (Lipton).
- 3/ Blocs avec répartition au hasard, avec parcelles subdivisées, jusqu'à 128 parcelles (Gower).
- 4/ Blocs avec répartition au hasard, avec un système p x q de traitements factoriels, jusqu'à 128 parcelles (Gower).
- 5/ Plans factoriels 3³, répétition unique (Yates et Lipton).
- 6/ Plans factoriels 2ⁿ, jusqu'à 128 parcelles, y compris les carrés quasi-latins 8 x 8 (Yates).

Ces programmes correspondent à environ 90 % de toutes les expériences de courte durée parvenant au Service, et d'autres sont en préparation. Les programmes ont été rédigés pendant une période de deux ans et demi, et plusieurs différences de détail peuvent être imputées aux programmeurs, qui s'instruisent par leur propre expérience et par celle des autres ; un des buts de cette communication est de relever les leçons de cette période. Comme résultat pratique, il faut mentionner que le nombre des analyses sur machine (en comptant chaque variable aléatoire analysée comme unité) a été de 834 en 1955, et de 1 701 en

(1) Au cours de cette communication, les noms des chercheurs qui sont les responsables principaux des divers programmes sont mentionnés entre parenthèses.

1956. Concurrément la période moyenne entre réception des données d'une expérience et achèvement de l'analyse a été considérablement réduite.

II - REMARQUES GENERALES SUR LES CALCULATRICES ELECTRONIQUES.

Il peut être utile de donner à ceux qui ne sont pas familiarisés avec les calculatrices électroniques une brève explication de leur fonctionnement.

On croit souvent que la caractéristique principale de ces calculatrices est leur vitesse. Ce n'est qu'une partie de l'histoire. Toute aussi importante est leur aptitude à effectuer des suites d'opérations longues et compliquées sans l'intervention d'un opérateur humain ; en fait, sans cette caractéristique, leur vitesse n'aurait que peu d'intérêt. Il est donc possible d'exécuter l'ensemble d'une analyse statistique compliquée sur la machine "sans que la main humaine n'y touche".

Les caractéristiques essentielles d'une calculatrice électronique sont :

1/ Un organe arithmétique avec des registres associés (dont chacun contient un nombre), qui est capable d'exécuter des opérations arithmétiques simples, telles que l'addition, la soustraction et la multiplication.

2/ Une mémoire, dans laquelle on peut ranger des nombres, s'y reporter, et les remplacer par d'autres nombres selon besoins. Les emplacements de la mémoire sont qualifiés d'"adresses".

3/ Un organe de commande qui fait exécuter à la machine les opérations requises. Les opérations réellement exécutées sont déterminées par ce qu'on appelle des instructions, qui sont elles-mêmes des nombres rangés en mémoire (1). Après exécution de chaque instruction, la machine se reporte automatiquement à l'instruction suivante.

4/ Des dispositifs d'entrée et de sortie qui permettent de lire dans la machine les nombres (données, instructions, etc.) et de perforer les résultats.

Pour accélérer le fonctionnement de la machine sans un appareillage extrêmement coûteux et compliqué, la mise en mémoire s'effectue souvent à deux ou plusieurs "niveaux", dont la facilité et la vitesse d'accès varient.

Dans plusieurs machines, dont la 401, on utilise partout la base de numération binaire, les nombres étant convertis en binaire et reconvertis en décimal par la machine elle-même pendant l'entrée et la sortie. Dans les machines les plus simples, en outre, la calculatrice fonctionne "en série", c'est à dire que les nombres sont représentés par des trains d'impulsions pendant les opérations arithmétiques.

Chaque type de machine a son propre code d'instructions. Elles varient beaucoup en complexité, et, pour programmer (c'est à dire rédiger les instructions) d'une machine particulière, il faut d'abord que le programmeur se familiarise avec le code des instructions et les règles de fonctionnement associées.

Une caractéristique importante de tous les systèmes d'instructions est la possibilité de choix offerte par la discrimination. Dans une discrimination, ou

(1) Dans la terminologie des calculatrices, on emploie d'une façon générale le terme "mot" pour qualifier un nombre contenu dans la machine selon un code approprié (habituellement binaire). Un mot peut représenter un nombre réel, une instruction (ou, dans certaines machines, une paire d'instructions), où l'on peut utiliser ses chiffres séparément ou par groupes comme indicatifs ou à d'autres fins.

instruction de test, la machine teste le contenu d'un certain registre, pour voir s'il contient, par exemple, zéro. Selon le résultat du test, elle choisit ensuite l'une ou l'autre des séquences d'instructions prévues dans le programme.

Une autre caractéristique importante qu'on trouve dans plusieurs machines est ce qu'on appelle la "modification B" des instructions. Elle permet l'addition temporaire du contenu d'un certain registre à l'instruction (qui, comme nous le mentionnons ci-dessus, est en fait un nombre) avant qu'elle soit exécutée.

Les instructions de discrimination ont plusieurs fonctions. Premièrement, la machine peut exécuter des opérations différentes selon des éventualités différentes. Elle peut aussi exécuter un calcul par approximations successives, en testant à chaque stade si un degré suffisant d'exactitude est obtenu. En outre, elle peut exécuter une opération un nombre de fois spécifié en comptant et en testant à chaque cycle si le nombre requis a été atteint. Dans le dernier cas, on ne demande habituellement pas à la machine d'exécuter la même opération à chaque cycle, et c'est là que la modification B présente un intérêt particulier. Ainsi, par l'exemple, si nous désirons additionner n nombres placés dans la mémoire à n adresses consécutives, h à $h + n - 1$, il faut prendre chaque nombre tour à tour et l'additionner au total déjà accumulé. Cela signifie que l'adresse du nombre à prendre doit être modifiée d'une unité à chaque cycle, et que l'opération doit être arrêtée quand n cycles ont été exécutés. La séquence réelle des opérations nécessaires pour exécuter ce travail est, en gros, la suivante :

Soit A le registre dans lequel le total est accumulé, et C le registre exécutant le comptage. Appelons (x) le contenu de l'adresse x de la mémoire.

- 1/ Mettre A à zéro.
- 2/ Mettre n dans C.
- 3/ Tester si (C) est zéro ; s'il est nul, exécuter la première instruction de la séquence suivante ; s'il n'est pas nul, passer à 4
- 4/ Soustraire 1 de C.
- 5/ Additionner $(h + (C))$ à A ; revenir à 3.

Si l'on dispose de la modification B, la modification de l'adresse nécessaire dans l'opération 5 à chaque cycle peut s'effectuer par l'addition temporaire de (C) à la partie de l'instruction "Additionner (h) à A", qui contient l'adresse h.

Cette séquence d'opérations ne peut s'exécuter de la manière fixée ci-dessus que s'il est possible de soustraire une unité directement d'un registre de comptage et de tester ce registre. Dans plusieurs des machines les plus simples, comme la 401, toutes les opérations d'arithmétique et de test sont exécutées dans un registre unique (habituellement appelé le registre accumulateur). Dans ce cas, il faut utiliser un autre registre pour accumuler le total. Le contenu de ce registre et celui du registre utilisé pour le comptage sont transférés dans l'accumulateur et y sont repris alternativement.

Comme on le voit dans la description précédente, pour exécuter un type particulier d'analyse, il faut préparer une séquence complète d'instructions spécifiant tous les stades nécessaires et tenant compte de toutes les éventualités qui peuvent se produire. On appelle programme une telle séquence d'instructions. La préparation des programmes pour les types plus complexes d'analyse requiert un temps et une habileté considérable, mais, une fois que les programmes nécessaires ont été rédigés et soigneusement testés, la machine peut être appliquée à l'exécution de n'importe quelle analyse particulière par simple lecture du programme approprié.

Comme seules les opérations les plus simples peuvent être effectuées directement par les calculatrices électroniques - par exemple, la 401, comme de

nombreuses autres machines, ne peut exécuter la division directement - il est essentiel que des séries standard d'instructions soient rédigées pour exécuter toutes les opérations couramment employées. De telles séries standard d'instructions sont appelées sous-programmes. Elles sont rédigées de façon que la machine puisse recevoir l'ordre d'entrer dans le sous-programme en tout point requis du programme, et de revenir au programme principal quand l'opération de sous-programme a été exécutée. Une librairie appropriée de sous-programmes est essentielle pour la programmation des opérations plus complexes.

On a déjà publié une courte description de la calculatrice Elliott 401 (Lipton, 1955). Au point de vue du programmeur, les caractéristiques essentielles sont les opérations arithmétiques, qui comprennent l'addition, la soustraction, la multiplication, les décalages vers la droite et vers la gauche, la non-équivalence et la comparaison ; les tests zéro et négatifs de l'accumulateur ; la mémoire qui contient 2 944 emplacements de 32 chiffres binaires chacun ; trois registres à accès immédiat qu'on peut utiliser pour la modification B ; et un autre registre qu'on peut combiner avec l'accumulateur pour travailler avec des mots de longueur double. L'entrée dans la machine se fait par bande de papier perforée (on a récemment installé l'entrée par cartes perforées) ; la sortie se fait sur une perforatrice de bande à grande vitesse, ou autrement sur une machine à écrire électrique. La mémoire est un disque rotatif à revêtement magnétique, portant 23 "pistes" de 128 emplacements chacun. Une de ces pistes contient l'information permanente, comprenant divers sous-programmes d'entrée et un sous-programme de division. Parmi les autres pistes, seize sont appelées pistes à relais et sont toutes référencées piste 7 (individuellement piste 7/0, 7/1, . . . , 7/15) ; une seule de ces pistes est disponible à tout moment donné, une instruction machine étant nécessaire pour la commutation à une piste différente au moyen d'une série de relais à grande vitesse. Cette caractéristique, qui ressemble à l'emploi d'une mémoire "à retour" dans certaines autres machines, a eu une influence considérable sur la conception de nos programmes. La machine est raisonnablement rapide : l'addition et la multiplication demandant respectivement 0, 1 et 3, 1 millisecondes, mais une unité de temps plus pratique est le temps de révolution du disque, 12, 8 millisecondes, qui représente le temps maximum d'accès à un nombre ou à une instruction. La commutation d'une piste à relais à une autre prend environ 10 millisecondes.

III - LE PROBLEME GENERAL DE L'ANALYSE DES EXPERIENCES REPETEES.

L'analyse d'au moins les types les plus simples de plans expérimentaux est un travail assez direct, et l'on a suggéré diverses techniques générales par lesquelles on peut rédiger un programme unique pour analyser les données de plusieurs types différents de plans (Tocher, 1952 ; Hartley, 1956). Toutefois, on ne se rend pas compte couramment que le processus complet, depuis le stade auquel les chiffres arrivent de l'expérimentateur jusqu'au stade auquel les résultats analysés lui reviennent, représente presque autant de "travail de bureau" que de calcul au sens étroit. Un de nos principaux objectifs a été de faire traiter à la calculatrice le plus possible de ces processus, y compris la présentation des résultats sous une forme qui fût appropriée à une transmission immédiate à l'expérimentateur. Cela nous a conduit, en même temps que la mémoire assez petite de la 401, à rédiger un certain nombre de programmes nettement spécialisés, au lieu d'essayer de rédiger un programme unique couvrant tous les types de plans les plus courants (1).

