Yates, F., Gower, J. C. and Simpson, H. R. 1963. A specialized autocode for the analysis of replicated experiments. *The Computer Journal.* 5 (4), pp. 313-319.

A specialized autocode for the analysis of replicated experiments

By F. Yates, J. C. Gower and H. R. Simpson

The paper describes a general program, written for the Elliott 401, for the analysis of orthogonal or nearly orthogonal data, such as arise from replicated experiments. This is in essence a specialized autocode for performing on tables the types of operation required in such analyses, and is similar to Part 2 of our general survey program. Modifications and extensions planned for the Orion are briefly discussed.

The utility of specialized autocodes was stressed by one of us in the 1961 Presidential Address to The British Computer Society (Yates, 1961). A general program for the analysis of surveys, which is in essence a specialized autocode, was described by Yates and Simpson (1960, 1961). The present paper gives an outline of a somewhat similar specialized autocode for the analysis of orthogonal or nearly orthogonal data, such as arise from properly planned experiments.

Replicated experiments are of very varying degrees of complexity. The general method of analysis is that known as the analysis of variance; for each type of design there is an appropriate form of analysis which varies in complexity with the complexity of the design. The simpler types of analysis can be easily performed on desk calculators, but even in these simpler types incomplete data give rise to tiresome complications. Moreover, to study the relations between variates, or to use supplementary observations to improve accuracy, analyses of covariance are necessary, and these, though similar in form, are much more tiresome than the corresponding analyses of variance, as they involve sums of products as well as sums of squares.

At Rothamsted we are much concerned with the analysis of experiments, and have for several years used our computer for the analysis of the more commonly occurring types of design. The increasing number of analyses we are asked to perform indicates that this provides a useful service:

	1955	1956	1957	1958	1959	1960	1961
Experiments	419	682	1,253	1,664	2,649	3,687*	2,862
Variate and covariance analyses	834	1,701	5,041	6,260	11,102	11,147	15,184
Analyses per experiment	2·0	2·5	4·0	3·8	4·2	3·0	5·3

* Including a large batch of very simple one-variate experiments.

It will be noted that the number of analyses per experiment has risen from 2·0 in 1955 to 5·3 in 1961; the ability to get analyses done has encouraged research workers to make more use of the subsidiary data which they collect in their experiments.

These analyses have been effected by a few programs, each of which covers a relatively limited set of designs of a given type, e.g. 2^n designs in randomized blocks or quasi-Latin squares. We were still, at the beginning of 1962, unable to undertake the analysis of some of the less commonly used types of design, and the analysis of others required special modifications which were both time-consuming and tricky.

The success of our general survey program in dealing with the analysis of surveys of all types encouraged us to construct a similar program for the analysis of experiments. In essence our general experiments program is a specialized autocode for performing varied operations on tables. The operations are similar to those of Part 2 of the general survey program, but certain additional operations are required and also facilities for the modification of the autocode instructions analogous to B-modification. Because of the small store and slow speed of the Elliott 401, and also to simplify programming, no attempt was made to write a very refined program. In particular the specification of the parameters required for the various autocode instructions is in a form dictated by machine convenience; this could be improved even for the 401 by constructing a translation routine which would enable the user to write the instructions in the form most convenient to him. However, as the 401 will shortly be replaced by a Ferranti Orion there seemed little point in this. One of the main uses of the 401 program has been to provide a mock-up for a similar but more sophisticated Orion program.

General problems in construction of specialized autocodes

The first step in the construction of a specialized autocode is to determine what types of operation shall be covered by the autocode instructions. For the present program this was done by taking typical examples of actual experimental designs, and building up a list of the operations which were necessary for their analysis. The specification of the operations was modified and extended as the work progressed.

When a list of operations and their exact specification has been determined, or concurrently, machine strategy must be decided. This involves decisions on the form of the autocode instructions, parameters, storage, whether interpreter or compiler techniques are to be used, etc. Here we followed fairly closely the methods adopted for Part 2 of the general survey program; indeed parts of this program were used with little alteration.

Operations required in the analysis of experiments

A simple example is provided by randomized blocks. In a randomized block experiment for P treatments, each with Q replicates, the PQ experimental units are divided into Q blocks, the units in each block being as homogeneous as possible, and the P treatments are assigned at random to the units in each block independently. In a Latin square there is a double restriction, each of P treatments being assigned once and once only to each row and each column of a $P \times P$ arrangement of P^2 experimental units.