(1) Cf. également Rowel (1954), où est décrit un programme pour le calcul séparé de chaque répétition des principaux effets et interactions d'un plan 2^k avec confusion, pour les totaux de traitement présentés dans un ordre standard. Certaines transformations préliminaires des données (122 séries de résultats) et des récapitulatifs des résultats combinés étaient également exécutées par la machine.

Le processus complet d'analyse suivi dans notre programme est exposé ci-après en trois stades : entrée, analyse et sortie. Un programme particulier, celui de l'analyse des expérimentations factorielles 2ⁿ, sera ensuite exposé en détail comme exemple de la façon dont les diverses éventualités survenant dans les expérimentations du même type général peuvent être traitées dans la structure d'un programme unique.

IV - ENTREE ET CALCULS PRELIMINAIRES.

Le premier programme rédigé, celui de l'analyse des blocs avec répartition au hasard (Healy), nécessitait la perforation du numéro de bloc, du numéro de traitement, et du rendement consécutivement pour chaque parcelle. Il se révéla rapidement non-satisfaisant. En premier lieu, il y avait couramment deux ou plusieurs chiffres relevés pour chaque parcelle qui demandaient analyse - par exemple, rendement en grain et rendement en paille dans des expérimentations sur les céréales - et la méthode précédente de présentation des données nécessite une répartition des codages du bloc et du traitement pour chaque variable aléatoire. Et, ce qui est plus important, les chiffres relevés n'étaient habituellement pas ceux dont on avait besoin pour l'analyse. Pour le moins, une conversion (disons de livres par parcelle en quintaux par acre) (1), est nécessaire, et fréquemment de nouvelles quantités sont à déduire de celles qui ont été relevées initialement. Ainsi, dans une expérimentation sur les betteraves, les chiffres relevés peuvent être :

- A - Rendement des racines brutes.
- B - Poids de l'échantillon brut.
- C - Poids de l'échantillon net.
- D - Pourcentage de sucre dans l'échantillon net.
- E - Poids des tiges.

Dans ce cas, les chiffres nécessaires à l'analyse sont :

- 1/ Racines nettes, $A \times C/B$, converti en tonnes par acre.
- 2/ Pourcentage de sucre, D.
- 3/ Sucre total, $D \times A \times C/B$, converti en quintaux par acre.
- 4/ Tiges, E, converti en tonnes par acre.

Sur une machine de bureau, les conversions sont habituellement exécutées après analyse, mais, même dans ce cas, les calculs préliminaires nécessaires demandent presque autant de temps que l'analyse réelle : ils sont également une source féconde d'erreurs.

Il est nettement désirable de faire exécuter la conversion par la calculatrice et, également le calcul de variables déduites, comme le sucre-total si le poids des racines, le pourcentage en sucre, et le sucre total demandent tous une analyse. Le calcul de quantités telles que le poids des racines nettes à partir du poids des racines brutes, et le poids des échantillons bruts et nets, est plus marginal, mais, dans notre cas, nous l'avons également trouvé avantageux.

Comme ces calculs préliminaires ne sont pas affectés par le plan d'expérimentation, un programme général d'entrée pour les plans expérimentaux a été établi (Lipton, Healy). Ce programme lit les données, exécute sur elles les calculs appropriés et enregistre les variables résultantes dans des emplacements

(1) Note du traducteur : livre (pound) = 453, 59 grammes ; quintal (hundredweight ou c. w. t) = 112 livres = 50 k. 80 : acre = 40, 4672 ares ; tonne (20 c. w. t) = 1016, 04 kg.

appropriés de la mémoire. Le programme exécute en outre certaines autres opérations. Premièrement, il transcrit (sur la bande de sortie) un en-tête perforé, dans la bande des données. L'identification des sujets à la sortie est essentielle dans le travail de production courant. Deuxièmement, il procède à certaines vérifications : si le nombre d'éléments corrects n'est pas perforé dans un groupe de données, un indicatif est perforé de façon répétée ; et, pour permettre un certain contrôle sur les relevés bruts et les erreurs de perforation, les valeurs du maximum, du minimum et de la moyenne sont perforées à la lecture de chaque section de données et à chaque formation d'une nouvelle variable aléatoire. Troisièmement, la moyenne et le total de chaque variable sont rangés en mémoire, en même temps que les valeurs de la variable, pour les utiliser dans l'analyse subséquente.

L'emploi de ce programme permet de standardiser la perforation des données, caractéristique qui simplifie beaucoup la tâche de préparation des bandes d'entrée dans la machine.

Le programme opère de la manière suivante : avant chaque opération certains indicatifs sont lus. Ils indiquent au programme s'il doit lire les données (et dans ce cas, les données sont rangées en mémoire), la fonction (s'il y en a une) à calculer, les emplacements des opérantes, et où placer la variable résultante. On dispose des opérations suivantes :

- 0/ Pas de fonction (entrée seulement).
- 1/ $z = x + y$
- 2/ $z = x - y$
- 3/ $z = x \cdot y$
- 4/ $z = x / y$
- 5/ $z = x \cdot c$
- 6/ $z = x$ (transférer les données à de nouveaux emplacements).
- 7/ $z = x + c$
- 8/ $z = \log_{10}(x + c)$
- 9/ $z = \arcsin \sqrt{x}$
- 10/ $z = \sqrt{(x + c)}$

où x et y sont des variables, z la fonction et c une constante. Les produits et les quotients peuvent être mis à l'échelle par des puissances de 10. Dans le programme actuel, les indicatifs sont codés sous forme de pseudo-instructions (1). C'est commode pour l'entrée, car un programme pour l'entrée des instructions est en permanence disponible dans la mémoire. Nous nous proposons toutefois d'introduire un code plus simple pour les indicatifs, qui permettra au programmeur de rédiger les opérations qu'il désire en symboles algébriques, au lieu d'avoir à établir les pseudo-instructions.

Une caractéristique importante de ce programme est que toutes les données relatives à une variable unique sont perforées consécutivement dans la séquence où elles sont enregistrées. Tout ce qui est nécessaire, c'est que la séquence soit la même pour toutes les variables. C'est un avantage considérable, car l'information sur les différentes variables est souvent enregistrée séparément.

 (1) Dans la 401, une instruction (32 chiffres binaires) se compose d'une série de huit nombres (3, 7, 3, 3, 3, 3, 3, 7 chiffres binaires respectivement), perforés chacun en base décimale (excepté que les nombres supérieurs à 99 sont perforés sous forme de deux caractères).

V - ENTREE DES CODAGES BLOC ET TRAITEMENT.

Le numéro de bloc et le numéro de traitement (en code factoriel pour les expériences factorielles) de chaque parcelle sont perforés une seule fois, à la suite des données. Si, comme c'est habituellement le cas, la séquence est telle que les données se présentent bloc par bloc, on peut omettre les numéros des blocs.

Les besoins pour l'entrée du codage du bloc et du traitement varient selon le plan expérimental, et, par conséquent, cette partie de l'entrée est exécutée par le programme relatif à ce plan. Quand le codage est lu, (ou après vérification), il est perforé à la sortie, de sorte qu'on dispose d'un relevé imprimé. Pourvu que les données soient enregistrées dans l'ordre où elles se présentent dans le champ d'expériences (sans suppression de la caractéristique aléatoire), l'implantation réelle du champ sera (sauf certaines limitations) reproduite dans le relevé imprimé.

En outre, on a tenu compte des variantes possibles du code des traitements: ainsi, dans une expérimentation factorielle $3 \times 3 \times 3$, les niveaux d'un facteur peuvent être codés 0, 1, 2 ou 1, 2, 3, selon la nature des traitements. Il serait bien entendu possible à un perforateur habile de transposer mentalement un de ces codes en l'autre, mais un tel processus est évidemment sujet à amener des erreurs, particulièrement si le code peut varier pour des facteurs différents dans la même expérience. Par conséquent, le code est perforé tel qu'il est écrit, et transposé par la machine à l'entrée. En fait, cette opération se programme très simplement, et elle illustre la facilité avec laquelle on peut donner l'ordre aux calculatrices électroniques d'accepter les données sous la forme la plus commode pour l'utilisateur. Tout ce qui est nécessaire, c'est que les trois valeurs de code des niveaux les plus bas apparaissent d'abord sur la bande, suivies du codage de chaque parcelle. Lors de la lecture, les valeurs sont assemblées en groupes de trois, le premier groupe de trois étant soustrait de chaque groupe subséquent. Une simple boucle double d'instructions assure l'ensemble de la lecture. Les opérations pour une répétition unique (27 parcelles) sans codage des blocs sont essentiellement les suivantes (P et Q sont les registres) :

Répéter 28 fois	}	Répéter 3 fois	{	<p>Remettre P à zéro.</p> <p>Décaler P.</p> <p>Lire un caractère perforé, et additionner à P.</p> <p>A la première occurrence, transférer(P) à (Q). A la seconde occurrence et aux suivantes, former (P) - (Q) et mettre en mémoire.</p>
--------------------	---	-------------------	---	--

En réalité, il y a naturellement des complications supplémentaires. On doit prévoir la lecture du codage des blocs s'il est perforé, ou la suppléer, s'il est implicite. Au commencement, il faut lire la bande jusqu'à la rencontre du caractère de début, et à la fin, on doit faire une vérification de ce que le caractère final a été atteint (Si ce n'est pas le cas, des 1 seront continuellement perforés, ce qui indiquera alors un nombre incorrect de caractères sur la bande). On doit en outre prévoir la lecture et la perforation du retour de chariot et du saut de ligne à des intervalles appropriés.

Le codage du bloc et du traitement est vérifié après l'entrée, pour s'assurer, par exemple, que chaque traitement se présente une seule fois dans chaque bloc d'une expérience sur blocs traités au hasard. En pratique, on peut ordinairement le faire en utilisant une section du programme d'analyse principale.

Dans le programme 2ⁿ (décrit ci-après), l'énumération des degrés "confondus" de liberté assure une vérification suffisante du codage du bloc et du traitement.

VI - ANALYSE.

En tant que problème de calcul, l'analyse de données expérimentales ne présente pas de difficultés formidables. Les processus que nous employons sont, pour leur partie principale, semblables à ceux qu'on emploie sur machines de bureau, bien qu'on y ait apporté certaines modifications.

Un stade de base de l'analyse est la formation de divers totaux pour les blocs, les lignes et les colonnes, les traitements, etc.. C'est tout à fait direct sur une machine à modification automatique des instructions (modification B). La partie du mot de code correspondant au bloc et du traitement pour chaque parcelle est utilisée pour indiquer à quel membre de chaque série de totaux le rendement correspondant est à ajouter. Ainsi, par exemple, l'indicatif d'une expérimentation factorielle 3³ est perforé sous la forme BPQR, où chacune des lettres B, P, Q et R est un nombre compris entre 0 et 2 ou entre 1 et 3, spécifiant respectivement le numéro de bloc, et le niveau de chacun des 3 facteurs. Pendant l'entrée il est modifié et mis en mémoire sous forme de 10 chiffres binaires (chaque paire ayant l'étendue 0 à 2) sous la forme BPQRP. Par décalages successifs, il est décomposé pour donner B, PQ, QR, et RP, et on utilise ces nombres pour identifier le total de bloc et l'inscription particulière dans chacune des 3 tables à double entrée. Les chiffres qui composent (par exemple) PQ vont en réalité de 0000 = 0 à 1010 = 10, en omettant le 3 et le 7 ; on attribue à la table 12 emplacements de mémoire consécutifs, les trois lacunes étant utilisées pour procurer les totaux de lignes qui constituent un des principaux effets. Après conversion en moyennes, les 12 nombres sont imprimés sous forme d'une table 3 x 4 par la méthode décrite dans la section suivante.

Quand on a formé les totaux de blocs de traitements, et qu'on les a convertis en moyennes, on calcule et on place en mémoire les résidus pour chaque parcelle. Les résultats ne sont plus exacts, et pour éviter des erreurs d'arrondi, nous prenons 10 chiffres binaires supplémentaires (environ 3 positions décimales supplémentaires), à la fois pour les moyennes et pour les résidus. Quand on forme les résidus, le problème d'associer les moyennes appropriées de blocs et de traitements à chaque rendement est un de ceux qui ont déjà été résolus pour la formation des totaux, et on utilise la même séquence d'instructions, les modifications mineures nécessaires étant faites par la machine.

La formation et la mise en mémoire des résidus jouent un rôle important dans nos programmes. Pour commencer, ils forment la base de vérifications des totaux et des sommes de carrés ils forment également une partie de notre mécanisme pour traiter la covariance et les parcelles manquantes, décrit ci-après. Indépendamment de cela, la possibilité d'imprimer le résultat sur le même tableau que les parcelles initiales a montré son intérêt pour examiner les causes d'erreurs-types trop élevées.

Le stade suivant de l'analyse est le calcul des sommes de carrés pour l'analyse de la variance. Ici, le principal danger est qu'un débordement se produise, c'est à dire qu'on dépasse la capacité du registre. Cela peut s'éviter en employant une moyenne de travail. Comme la machine n'a pas de préférence pour les nombres "simples", il est aussi commode de travailler avec les écarts à la moyenne réelle. Cela a le grand avantage qu'on n'a pas besoin de "correction à la moyenne".

Il faut, à cet égard, noter le contraste entre calculatrices électroniques et calculateurs de bureau. L'emploi des écarts à la moyenne dans les calculs sur machine de bureau serait excessivement fastidieux, parce qu'ils sont pénibles à

calculer, doivent être écrits quand on les calcule, et requièrent la conservation de décimales supplémentaires pour assurer la précision nécessaire. D'autre part, sur les calculatrices électroniques, on peut programmer très simplement le calcul des écarts, qui prend peu de temps pour son exécution. En outre, une calculatrice électronique est complètement indifférente au nombre de chiffres dans les nombres qu'elle manipule, pourvu qu'ils restent dans la capacité des registres.

Deux autres alternatives possibles, que nous avons toutes les deux utilisées, impliquent de travailler avec des chiffres exacts, en formant soit $n\sum x^2 - (\sum x)^2$, soit $\sum [nx - \Sigma(\bar{x})]^2$. La première expression est n , et la seconde n^2 , multipliées par $\Sigma(x - \bar{x})^2$. Dans ces alternatives, la plus grande taille des nombres est dans une certaine mesure compensée par l'absence de chiffres à surveiller.

Indépendamment de l'entrée et de la sortie, la calculatrice 401 n'a pas de caractéristiques de vérification incorporées par construction, et tous nos programmes comportent des vérifications pour nous prémunir contre les fautes momentanées de la machine qui se produisent éventuellement. Une excellente vérification générale est basée sur la propriété additive de l'analyse du tableau de la variance, du fait que tous ses éléments y compris la somme résiduelle des carrés, sont calculés directement. Une autre vérification qu'on peut exécuter est celle de former les totaux de bloc, de traitement, etc., des résidus, qui doivent être tous égaux à zéro (indépendamment des erreurs d'arrondi) ; on peut utiliser à cette fin la même séquence d'instructions pour former les totaux et les résidus.

L'action entreprise quand une vérification est défailante demande à être considérée. Dans nos premiers programmes, on faisait simplement arrêter la machine. Toutefois, cela ne fut pas trouvé satisfaisant parce que cela entraînait trop de contemplation stérile des lampes signal. Par suite, notre politique actuelle, quand une vérification est défailante, est de répéter la section appropriée du programme, ou autrement de continuer de la façon habituelle, laissant l'action ultérieure à décider après examen des résultats. Les vérifications qui ont été satisfaisantes, sont indiquées par la perforation d'un point. A ce propos, il faut mentionner que quand on utilise la sortie par perforation (comme c'est toujours le cas dans le travail de production), les résultats ne sont disponibles pour examen qu'après avoir été imprimés sur la téléimprimante, processus qui introduit un décalage de temps considérable.

Comme notre pratique est de ranger en mémoire les rendements des variables sur des pistes à relais différentes, l'analyse successive des différentes variables est très simple. Comme mentionné précédemment, une référence à la piste 7 implique une référence à la piste à relais qui est en circuit à ce moment. Une modification d'une variable à la suivante s'effectue donc simplement en changeant les instructions de commutation des relais. En pratique, il est souvent commode d'employer un petit sous-programme qui commute à la piste de la variable en cours chaque fois que c'est nécessaire.

VII - SORTIE.

Ici, nous nous sommes efforcés de simplifier les choses autant que possible à l'expérimentateur qui doit faire usage des résultats, cela requiert que :

- a/ tous les résultats nécessaires soient fournis, et que
- b/ la présentation soit satisfaisante.

La fourniture de résultats supplémentaires qui, en fait, ne sont pas nécessaires, alors qu'elle gaspille le temps de sortie, n'a pas grande importance pour

l'expérimentateur, puisqu'il peut aisément les ignorer. Par conséquent, nous tendons à nous écarter de l'attitude de donner trop plutôt que trop peu, car l'expérimentateur souvent ne sait pas lui-même, jusqu'à ce qu'il examine les résultats, ce qui lui est réellement nécessaire. Cette pléthore de résultats numériques est en concomitance inévitable avec beaucoup de travail de recherche statistique produit mécaniquement. Le crayon bleu doit être gardé à portée de la main.

Ces besoins de sortie ont pour conséquence qu'une partie considérable de la plupart de nos programmes d'analyse des expériences est consacrée à organiser la mise en page des résultats imprimés.

La base de toute sortie de matière numérique est un sous-programme de sortie universel (Lipton). Etant donné un nombre de base binaire dans la machine, ce programme l'interprète soit comme fraction, soit comme nombre entier mis à l'échelle par une puissance de 10 (auquel cas on introduit une virgule convenablement placée); convertit le résultat en base décimale et perfore le résultat avec un nombre prédéterminé de chiffres. Le sous-programme prévoit l'insertion d'un espace ou d'un retour de chariot avec saut de ligne avant chaque nombre, selon besoins.

Avec ce sous-programme de base, on peut former des programmes de sortie plus complexes. Ainsi, il peut être nécessaire d'imprimer une série de nombres classés dans des emplacements consécutifs sous forme d'une table à deux entrées. Avec 32 nombres au plus, cela peut s'obtenir en employant un mot indicatif (1) consistant en un schéma de 1 et de 0. Quand chaque nombre est imprimé, l'indicatif est décalé d'une position binaire vers la gauche, et le chiffre de gauche est examiné (ces opérations sont toutes les deux comprises dans le code de la machine). Un 1 provoque le début de l'impression d'une nouvelle ligne, et l'achèvement du tableau est indiqué par l'ensemble de l'indicatif devenant zéro.

Indépendamment de l'emploi de mise en page standard, on aide considérablement l'expérimentateur s'il a une certaine quantité de repères sur les résultats imprimés. Les repères sont coûteux, à la fois en temps de machine et en instructions, mais nous introduisons habituellement des signes + devant les erreurs-types, et nous prévoyons quelques autres indications de la signification des divers chiffres.

Quand une variable aléatoire est complètement analysée, il est possible de perfore les chiffres réels analysés, les résidus, l'analyse du tableau de la variance, et les moyennes et effets du traitement avec leurs erreurs-types. Ce ne sera pas nécessaire pour toutes les variables et, la suppression facultative de certains ou de tous d'entre eux est commandée par les indicatifs sous forme de pseudo-instructions, une pour chaque variable analysée. Ces indicatifs donnent également une information sur le nombre de positions décimales dans les données et dans les moyennes (les virgules étant supprimées quand on perfore les données) et sur les sommes résiduelles de produits qui peuvent être nécessaires.

VIII - CARACTERISTIQUES SPECIALES.

Trois complications qui peuvent se rencontrer dans les plans expérimentaux ordinaires peuvent être traitées, partiellement ou complètement, par nos programmes. La première est la covariance. En l'absence d'une demande pressante, nous n'avons jusqu'ici pas rédigé de programmes complets pour l'analyse de la covariance, mais la somme résiduelle des produits nécessaires pour une telle analyse peut être fournie sans difficulté; elle s'obtient en multipliant les

(1) Voir note au bas de la page 50

rendements d'une variable par les résidus correspondants de l'autre, ou en multipliant les résidus correspondants des deux variables. Etant donné ces chiffres, on peut obtenir sans trop de travail sur une machine de bureau le coefficient d'estimation et la somme réduite des carrés, et de là, les moyennes ajustées avec leurs erreurs-types. Dans le cas de blocs avec répartition au hasard des traitements, avec parcelles subdivisées, le programme a été étendu pour fournir, pour les deux parties de l'analyse, les coefficients d'estimation et les sommes des carrés expliquées par les estimations (Gower).

Un plan peut avoir ses parcelles divisées en deux ou plusieurs sous-parcelles, pour l'introduction d'un nouveau facteur. Les blocs avec traitements répartis au hasard et avec parcelles subdivisées sont suffisamment courants pour mériter un programme spécialement rédigé avec entrée et sortie complètement organisée (N° 3 dans la liste de la section 1), mais un plan à parcelles subdivisées peut s'analyser presque complètement en employant le programme général d'entrée. A cette fin, les sous-parcelles recevant chaque niveau du facteur de sous-parcelle sont traitées comme des variables aléatoires séparées et d'après cela, le programme d'entrée forme une série de fonctions orthogonales ; elles sont analysées séparément, et les analyses résultantes sont ensuite groupées. Dans le cas le plus courant de parcelles divisées en deux parties, l'analyse des sommes et des différences exécutée de cette manière permet ce qui est effectivement une analyse complète.

La troisième complication est celle des valeurs manquantes. Nous avons développé une méthode générale pour les traiter, qui est indépendante du plan et qui a des avantages certains sur d'autres méthodes qui ont été proposées (Tocher, 1952 ; Hartley, 1956). Une description complète en a été publiée (Healy et Westmacott, 1956), et nous nous bornerons à l'ébaucher ici. En bref, des valeurs supposées (qui peuvent être très grossières) sont introduites pour les unités manquantes, et on procède normalement à l'analyse jusqu'à la formation des résidus. A ce stade, on soustrait les résidus correspondant aux unités manquantes des valeurs supposées, et on répète le processus jusqu'à ce que les résidus deviennent petits (en pratique, jusqu'à ce que le carré résiduel de la moyenne cesse de décroître). Cette méthode utilise très peu d'instructions et est tout à fait générale. On peut, par conséquent l'incorporer facilement dans un programme. Elle est presque aussi rapide pour plusieurs valeurs manquantes que pour une seule.