The observed values and the block and treatment means in a randomized block experiment may be set out in a two-way table as in Table 1. If the sums of squares of the body of the table and the margins, and the square of the general mean, are denoted by [1], [2], [3], [4], as indicated, the analysis of variance takes the form shown in Table 2.

The results commonly required by the experimenter are the treatment means, often with a conversion to standard units, the analysis of variance, and the standard error of the treatment means (derived from the error mean square of the analysis of variance). The residuals, i.e. what is left after allowing for block and treatment effects, are also sometimes of interest; excessively large residuals, for example, indicate aberrant values.

Residuals are troublesome to compute on desk calculators, but can be easily provided by electronic computers. In a randomized block experiment the residuals are given by

$$e_{pq} = y_{pq} - \bar{y}_{pQ} - \bar{y}_{Pq} + \bar{y}_{PQ}.$$

We have also

$$\Sigma(e_{pq}^2) = \text{error sum of squares},$$

which provides a useful check on the arithmetic and also, in complex experiments, on the method of analysis. Furthermore, residuals form the basis of a convenient iterative method for providing estimates for any observations that are missing. In this method approximate values are assigned to the missing observations; the residuals for these observations are then calculated and subtracted from the approximate values to give second approximations, and so on. The estimates so obtained can then be used in the final analysis, with suitable adjustment of the error degrees of freedom.

Many experiments have additional complexities. In factorial designs all combinations of several treatment factors are included, and the table of treatment means requires arrangement in multi-way form by factors, with marginal means. The treatment term in the analysis of variance also requires partition, in an analogous manner to the partition of the total sum of squares in a randomized block experiment (see Tables 1 and 2 above). Complete replicates of a factorial design may be split into two or more blocks to give partial or total confounding of certain degrees of freedom. There may be only a single replicate of a factorial design (or indeed only part of a complete replicate), in which case the error has to be estimated from high-order interactions. Experimental units may be split for subsidiary treatments (split plots), giving rise to two estimates of error. Designs such as balanced and partially balanced incomplete blocks, lattice squares, changeover designs, etc., and long-term experiments, introduce further complexities.

Table 1

Observed values and means, randomized block experiment

		Treatment (p) 0 1 2 3 . . . ($P-1$)	Mean (P)
	0		
	1		
Blocks (q)	2	y_{pq}[1]	$\bar{y}_{Pq}$[2]
	. .		
	$Q-1$		
Mean (Q)		$\bar{y}_{pQ}$[3]	$\bar{y}_{PQ}$[4]

Table 2

Analysis of variance, randomized block experiment

	DEGREES OF FREEDOM	SUMS OF SQUARES	MEAN SQUARES
Blocks	$Q-1$	$P.[2] - PQ.[4]$	
Treatments	$P-1$	$Q.[3] - PQ.[4]$	$SS/d.f.$
Error	$(P-1)(Q-1)$	By difference	
Total	$PQ-1$	$[1] - PQ.[4]$	

To study relations between two or more variates an analysis of covariance between each pair of variates is required, in addition to the variance analyses. The covariance analysis takes the same form as the variance analyses except that sums of squares are replaced by the corresponding sums of products. The variance–covariance matrix provided by the error components of these analyses can then be used to calculate regression coefficients, which can in turn be used, for example, to calculate tables of adjusted treatment means from the tables of treatment means for each variate.

The observations are usually recorded in the (numbered) order in which the units occur in the experiment; thus in a randomized block experiment the observations will be by blocks with random order within blocks, in a Latin square by rows and columns. Variates observed at different times are often recorded separately. The simplest form of punching is therefore to punch the data for each variate separately in the numbered order, and also the treatment code (which in a factorial experiment will be a multi-factor code) in the same order.

With the data stored in this order, operations must be included which (a) enable block means, or row and column means, to be computed and stored in new tables, and (*b*) form the treatment means and store these in a table classified by the treatment factors. Operation (*a*) is the same as that of taking marginal means, except that the means are written in new tables. Operation (*b*) is one of a class of what we term *code operations*.

In code operations the code (or a collated part of it, if packed) gives the levels of the factors of the cells of a table which can thus be combined with other tables which are stored in the standard order for the relevant factors. To form treatment means, for example, the table for treatment means is first set to zero, and each value of the variate is taken in turn and added into the cell given by the corresponding value in the code table; the resultant table is then divided by the number of replicates. If some treatments have more replicates than others a table of numbers of replicates is constructed by a similar operation, adding in $+1$ instead of the variate value, and the treatment-total table is then divided by the replicate-number table.