IX - L'ANALYSE DES PLANS FACTORIELS 2°

Comme exemple d'un programme complexe qui illustre plusieurs points généraux exposés précédemment, et qui contient en outre des points d'intérêt particulier, nous décrirons en assez grand détail le programme pour les plans 2°. Ce programme a été établi pour traiter des expérimentations contenant jusqu'à 128 parcelles, y compris les expérimentations avec une seule répétition, dans lesquelles une estimation de l'erreur due aux interactions d'ordre supérieur est nécessaire. Il peut aussi traiter des plans impliquant une répétition fractionnaire. Il tient compte des confusions, qu'elles soient totales ou partielles, sans nécessiter la spécification avant analyse des degrés de liberté qui sont réellement "confondus" (sujet qui est souvent pénible et amène des erreurs). Il peut également traiter des facteurs à 4 et 8 niveaux. Une simple addition au programme permet l'analyse du carré latin 8 x 8 et du carré quasi-latin 8 x 8.

Nous appellerons n le nombre des facteurs (ou pseudo-facteurs) à deux niveaux, r le nombre des répétitions, b le nombre des blocs, et N le nombre des parcelles.

a/ Sous-programmes spéciaux.

Diverses parties du programme ont été organisées sous la forme de sous-programmes formels. Il y a 6 tels sous-programmes.

- 1/ Calcul des effets.
- 2/ Carrés des sommes.
- 3/ Formation des totaux des traitements et classement de ces derniers dans un ordre standard.
- 4/ Test du désaccord entre deux quantités.
- 5/ Impression du contenu d'emplacements de mémoire consécutifs en lignes de quatre.
- 6/ Commutation à la piste à relais des variables aléatoires analysées. Indépendamment de (1) qui est décrit ci-après, il faut mentionner quelques points :
 - 2/ imprime un indicatif si la somme des carrés déborde.
 - 4/ à trois variantes, la première étant exacte, la seconde et la troisième donnant des désaccords différents admissibles, selon la vérification des totaux, des moyennes et des sommes de carrés.
 - 5/ utilise le programme de sortie universel, en l'organisant de façon à introduire correctement les virgules et à ajouter la moyenne générale avant impression, si c'est nécessaire ; si on le désire, on peut introduire dans les résultats une position décimale supplémentaire.

L'emploi libre de sous-programmes spéciaux a deux avantages. Quand une opération, ou quand des variantes de la même opération qu'on peut traiter avec substantiellement la même séquence d'instructions, doivent être exécutées un certain nombre de fois dans le programme, la rédaction du programme est considérablement simplifiée si l'on peut appeler l'opération chaque fois qu'elle est nécessaire, en entrant dans le sous-programme. Les tests du programme et la localisation et la correction des erreurs sont en outre rendus considérablement plus faciles, puisque les sous-programmes, constituant des entités indépendantes, peuvent être testés séparément, sans difficultés, en ne permettant au programme principal d'y recourir que quand ils opèrent tous correctement.

b/ Sous-programme des effets.

Le coeur de l'analyse d'une expérimentation 2^n est le calcul des estimations des effets principaux et des principales interactions (que nous dénommerons collectivement les effets), d'après les moyennes des traitements.

La moyenne employée est la technique standard "plus et moins" (Yates, 1937) qui se programme très simplement. Le processus utilisé est d'osciller entre une piste sans relais E et une piste à relais W. Initialement, E contient les moyennes de traitements (en réalité la moyenne de l'écart ou les moyennes de blocs) classées dans un ordre standard. Les sommes et les différences des paires successives sont formées et placées en mémoire sur W. En changeant les constantes de modification, on répète le processus en sens inverse. Après n répétitions du processus, on extrait les résultats finaux de la piste E ou de la piste W, selon le cas, on les divise par 2^{n-1} et on les place en mémoire sur la piste E, qui contient alors les estimations des effets en unités classiques, et également dans l'ordre standard.

Trois autres variantes de ce processus sont incorporées dans le sous-programme. La seconde omet simplement la division finale. La troisième et la quatrième exécutent le processus inverse, de calculer les moyennes de traitement à partir des effets, en opérant sur les valeurs de la piste E, ou sur une autre piste T, respectivement.

c/ Identification des degrés "confondus" de liberté.

Avant de procéder à l'analyse des variables distinctes, il est nécessaire de déterminer quels contrastes sont confondus dans les blocs. Cela se fait en établissant un codage de bloc "étendu", dans lequel les b derniers chiffres d'une série de N mots, correspondant aux N parcelles, sont attribués à raison d'un à chacun des b blocs ; toutes les parcelles se présentant dans le bloc c, par exemple, ayant le chiffre 32 - (c - 1) codé 1 et le reste 0. Ces N mots sont alors traités comme s'ils étaient des rendements, les effets et les interactions étant calculés par le sous-programme d'effets. Tout effet non-zéro indique une confusion (partielle ou complète) du degré de liberté correspondant. Pour éviter la perte de chiffres à l'extrémité la moins significative, il faut examiner les totaux d'effets et, pour cette raison, on utilise la seconde variante du sous-programme (s'il y a 32 blocs et 128 parcelles, une confusion survenant seulement dans les blocs 31 et 32 ne sera pas identifiée par suite du débordement, mais c'est une éventualité qui ne peut presque pas se présenter en pratique).

Quand les "effets de bloc" ont été calculés, on les teste chacun à son tour, et on relève les degrés "confondus" de liberté avec un 1 dans les chiffres successifs de jusqu'à quatre mots dans la mémoire. En même temps, on perfore un relevé de degrés confondus de liberté ; par exemple, si, dans une expérimentation 2^5 , ABD est confondu, on perfore 11010.

d/ Calcul des moyennes et effets de blocs et de traitements.

Le premier stade de l'analyse de chaque variable aléatoire est de former les totaux de blocs, et de remplacer chaque rendement par son écart à la moyenne de bloc appropriée. Ce processus a plusieurs avantages. Les degrés de liberté totalement confondus apparaîtront sous la forme zéro dans la table des effets, et les estimations des effets partiellement confondus seront exemptes de différences de blocs. Opérer avec les écarts aux moyennes de blocs évite en outre le besoin de déduire une correction générale de la moyenne et un élément des blocs quand on calcule la somme erreur des carrés.

Après vérification que la somme des écarts est nulle et formation de leur somme de carrés, les totaux traitement des écarts sont formés et rangés en mémoire sur la piste non-relais E, dans l'ordre standard habituel (1), a, b, ab, c...), et convertis en moyennes. D'après elles, on calcule la somme des carrés des traitements et la moyenne des traitements est ensuite convertie en effets et interactions principaux par la première variante du sous-programme d'effets.

e/ Tableaux de rendements moyens pour un ou plusieurs facteurs.

En plus du tableau des effets et interactions principaux, des tableaux à deux entrées ou plus de rendements moyens sont souvent nécessaires. On peut déduire de tels tableaux du tableau des effets, par extraction de tous les effets impliquant seulement ceux des facteurs qui entrent dans le tableau, et en appliquant un processus inverse de celui qu'on utilise pour calculer les effets eux-mêmes. Ainsi, pour établir le tableau à double entrée pour A et B, on a besoin des effets O, A, B et AB.

Pour exécuter ce calcul, on sélectionne les effets nécessaires pour un tableau donné, et on les place consécutivement et dans le même ordre (qui est en fait l'ordre standard) sur une autre piste T. On exécute ensuite le processus inverse du calcul des effets, au moyen de la quatrième variante du sous-programme des effets.

Aussitôt qu'un tableau est calculé, on le perfore, de sorte que la piste T est libérée pour le calcul d'autres tableaux. Les tableaux réellement établis s'exécutent à la demande de l'expérimentateur. La structure de chaque tableau est régie par un mot de commande, dont les chiffres spécifient les facteurs à y in-

troduire. Les mots de commande pour les tableaux à trois entrées ou plus sont fournis par l'expérimentateur, et la machine peut, en outre, recevoir instruction d'imprimer tous les tableaux à une entrée, ou à deux entrées, ou les deux. Pour cela, elle calcule ses propres mots de commande. Le mot de commande de chaque tableau est imprimé en tête du tableau.

f/ Estimation des erreurs d'interactions d'ordre supérieur.

Dans les expérimentations à répétition unique, les degrés de liberté correspondant à des interactions d'ordre supérieur sont à utiliser pour l'estimation de l'erreur. Les degrés "confondus" de liberté doivent, bien entendu être exclus. Le choix des degrés de liberté de l'erreur, appropriés, est régi par jusqu'à quatre mots de commande, dont les chiffres représentent les degrés de liberté successifs dans l'ordre standard. La machine elle-même traite les mots de commande nécessaires pour extraire toutes les interactions à trois facteurs et d'ordre supérieur, le choix particulier requis étant spécifié par un nombre de commande unique fourni par l'expérimentateur. Si une certaine combinaison est nécessaire, par exemple, s'il faut introduire certaines interactions à trois facteurs, mais non toutes dans l'erreur, on introduit alors une série spéciale de mots de commande au début de l'analyse.

Au cours du calcul, la machine fixe à zéro les effets correspondants aux degrés de liberté spécifiés par les mots de commande désignés, et, en même temps, ajoute le nombre ainsi fixé pour donner les degrés de liberté pour l'erreur en omettant dans ce comptage les degrés de liberté qui sont confondus. La différence entre la somme des carrés pour tous les effets et celle pour les effets après fixation à zéro des termes d'erreur donne la somme des carrés pour l'erreur.

Les effets, restant après fixation à zéro des termes d'erreur sont en outre utilisés pour calculer les résidus. Ils sont d'abord mis en oeuvre par la troisième variante du sous-programme d'effets, obtenant ainsi des rendements "reconstitués" qui sont exempts des constituants "erreur" de la variation comme constituants de blocs. Ces rendements reconstitués sont alors soustraits des écarts à la moyenne de bloc pour donner les résidus.

Quand les résidus ont été formés et vérifiés, on calcule et on perfore le carré moyen de l'erreur, les erreurs-types et toutes les sommes résiduelles de produit qu'on désire.

g/ Confusion partielle.

Les expérimentations à confusion partielle sont traitées de la même manière que les expérimentations à confusion totale, sans différenciation entre les degrés de liberté partiellement et totalement confondus. Cela affecte l'estimation des effets partiellement confondus et l'estimation de l'erreur de la manière suivante.

Les estimations des effets partiellement confondus seront "atténuées". Si, par exemple, dans une expérimentation à quatre répétitions, quatre degrés de liberté sont partiellement confondus, un dans chaque répétition, les estimations des effets correspondants fournies par les analyses seront les trois quarts de leur valeur correcte. Bien entendu, il serait possible (bien qu'avec certaines complications supplémentaires), de multiplier de telles estimations par les facteurs voulus, mais nous l'avons considéré comme inopportun, car le constituant erreur de tels effets se trouve de cette façon grossi, et des irrégularités excessives seront introduites dans certains tableaux à plusieurs entrées dans lesquelles ils entrent ; on doit s'attendre à ce que les effets vrais correspondants, qui sont en général des interactions d'ordre supérieur, soient faibles. Comme les degrés confondus de liberté sont identifiés par l'analyse (bien que non dans la mesure

où ils sont confondus) les ajustements nécessaires peuvent se faire manuellement si l'on considère qu'ils sont nécessaires.

L'estimation de l'erreur fournie par l'analyse contiendra également des composants de ces effets partiellement confondus. C'est une conséquence de la méthode de calcul de la somme erreur de carrés employée par le programme, quand il y a plus d'une répétition, qui est de déduire la somme des carrés des traitements, en ignorant la confusion, de la somme des carrés des écarts aux moyennes de bloc. Quand on suit ce processus dans une expérience partiellement confondue, par exemple si les effets E_1, E_2, E_3 ayant des carrés moyens R_1, R_2, R_3 sont partiellement confondus dans des fractions p_1, p_2, p_3 de toutes les répétitions ($p_1 + p_2 + p_3 = 1$), la somme erreur des carrés sera augmentée de $p_1 R_1 + p_2 R_2 + p_3 R_3$, et les degrés de liberté pour l'erreur seront augmentés de 1. Par conséquent, pourvu que les effets partiellement confondus soient faibles par rapport à leurs erreurs d'estimation, il n'en résultera pas de perturbation sérieuse dans l'estimation de l'erreur.

h/ Répétitions fractionnaires.

On peut analyser les expérimentations à répétition fractionnaire sans modification du programme, en omettant certains facteurs du codage du traitement. Les facteurs omis doivent être tels que les facteurs restants spécifient uniquement les diverses combinaisons de traitements.

i/ Carrés latins et quasi-latins.

Les modifications nécessaires pour adapter le programme à l'analyse des carrés latins et quasi-latins 8×8 sont très simples. Elles ne requièrent essentiellement que le remplacement des rendements par les écarts aux moyennes de ligne et de colonne, au lieu des écarts aux moyennes de bloc. Le codage du bloc est remplacé par un codage de ligne et de colonne (3 + 3 chiffres). Pour identifier les degrés confondus de liberté, il est nécessaire d'étendre les constituants de ligne et de colonne de ce codage indépendamment, de façon à donner un codage étendu occupant 16 chiffres.

j/ Organisation de la mémoire.

L'organisation de la mémoire pour un grand programme de cette nature mérite une certaine considération. Deux points doivent rester à l'esprit. Premièrement, pour réduire la commutation des relais, il faut autant que possible éviter une référence simultanée à plus d'une piste à relais pour les données etc. Deuxièmement, pour économiser l'espace de mémoire, il faut ordonner les calculs de façon à ce que la même piste puisse être utilisée à un certain nombre de fins.

Le schéma réel adopté a été le suivant : les données (rendements, etc.), pour jusqu'à 8 variables aléatoires, sont placées sur les pistes à relais 7/0 à 7/7. Quand chaque variable est analysée, les données initiales sur la piste de la variable sont remplacées successivement par les écarts aux moyennes de bloc et par les résidus. Le codage bloc et traitement est rassemblé et placé en mémoire pendant l'analyse sur la piste 5. Les pistes 6 et 7/8 sont les pistes E et W des sous-programmes des effets. On utilise en outre la piste 6 pendant l'entrée des données pour placer une partie du sous-programme d'entrée. La piste 3 (00 - 31) reçoit successivement les totaux de bloc, les moyennes de bloc, et opère comme piste T dans le sous-programme des effets. Le sous-programme général de sortie est placé sur la piste 7/10, et le sous-programme général d'entrée pour les plans d'expérimentation sur les pistes 7/14 et 7/15. Le programme lui-même est placé sur les pistes 1 à 4, 7/9 et 7/11 à 7/13, toutes les séquences importantes d'instructions qui ne contiennent pas de référence à une piste à relais étant elles-mêmes placées sur une piste à relais. (Cet emploi combiné de

pistes à relais et sans relais pour les instructions d'un programme s'est révélé très utile dans la rédaction de programmes longs pour la 401). Ni les pistes à relais ni les pistes sans relais ne sont remplies de façon serrée, de sorte qu'on peut y loger des modifications et des additions.

k/ Etablissement du programme.

Le programme complet a été établi à l'aide d'un programme d'assemblage automatique qui a été rédigé pour la 401 (Yates et Lipton, 1957). Ce programme simplifie considérablement la rédaction des instructions, et élimine plusieurs types d'erreurs. Il permet en outre de prendre des décisions sur l'organisation de la mémoire après achèvement du programme prévisionnel. Cela s'est révélé très avantageux dans des programmes pour l'analyse d'expérimentations répétées, car il est extrêmement difficile de décider l'organisation la plus satisfaisante tant que le programme n'est pas presque terminé, et que toutes les difficultés n'ont pas été explorées.

X - CONCLUSIONS.

Malgré les doutes exprimés assez imprudemment par l'un des auteurs (Mr. J. R. H. - discussion avec Tocher, 1952), avant qu'il eut accès à une calculatrice à grande vitesse, l'analyse programmée des expérimentations s'est révélée une application très utile des calculatrices électroniques aux statistiques. Indépendamment de la perforation et de la vérification des bandes de données, l'analyse d'une expérimentation-type prend environ une minute et demie par variable aléatoire, la partie principale de ce temps étant consacrée au volume assez important de la sortie nécessaire. Les résultats sont précis, et nettement présentés sous forme aisément exploitable. L'agriculture étant une occupation saisonnière, la charge des travaux de calcul pendant les mois d'hiver était antérieurement une surcharge considérable pour les moyens du service, et il en résultait beaucoup de retard. L'emploi d'une calculatrice nous a permis de rendre des services plus efficaces, et, en outre, d'entreprendre un volume substantiel d'analyse expérimentale pour d'autres organisations de recherche, que nous n'aurions certainement pas pu envisager antérieurement.

En plus de la vitesse et de la précision, d'autres avantages sont la prévision des résidus, l'aptitude à traiter simplement les parcelles manquantes, la réduction du travail manuel, et la suppression de la nécessité d'enseigner à des calculateurs humains des analyses compliquées telles que celles des plans 3^3 et 2^n .

Le temps et l'effort requis pour rédiger les programmes nécessaires sont, bien entendu, considérables, et il n'en vaudrait pas la peine s'il fallait programmer la machine pour quelques analyses d'un plan exceptionnel. Pour des tâches répétitives, telles que l'analyse des types les plus courants d'expérimentation, toutefois, le temps consacré à la programmation est de la nature d'un investissement de capital. En outre, bien que les codes d'instruction et les modes de fonctionnement des différents types de machines varient considérablement, on peut habituellement transposer un programme rédigé pour une machine, pour l'utiliser sur une autre machine d'un type différent, mais ayant une capacité de mémoire identique ou supérieure, avec un peu de travail, certainement beaucoup moins qu'il n'en faudrait si le problème entier devait être considéré ab initio. Pour cette raison, l'effort de programmation que nous avons fait sur la 401 ne serait nullement perdu si nous changions de machine.

REFERENCES

- ASHTON, HEALY et LIPTON (1957) - "The descriptive use of discriminant functions in physical anthropology" - Proc. Roy. Soc. B - 146 - p. 552 à 572.
- CHURCH et LIPTON (1956) - "The use of an electronic computer in the estimation of sampling errors in a nutritional survey". Brit. J. Nutr - 10 - pp. 27 à 32.
- FISHER et HEALY (1956) - "New tables of Behren's test of significance". J.R.S.S. B - 18 - pp. 212 à 216.
- FISHER et YATES (1957) - "Statistical tables" 5^e édition - Edimbourg et Londres - Oliver and Boyd.
- FOSTER et REES (1957) - "Upper percentage points of the generalized Beta distribution" - Biometrika - 44 - pp. 237 à 247.
- HARTLEY (1956) - "Programming analysis of variance for general purpose computers" - Biometrics - 12 - pp. 110 à 122.
- HEALY et WESTMACOTT (1956) - "Missing values in experiments analysed on automatic computers". - Appl. Stat - 5 - pp. 203 à 206.
- LE COUTER et LIPTON (1955) - "Non linear regenerative extraction of synchrocyclotron beams" - Phil. Mag - 46 - pp. 1265 à 1280.
- LIPTON (1955) - "Note on the electronic computer at Rothamsted" Math. Tab. Wash - 9 - p. 69.
- LIPTON (1957) - "Some statistical applications of electronic computers" - Appl. Stat. - 6 - pp. 102 à 103.
- PATTERSON (1956) - "A simple method for fitting an asymptotic regression curve" - Biometrics - 12 - pp. 323 à 329.
- ROWELL (1954) - "The analysis of a factorial experiment (with confounding) on an electronic calculator". J. R. S. S. - B - 16 - pp. 242 à 246.
- SHOFIELD et REES - "Calculation of membrane pressure for a sol of spherical particles". J. Colloid - Sci. - 11 - pp. 617 à 622.
- TOCHER (1952) - "The design of analysis of bloc experiments". J. R. S. S. - B - 14 - pp. 45 à 100.
- YATES (1937) - "The design and analysis of factorial experiments" Tech. Comm. 35, Comm. Bur. Soil. Sci.
- YATES et LIPTON (1957) - "An automatic programming routine for the Elliott 401" - J. Ass. Comp. Machinery - 4 - pp. 151 à 156.

DISCUSSION DE LA COMMUNICATION.

Dr. K. D. TOCHER - Ayant à l'esprit que Mr. Healy et ses coauteurs commencèrent cette réalisation avec le grand inconvénient de ne pas croire qu'ils obtiendraient des résultats utiles, je pense qu'il faut les féliciter de la persévérance de l'ingéniosité et du travail difficile qu'implique l'obtention actuelle des très solides résultats qu'ils ont décrits - d'autant plus que, comme je le suppose, des gens à l'esprit assez vicieux peuvent dire que la machine 401 de Rothamsted est probablement l'une des machines les moins appropriées du monde aux calculs statistiques. Avec toutes ces difficultés, un résultat important est tout à fait digne d'éloges.

Bien que cette machine puisse se qualifier de machine à un seul niveau, elle est, en fait, comme l'a mentionné M. Healy réellement très semblable à une machine à deux niveaux, dans laquelle la commutation assure le transfert entre mémoires, employé sur les types les plus classiques de machines, donnant un accès immédiat à 128 mots. Comme c'est une forme de mémoire par ligne à retard, selon les standards plus modernes, il faut le considérer comme une mémoire à accès intermédiaire et non à accès immédiat. Ainsi, la 401 peut se considérer comme n'ayant pas de mémoire à accès immédiat, et celà, je suis sûr que les auteurs seront d'accord, rend la vie un peu plus pénible à ceux qui doivent obtenir une certaine vitesse de fonctionnement.

Ce problème n'est pas particulier aux statistiques. Pour traiter de grands volumes de données qui ont à être divisées en groupes pour les loger dans la mémoire à accès immédiat, le problème est d'articuler les données elles-mêmes en bloc par une division naturelle qui coïncide avec les divisions de la machine. Ou l'on trouve que la machine saisit des groupes beaucoup trop grands, et par conséquent, trop peu de groupes, ou bien la possibilité opposée, que les groupes sont réellement trop petits, et qu'il faut faire procéder à un grand nombre de transferts entre mémoires pour traiter ce qu'on doit regarder comme une entité unique de données. C'est en fait une difficulté commune à presque toutes les machines à deux niveaux. L'étendue des variations naturelles de la taille des groupes qu'on souhaiterait traiter est si grande qu'on ne peut fixer une taille du bloc d'informations qui serait appropriée à la majorité des cas : il me semble qu'il est grand temps que les constructeurs commencent à se rendre compte que c'est un des principaux problèmes de l'emploi facile de ces machines, et qu'ils doivent songer à réaliser des machines qui se comportent comme si elles étaient des machines à un seul niveau.

Nous voudrions qu'ils construisent de grandes machines à un seul niveau, mais, soyons réalistes, nous ne les aurons pas dans notre pays dans les années à venir - toutefois, nous pouvons facilement disposer dans la machine, des dispositifs qui permettent de la programmer comme si elles étaient à niveau unique, de sorte que, quand on appelle un nombre qui n'est pas dans le bloc de la mémoire à accès immédiat, tous les stades d'obtention de cette information aient lieu automatiquement. Il faudra toujours plus de temps pour obtenir ce nombre que pour obtenir un nombre déjà dans la mémoire à accès immédiat, mais, au moins, la rédaction du programme sera d'une grande simplicité. Bien entendu, le programmeur aura encore à garder conscience de l'arrangement à deux niveaux, s'il désire réduire au minimum la durée des opérations, mais il sera libéré des mécanismes détaillés du processus. Cela a été réalisé sous forme de programme sur la Manchester Mark I ; un programme a été rédigé pour permettre d'employer la Mark I, machine à deux niveaux, comme si elle était à niveau unique. L'exagération de la consommation du temps opératoire peut évidemment être réduite pour une étude soignée.

Toutefois, cet inconvénient de la 401 est éclipsé par les dispositifs de sortie ; pour ce type de travail qui demande, relativement parlant, une grande quantité d'impression de données manipulées, il y a, j'estime, un inconvénient par comparaison avec les machines plus grosses. Toutefois, je répète qu'adopter le point de vue qu'on ne doit pas attendre une machine idéale, mais utiliser celle qu'on a, est vraiment très valable.

Je ne doute pas qu'il y ait plusieurs difficultés pratiques, qui ont détourné les auteurs d'établir un programme général.

Toutefois, ils semblent indiquer que la raison de cette omission est la difficulté que les données ont presque toujours besoin d'être transformées avant d'être prêtes pour l'analyse ; pour remédier à cette situation, ils ont réalisé un programme de transformation universel. Il me semble qu'une fois qu'il existe un programme général pour la traduction des données, un programme général d'analyse des expérimentations devient désirable, en outre, j'ai entendu murmurer qu'au moins un des auditeurs envisage, en fait, d'étudier un programme universel.

Je voudrais maintenant faire un ou deux commentaires sur la communication prise dans son ensemble. Je pense qu'elle nous a donné une opinion exacte sur le sujet, avec une mise en relief très valable de toutes sortes de détails dont quelqu'un examinant les choses de l'extérieur pourrait ne pas se rendre compte. Elle insiste très valablement sur la présentation des résultats, sur les vérifications, sur l'impression des résidus, et sur un grand nombre de stratagèmes ingénieux, ce qui montre la quantité d'efforts qui y entrent. Une astuce qui a ma faveur est l'emploi d'une analyse des noms d'indicatifs de blocs étendus pour déterminer les termes confondus. Ma solution à ce problème, donné précédemment, est beaucoup moins élégante.

Dr. VAJDA - Les auteurs méritent notre gratitude à plusieurs points de vue. La communication rend compte de l'expérience d'une calculatrice électronique pour les calculs statistiques, et sans doute, d'autres orateurs en commenteront les détails. Toutefois, il ne faut pas oublier que l'histoire de la calculatrice 401, qui nous intéresse a commencé avant que la machine ne fut définitivement installée à Rothamsted. Quelqu'un - ce doit être l'un des auteurs - doit avoir eu cette idée splendide que la calculatrice, à ce moment sans affectation quand elle était inoccupée, dans son service au Laboratoire Mathématique de Cambridge, pouvait fournir un travail permanent et utile.

Sous ce rapport, nous devons être sûrs que certaines études de recherches étaient poursuivies sur la façon d'utiliser une calculatrice, quand on en dispose d'une. La présente communication ne contient à peu près rien sur l'organisation d'un groupe de calcul (ce n'était pas son objet), mais elle provoque quelques idées sur la question. Il apparaît que le Dr. Yates et ses collègues se sont rangés du côté de ceux qui estiment, comme moi-même, que c'est le mathématicien, ou le statisticien ou l'actuaire, qui doit apprendre le calcul électronique, et non le spécialiste électronique qui doit essayer de devenir un mathématicien ou un actuaire - et bien d'autres choses encore. Ceux qui gèrent commercialement un service général de calcul peuvent avoir des idées différentes sur ce sujet, et de telles idées différentes ont, en fait, été exprimées à une réunion de la société il y a une quinzaine. Toutefois, pour un organisateur qui a besoin d'une calculatrice comme outillage, pour satisfaire à des obligations bien établies plus efficacement, je suis certain que la méthode adoptée à Rothamsted, et pour les objectifs de mon propre groupe au Laboratoire de Recherches de l'Amirauté, est la seule correcte.

Les programmes disponibles pour la calculatrice à Rothamsted sont énu-

mérés dans la section I de la communication. En lisant cette liste, je me demandais si un programme était disponible, ou envisagé, pour la construction des carrés latins, ou peut être pour produire un carré latin aléatoire, quand on en a besoin. Je n'ai pas été surpris que cette idée soit venue au Dr. Yates.

La communication contient quelques à-côtés qui sont tout à fait bienvenus. Il est utile de se voir rappeler qu'une calculatrice électronique n'a pas de préférence pour les nombres "simples", et il est également intéressant d'apprendre le processus adopté quand une vérification est défailante. Je ne pense pas que le processus indiqué soit à recommander dans toutes les circonstances. C'est un problème économique, ou peut être de recherche opérationnelle, de savoir s'il vaut mieux examiner des lampes de signalisation, ou permettre à la machine de continuer à triturer des nombres que l'on sait déjà être complètement dépourvus de sens. Je ne vois pas comment la machine décide si "elle répètera la section appropriée du programme", ou si "autrement, elle continuera de la manière habituelle". Peut être un des auteurs pourra-t-il nous expliquer ce point.

Je suis également très heureux de voir que les auteurs sont humains, même si la calculatrice ne l'est pas. Rarement un programme est correct dès sa première rédaction, et certaines erreurs se sont révélées typiques. Une d'entre elles est liée au comptage. L'exemple donné à la section 2 a dû être abandonné après une première rédaction. On voit facilement que le processus n'additionne pas, en fait, "n" nombres comme il le devrait (1).

Dr. E. S. PAGE - Il est utile de considérer quel type de calculatrice est le mieux approprié aux calculs statistiques. Comme certaines applications ne nécessitent que des opérations simples sur de grandes quantités de données, tandis que d'autres, telles que les analyses à plusieurs variables aléatoires, demandent une analyse numérique appréciable, nous devrions souhaiter une machine ayant les facilités d'exploitation des données des machines d'exploitation rapides et la vitesse d'une calculatrice rapide, malheureusement, cela revient à demander le meilleur de chacune, et à la rendre coûteuse, même par rapport aux prix habituels des calculatrices automatiques. En lisant "entre les lignes" de la communication, peut être incorrectement, je soupçonne que la machine de Rothamsted n'est pas très lourdement chargée, de sorte que l'importance de l'entrée et de la sortie n'est pas un souci sérieux. Peut être cela provient de son acquisition aux moindres frais. Sur un point mineur de terminologie, j'aimerais voir employer moins les mots "à grande vitesse". Les auteurs ont parlé d'une "perforation de bande à grande vitesse", cela doit signifier une allure de sortie d'environ trois nombres de dix chiffres par seconde, alors que la même calculatrice à laquelle la perforatrice est connectée peut additionner deux nombres à dix chiffres un millier de fois plus rapidement sans être pourtant la machine la plus rapide du monde.

Alors que je suis fatigué d'entendre des affirmations générales sur ce qu'on pourrait faire, il est toujours intéressant d'apprendre ce qui a été fait, et comment on l'a fait. L'expérience des auteurs sur la sortie met en relief un des principes cardinaux du calcul automatique ; la forme de la sortie doit être exactement celle dont on a besoin. Il est sot d'avoir à effectuer un travail d'exploitation manuelle des données après que la calculatrice a terminé.

Il y a deux points très mineurs que je souhaiterais soulever. Quand la cal-

(1) Cela a été corrigé depuis. Dans la communication lue, on a interchangé les opérations 3 et 4 ; dans la 401, une instruction combinée unique est utilisée - Ed.

culatrice perfore ses vérifications au milieu d'une masse d'autres sorties, il peut être utile d'imprimer les défaillances immédiatement, si une séquence de symboles fictifs (tels que des CR), qu'on peut facilement reconnaître sur la bande, étaient perforés avant et après la vérification, la portion vérification de la bande pourrait être introduite sous la tête d'interprétation immédiatement. La calculatrice pourrait encore exécuter ses calculs, tandis que la sortie supplémentaire serait négligeable.

Maintenant que les auteurs et leurs collègues ont rédigé des programmes pour l'analyse de plusieurs plans, je souhaiterais leur demander s'il existe une tendance à Rothamsted pour s'adapter plus étroitement à ces plans - ou des plans non-conformes se présentent-ils aussi fréquemment qu'avant ?

Professeur BARNARD - Je souhaiterais demander aux auteurs quelle sorte d'effets ils prévoient qu'aura l'emploi des calculatrices électroniques sur le genre d'enseignement qui se donne dans les branches statistiques des universités - s'il y aura une réduction de l'importance des processus standard de calcul par opposition aux processus non-standard, et si nous devons arriver à regarder ces processus standard comme s'exécutant de façon entièrement automatique.

Dr. PAGE - Il y a un autre point qui concerne une remarque faite par le Dr. Vajda : qui fait la programmation ? Le Dr. Vajda a émis le point de vue que le programmeur spécialisé ou l'expert électronique n'est pas une personne dont on doit désirer l'instruction statistique. Je suis d'accord. Il me semble que la meilleure chose est que chacun fasse lui-même sa propre programmation, en prenant si nécessaire l'avis de ceux dont l'expérience est plus spécialisée. Une personne qui a programmé son propre problème sera capable de réaliser beaucoup plus clairement l'applicabilité de la calculatrice à son propre domaine d'étude, et elle sera capable d'élargir les applications de la calculatrice. Il est peu probable que cela se produise si un spécialiste programmeur ne produit que les résultats immédiatement demandés.

Dr. VAJDA - Quand j'ai dit que je n'étais pas tout à fait sûr que, quand la machine fait une faute, elle doit s'arrêter ou non, je pensais à une machine qui, en plus de l'impression, donne d'autres indications de si et pourquoi elle a fait une faute : par exemple une indication de débordement, ou quelque chose de semblable, où vous avez besoin de faire quelque chose que la machine n'assume pas. Je ne sais pas très bien si la machine 401 a des petites lampes signal ou si elle se borne à s'arrêter. L'économie du sujet dépend généralement de l'organisation et de l'équipement auxiliaire de la machine.

Mr. R. H. TIZARD - Cette communication est extrêmement intéressante par sa mise en relief des points pratiques qu'implique l'analyse des expérimentations sur une calculatrice. Les auteurs ont dit que les mathématiques mises en oeuvre ne sont pas très compliquées, et ils emploient le mot "travail de bureau" qui peut induire en erreur certaines personnes - bien que je pense que nous comprenons très bien ce qu'ils entendent par là. En fait une analyse de données expérimentales, ou, à vrai dire, de tout genre de données, sur la machine, prend très souvent une forme dans laquelle l'énorme importance de la programmation réside dans l'organisation des données, au sens le plus large. Beaucoup de ce travail provient de la sorte de points que le Dr. Tocher a mentionnés - et non des faits de base de l'expérimentation, ou des besoins fondamentaux de l'expérimentateur, mais de l'organisation interne de la machine elle-même.

Dr. G. E. P. BOX - C'est un grand plaisir pour moi de contribuer à la discussion

de cette excellente communication. J'ai eu la chance de visiter Rothamsted deux fois pendant que le travail décrit était en cours. Il était alors très stimulant d'être témoin de l'intérêt et de l'enthousiasme avec lesquels le Dr. Yates et ses collègues mettaient leur calculatrice en oeuvre et il est maintenant très utile de disposer de ce compte-rendu de leurs expériences. Comme c'est, je pense, la première occasion où l'application des calculatrices électroniques au travail statistique est exposée à une réunion de la section des recherches, peut-être me permettra-t-on de considérer le sujet dans un plus large contexte.

L'avènement de la calculatrice électronique est important pour l'expérimentateur et pour le statisticien pour deux raisons.

1/ Elle leur permet d'oeuvrer avec une efficacité beaucoup plus grande qu'ils ne le faisaient jusqu'ici.

2/ Elle leur permet d'entreprendre des choses qui, par suite de la quantité des calculs en jeu, n'auraient pas été envisagées antérieurement.

Un exemple de ce dernier type d'application, qui ne pourrait être envisagé avec succès si on ne dispose pas d'une calculatrice, est le problème général de l'ajustement des fonctions non-linéaires. De telles fonctions se présentent couramment dans des théories particulières du mécanisme du processus étudié.

Supposons que nous ayons une fonction

$$\eta = f(\xi_1, \xi_2, \dots, \xi_k ; \theta_1, \theta_2, \dots, \theta_p)$$

qui contient p paramètres inconnus, et relie un résultat η au niveau de k variables $\xi_1, \xi_2, \dots, \xi_k$, et qu'on dispose des résultats de N expériences dans chacune desquelles η , est observé, et $\xi_1, \xi_2, \dots, \xi_k$ sont connus. Il n'est pas difficile de réaliser un sous-programme de calculatrice qui, partant de valeurs supposées des paramètres, calculera une séquence d'estimations améliorées qui donnera une valeur progressivement plus faible à une certaine mesure globale de la divergence entre résultat observé et calculé, telle que la somme des carrés des résidus.

Mr. Contie et moi-même avons récemment décrit un tel programme (Proc. Institution of Electrical Engineers, - 103, partie B, supplément 1, 1956), que nous avons exécuté sur la calculatrice Deuce. Des programmes améliorés ont été établis à Princeton pour la Maniac et pour l'IBM 704. Les erreurs-type approximatives des estimations peuvent s'obtenir aisément, et, quand on dispose d'une estimation externe de l'erreur expérimentale (par exemple d'après une répétition au hasard des expériences), on peut faire un test approximatif d'adaptation du modèle supposé. Comme le processus total est numérique, il n'est pas nécessaire de connaître la fonction explicitement ; il suffit qu'on puisse la calculer.

En chimie, par exemple, $f(\xi : \theta)$ peut être une relation théorique basée sur la théorie cinétique, auquel cas on l'exprimera par une série d'équations différentielles simultanées qui n'auront souvent pas de solution explicite. On peut alors exécuter le programme des moindres carrés en liaison avec un sous-programme numérique pour des équations différentielles. Par emploi de sous-programmes appropriés, on peut ajuster directement aux données des fonctions exprimées par des équations différentielles ordinaires, des équations différentielles partielles, des équations intégrales, etc. Il est clair qu'un programme universel de ce genre peut être également établi pour une estimation par le maximum de vraisemblance.

De tels programmes ont un grand intérêt pour la formulation de mécanismes théoriques, et pour les processus chimiques et biologiques. L'expérimentateur peut commencer avec un modèle théorique qu'il estime être presque certaine-

ment inadéquat, ou qui, du moins, décrit grossièrement le processus en jeu. Habituellement, l'ajustement de ce premier modèle se révélera inadéquat. L'analyse des résidus est alors d'un grand intérêt pour indiquer les directions possibles dans lesquelles l'idée du mécanisme a besoin d'être modifiée. D'autres séries de calculs peuvent être alors exécutées en utilisant progressivement des modèles modifiés, jusqu'à ce qu'on trouve un ou plusieurs mécanismes plausibles, qui décrivent de façon appropriée les données déjà disponibles. Bien entendu, un tel processus n'a pas besoin de conduire au modèle correct, mais seulement à une formulation, ou à des formulations qui sont plausibles sur les données déjà collectées.

Il est alors nécessaire de donner à la calculatrice une nouvelle tâche, celle de concevoir des expérimentations, par exemple des expérimentations qui discrimineront le mieux entre les divers modèles plausibles.

Le Dr. Lucas et moi-même nous sommes lancés dans ce problème, et nous sommes arrivés à des plans intéressants pour des cas simples, que nous espérons publier sous peu. La facilité avec laquelle, en utilisant de tels processus, il peut être possible d'évaluer la plausibilité de mécanismes possibles, et ensuite d'obtenir confirmation par les expériences conçues par la calculatrice, doit fournir un outil puissant pour la recherche scientifique.

Toutefois, je ne suis pas d'accord avec le Dr. Tocher sur la solution qu'il propose au problème. On peut souhaiter une mémoire à niveau unique pour la programmation, mais l'idée d'avoir une machine à mémoire à deux ou n niveaux, avec un moyen automatique de la convertir, dans la mesure où cela intéresse le programmeur, en une machine à niveau unique n'est pas bonne, je pense, parce qu'on est certain de gaspiller le temps machine. Il y a des difficultés à programmer une machine à n niveaux, spécialement dans cette sorte de travail impliquant des choses telles que la taille des blocs d'informations dont on a besoin pour calculer, comparée à la taille des blocs en mémoire, ce qui signifie que si, en fait, une machine à n niveaux est employée comme machine à niveau unique, elle consacra à peu près le maximum de son temps à extraire les données mises en mémoire.

Je suis aussi en désaccord avec lui sur ce point de vue fataliste, que nous sommes condamnés à des systèmes à deux niveaux ou plus pour une longue période à venir. Après tout, aux Etats-Unis, ils ne sont pas restés attachés à cela. Il y existe des choses du genre des mémoires à accès immédiat ; elles sont réalisées aux Etats-Unis pour contenir des milliers de mots - certainement autant que la machine 401. Je voudrais demander aux auteurs comment ils estiment que cela influencerait leur travail s'ils avaient une telle machine.

Je suis d'accord pour regretter que les auteurs n'aient pas trouvé l'occasion de dire un mot sur le type de machine qui est le plus valable pour cette sorte de travail. Dans leurs conclusions, ils disent qu'"un programme rédigé pour une machine peut habituellement être transposé pour son emploi sur une autre machine...". J'estime que le travail que cela implique est extrêmement important, bien qu'il soit parfaitement exact que le genre de suggestions et d'astuces qui ont été si précieusement données dans cette communication puisse être mis d'une machine sur une autre. Je soupçonne fortement que la dernière affirmation: "Pour cette raison, l'effort de programmation que nous avons effectué sur la 401 ne sera nullement perdu si nous changeons de machine" est une suggestion à leurs supérieurs financiers que c'est un bon argument pour en avoir une autre.

Je voudrais demander aux auteurs si, dans cette sorte de travail statistique, il pourrait être utile d'avoir un moyen de sortie graphique de la machine -

non nécessairement connecté à la machine, mais un moyen d'obtenir automatiquement des diagrammes à partir de sa sortie.

Les auteurs ont ultérieurement répondu par écrit, comme suit :

Nous tenons à remercier les personnes qui ont contribué à la discussion par leurs remarques constructives et stimulantes. Voici nos commentaires :

Premièrement, à l'égard de la machine elle-même, le Dr. Tocher a suggéré que notre machine était "probablement une des machines les moins appropriées au monde au calcul statistique". Nous ne souscrivons pas à cette opinion. Bien que nous n'ayions pas d'expérience directe de l'emploi d'autres machines, il nous semble que la machine elle-même est absolument comparable à d'autres de sa taille actuellement sur le marché, et nettement supérieure à certaines.

En passant, la machine a 1 024 mots de mémoire à accès intermédiaire, et non 128 comme l'a dit le Dr. Tocher. C'est en fait une machine à trois niveaux de mémoire : une mémoire à accès immédiat très petite (trois registres d'un mot chacun, en plus des deux registres nécessaires pour le travail en longueur ordinaire de mot et en longueur double), un niveau intermédiaire de 896 mots sur 7 pistes commandées entièrement par commutation électronique, et un troisième niveau de 2 048 mots sur 16 pistes commandées par commutation à relais. Pour les registres à accès immédiat, ou pour un de ces registres et un emplacement de mémoire, la machine opère comme une machine à $(2 + 1)$ adresses, c'est à dire que chaque instruction spécifie une source, une destination et l'adresse de l'instruction suivante.

Pour considérer la convenance d'une machine, il y a deux aspects distincts. L'un est la facilité de programmation ; l'autre la vitesse opératoire. A ces deux points de vue, la 401 serait considérablement améliorée si elle avait une mémoire à accès immédiat plus grande. Avec la machine actuelle, il faut exercer beaucoup d'ingéniosité pour faire le meilleur usage des trois registres à accès immédiat. D'autre part, nous trouvons extrêmement intéressantes les possibilités de modification B associées à ces registres. Une longueur de piste plus courte, mettons de 64 mots au lieu de 128, avec une augmentation correspondante du nombre de pistes, réduirait en outre substantiellement les durées d'attente. Il y a lieu de mentionner que la 402, a 15 registres à accès immédiat, donc 5 opèrent comme lignes B.

Nous sommes d'accord avec le Dr. Tocher, et en contradiction avec le Dr. Box en ce qui concerne le nombre de niveaux de mémoire dans la machine. Bien entendu, on peut avoir une mémoire à accès immédiat de taille considérable ; mais plusieurs types d'exploitation de données nécessitent une grande capacité de mémoire, si on veut les traiter efficacement. A vrai dire, la tendance générale du développement moderne est de prévoir différents niveaux de mémoire qui sont tous aisément accessibles au moyen du programme, mais dont le temps d'accès varie considérablement. Aussi, plusieurs machines ont une grande mémoire à accès immédiat, secondée par un tambour magnétique, et de plus secondée par la bande magnétique.

Le Dr. Tocher suggère qu'on peut concevoir une machine que l'on pourrait programmer comme si elle était à niveau unique, bien qu'elle soit réellement à deux niveaux. Une autre solution, qui ne demande pas de dispositifs supplémentaires (avec l'accroissement consécutif du coût et du risque de fautes), est l'em-

ploi de sous-programmes. Nous avons adopté ce système sous forme embryonnaire dans nos programmes d'analyse des expérimentations, en ce que la majorité d'entre eux contiennent un sous-programme provoquant la commutation à la piste à relais contenant la variable aléatoire en cours d'analyse.

Le Dr. Tocher a peut-être mal interprété certaines remarques fortuites sur la rédaction de programmes plus généraux. Tout ce qui est envisagé pour le moment, c'est le remplacement de nos programmes distincts pour les blocs avec répartition au hasard, les carrés latins et les blocs avec répartition au hasard à parcelles subdivisées, par un programme unique couvrant toutes ces éventualités. Un autre programme actuellement rédigé traitera un plan factoriel $1 \times m \times n \dots$, et couvrira donc les types de plans 3^n et $2^p \times 3^q$. Il sera fondamentalement semblable au programme 2^n , ainsi que le programme de répétition unique 3^3 , qui est un plan très courant. En fait, nous doutons qu'il soit valable de pousser la généralisation beaucoup plus loin qu'on ne l'a fait dans le programme 2^n , en partie parce qu'un programme général, à moins d'être simplement un amalgame de programmes spéciaux, tend à être substantiellement plus lent dans l'exécution des types d'analyse plus simples, et en partie à cause des formes différentes de sortie, de présentation et de contenu requises par les différents plans. Ainsi, pour des blocs avec répartition au hasard et des carrés latins, on n'a besoin que d'une simple table de moyennes ; tandis que pour des plans factoriels, une variété de tables à une, deux et plusieurs entrées est appropriée. Un programme général qui traite convenablement tous les types d'analyse nécessaires et les différentes formes de sortie nécessiterait, en fait, plus d'espace de mémoire qu'on n'en dispose ; même si l'on disposait de cet espace de mémoire, le temps pris par la lecture du programme serait à prendre en considération. Ce dernier facteur prendra évidemment moins d'importance quand on disposera d'une mémoire sur bande magnétique, mais il y aura également une petite difficulté à passer d'un programme à un autre.

Mr. Tizard conteste notre opinion qu'un programme rédigé par une machine peut se transposer sans grande difficulté pour l'utiliser sur une autre machine. Nous pouvons l'assurer que nous n'envisageons pas le cas où nous aurions une autre machine. C'était l'expression sincère de notre opinion, résultant de la quantité de temps consacré aux diverses opérations nécessaires à l'établissement d'un programme. A vrai dire, si nous recevions une autre machine, nous rédigerions un programme de transposition, de sorte que les programmes peu utilisés puissent être transposés automatiquement de façon à fonctionner sur la nouvelle machine. Bien entendu, une telle transposition automatique aurait pour résultat un programme plus lent que celui qu'on pourrait rédiger en reprenant le travail au début, mais elle serait indubitablement appropriée au cas où le volume du travail est faible.

Nous nous permettons d'expliquer que nous n'avons pas reçu notre machine libre. Bien qu'elle ne soit pas en ce moment lourdement chargée, - elle fonctionne en moyenne environ dix heures par jour - nous nous sommes beaucoup préoccupés d'accélérer son fonctionnement, et, à cet égard, la vitesse des dispositifs d'entrée et de sortie est indubitablement le goulot d'étranglement actuel dans la plupart des problèmes statistiques que nous traitons. Notre perforatrice de "bande à grande vitesse" n'est ainsi qualifiée que par comparaison avec celle qui l'a précédée, et non au sens absolu ; nous voulons mettre en relief l'importance d'avoir l'équipement auxiliaire voulu et la nécessité d'améliorer l'équipement actuellement présenté sur le marché, au point de vue à la fois de la vitesse et de la sécurité.

En réponse au Dr. Box, nous n'avons jusqu'ici pas éprouvé le besoin d'une forme graphique de sortie. Nous avons l'intention de programmer la construction

de diagrammes par points, mais cela s'exécutera très bien sur notre équipement existant.

Un certain nombre d'orateurs ont posé des questions sur notre politique à l'égard des vérifications. Le Dr. Vajda a mal interprété nos remarques. A chaque vérification, ou le programme est tel que la section appropriée du programme est répétée, ou le programme donne instruction à la calculatrice de continuer. Certaines vérifications suivent la première ligne d'action, d'autres la seconde, mais ces deux lignes d'action n'existent pas simultanément dans la même vérification.

Les vérifications peuvent être défailtantes pour un certain nombre de raisons : fautes de machines, fautes de programme ; données incorrectement perforées. Evidemment, si une donnée est incorrectement perforée et qu'on puisse le détecter, le fonctionnement sera stoppé immédiatement. Dans le cas d'une faute passagère de la machine, recalculer la partie erronée du calcul (si c'est possible) peut se trouver tout ce qui est nécessaire, dans le cas d'une faute plus persistante, un autre calcul est nécessaire pour établir le fait. Dans les cas de fautes de programme et de données inappropriées, il est souvent utile d'avoir un relevé du calcul complet pour mieux diagnostiquer le trouble, le diagnostic de ces cas demande du temps et ne peut se faire sur le champ. Bien entendu, on ne permet pas à la machine de continuer indéfiniment un long calcul quand on sait que, par suite d'erreurs dans une partie antérieure, les parties qui suivent sont sans valeur.

Indépendamment de vérifications arithmétiques exactes, dans lesquelles nous demandons habituellement que la machine refasse le calcul s'il y a une défaillance, puisque c'est de manière précise attribuable à la défaillance de la machine, nos programmes d'analyse des expérimentations comportent un certain nombre de vérifications approximatives, et, dans ces vérifications, la grandeur des divergences dépend habituellement beaucoup de la grandeur des données. Les programmes sont habituellement rédigés de sorte qu'avec les données normalement présentées, la grandeur permise de la divergence ne soit dépassée, et cela est indiqué par l'impression d'un point. Si ce n'est pas le cas, la divergence réelle est imprimée. Dans ces vérifications, la machine reçoit instruction de continuer le calcul, car, dans beaucoup de cas, l'impression d'une divergence n'indique pas une erreur réelle. La raison pour laquelle nous imprimons la divergence seulement quand elle dépasse une grandeur permise, est que cela attire l'attention sur les cas où un examen plus attentif est nécessaire. L'indication imprimée du succès ou de la défaillance d'une vérification est, pensons-nous, désirable, même si des lampes de signalisation sont prévues.

Par conséquent, on voit que la décision sur le type d'action à entreprendre quand une vérification est défailtante est incorporée dans le programme. La 401 n'est pas équipée de lampes de signalisation, et les possibilités d'avertissement ne sont pas étendues, de sorte que les opérateurs expérimentés ont de la difficulté quand la machine s'arrête, à décider avec précision ce qui ne va pas. C'est pour cette raison que nous essayons de nous assurer que, si la machine doit abandonner un problème, elle en fournit une évidence suffisante à la personne qui examine les résultats pour entreprendre une action appropriée avant de passer le problème à nouveau.

En réponse au Dr. Page, nous dirons que le fait que nous pouvons analyser certains types de plans sur la machine, et non d'autres, n'a pas eu d'influence appréciable sur les types de plans actuellement utilisés. Nous avons, il est vrai, fait de notre mieux pour éviter cette éventualité, car nous pensons que ce serait un retour en arrière si des plans moins efficaces pour le but cherché étaient a-

adoptés juste parce que quelqu'un se trouve avoir rédigé un programme pour leur analyse. A vrai dire, nous pressentons que les effets à long terme de la machine se produiront au sens opposé. Il y a eu pendant longtemps une tendance, dans les groupements agricoles, à éviter les plans factoriels avec confusion contenant des nombres mêlés de niveaux par exemple $3 \times 3 \times 2$ et $3 \times 2 \times 2$ en blocs de 6 parcelles, bien qu'ils soient remarquablement appropriés à plusieurs objets, simplement parce que leur analyse statistique est incontestablement plus pénible que celle des plans factoriels à nombre identique de niveaux, par exemple 3^3 en blocs de 9 parcelles. Comme mentionné précédemment, nous rédigeons en ce moment un programme pour l'analyse de tels plans, et nous espérons que son existence entraînera leur emploi plus étendu.

Cela a, en outre, un rapport avec la question du Professeur Barnard, concernant l'effet que l'emploi des calculatrices électroniques peut avoir sur l'enseignement de la statistique. Nous croyons que des modifications considérables seront nécessaires. En premier lieu, les étudiants devront être instruits de la manière dont les calculatrices électroniques peuvent être intéressantes dans les travaux statistiques. Deuxièmement, plusieurs méthodes statistiques, par exemple, les travaux avec variables aléatoires multiples, à présent peu employées par suite du fardeau excessif des calculs, prendront une importance largement accrue, et leur application correcte aura, par conséquent, à être enseignée. D'autre part, il sera possible de réduire substantiellement le montant de l'instruction donnée à présent sur plusieurs des types les plus élaborés de travaux numériques, qui peuvent être mieux traités sur une calculatrice. Ainsi, par exemple, il peut paraître ne plus être nécessaire de consacrer beaucoup de temps aux points plus délicats des calculs numériques nécessaires pour l'inversion efficace des matrices, si, comme ce sera bientôt le cas, seuls ceux qui ont la marotte du calcul inverseront manuellement une matrice d'une taille quelconque. De même, dans le domaine de l'analyse expérimentale, seul le spécialiste aura besoin de se préoccuper du processus numérique nécessaire à l'analyse des types plus complexes d'expérimentation avec confusion, ou des complications qu'entraîne le traitement des parcelles manquantes dans les plans complexes.

Le Dr. Page a suggéré qu'en statistique, il est meilleur que chacun exécute sa propre programmation. Nous croyons qu'il va trop loin. Un programmeur doit, bien entendu, être familiarisé avec les problèmes qu'il traite, mais il a aussi beaucoup à apprendre en ce qui concerne la technique générale de la programmation et les particularités spéciales de la machine dont il s'occupe. La programmation des problèmes statistiques sera certainement assurée par des statisticiens, mais cela ne signifie pas que chaque statisticien ayant un problème pour lequel une calculatrice électronique est nécessaire doive nécessairement le programmer lui-même. D'après notre expérience, c'est le programmeur expert qui tend à élargir les applications de la calculatrice, et qui tend également à rédiger ses programmes sous forme générale ; plutôt que sous une forme qui produira seulement les résultats immédiatement nécessaires. C'est en partie de l'autodéfense, puisqu'autrement il risque d'être exposé à des demandes répétées de modifications.