The first of these operations is also required to evaluate partially confounded effects. For example, in a $3 \times 3 \times 3$ factorial experiment in blocks of nine plots four codes, which can be packed, give the four partitions of the treatment combinations corresponding to the four sets of two degrees of freedom for possiblyconfounded three-factor interactions; the corresponding totals are obtained from the treatment means table. By performing a preliminary analysis on a set of pseudo-values so chosen that effects and interactions have standard values, the amount of confounding can be evaluated.

Two other types of code operation are required. For the computation of residuals in a randomized block experiment, for example, the block means and treatment means have to be subtracted from the variate values. The block means can be subtracted by the ordinary operation for the subtraction of one table from another of different order. The subtraction of the treatment means requires the identification of the appropriate means in the table of treatment means by the treatment code. The other operation is similar, with addition instead of subtraction, and is required in calculating adjusted treatment means when there is partial confounding, and also for arranging pseudo-values in the appropriate random order for the preliminary analysis.

Table index and storage allocation

The general scheme for manipulating the tables which are required in the computations is similar to that of the general survey program (see Yates and Simpson, 1960, 1961). The specification of a table is somewhat more general than in the survey program in that in addition to the classification variates (*factors*), the ranges of the factors included in the table are also specified. Thus any table can be specified as containing space for any or all the margins, or indeed with more limited dimensions. The factor numbers are packed in a single word, and their ranges are similarly packed in a separate word. (This range word is also used in the survey program, but since all tables there contain margins it is computed by the machine.) A single index number (S) is assigned to the factor set (FS) and dimension set (DS), and consequently two tables with the same factors but different dimensions require a different S.

Each table is referred to by number, and there is a table index containing the S of the table, assigned initially by the programmer, and the address of the first cell, assigned automatically with certain exceptions that need not be listed here. This differs from the survey program in that the assignment of the S ($=$ the classification set number, C) is there automatic for derived tables unless a reduction of order is required. The store of the 401 was recently extended by the addition of a further 8 tracks, giving 3,968 locations in all; 1,408 locations are available as working store for tables, etc., up to 256 of these being allocatable to single values (conventionally treated as components of Table 1 and coded 1.x). In addition any cell of any table may be referred to in single-value operations. The remainder of the store is occupied by the program. Without the extension of the store the program would clearly have been impractical.

Autocode instructions

A conspectus of the autocode instructions is given in Table 3. Each instruction is specified by a string of up to 18 parameters, 11 of which relate to the operation, the remaining 7 being used for control purposes, as described below. Although this implies a potentially complicated system, many of the parameters usually have the value 0, and thus only require the attention of the programmer in exceptional circumstances. Details of the auxiliary parameters are not given in Table 3, but their nature may be inferred from the explanatory remarks in the table.

The instructions are normally read and obeyed one at a time from the paper tape, but provision is made for the storage of loops of instructions (see below).

The contents of any location in the single-value store may be used to replace a parameter in an instruction, or a factor number or dimension in FS or DS. This extends the scope of standard sets of instructions, e.g. the analysis of a $P \times Q \times R$ factorial experiment can be covered by a single set of instructions, whatever the values of P, Q, R; and is useful in other ways.

The 401 is a fixed-point machine, and consequently control of scaling is provided where necessary in a similar manner to the survey program. We have, however, avoided the need for scaling in most circumstances by adopting standard locations for the binary point, which can accommodate the usual types of experimental data. Thus linear functions of the data (regarded as integers) are carried with 10 binary places, ratios (e.g. regression coefficients) with 20 binary places, and sums of squares and products with 7 binary places.

The address system and mechanism for scanning tables and generating addresses is identical with that of the survey program (see Gower, 1962), except that tables can contain up to six factors, with maximum numbers of

Table 3

Conspectus of autocode instructions

Table operations

1. $D = A + B$	6. $D = -kA + B$
2. $D = -A + B$	7. $D = k + B$
3. $D = A \times B$	8. $D = \sqrt{(kA/k')}$
4. $D = A/B$	9. $D =$ Collated part of A
5. $D = kA + B$	10. Spare

If $B = 0$, $D = A$, etc. If $B = -1$, the indicated margin of A is read for B. D, A, B need not be of the same order but one must contain all the factors in either of the other two. A or B may be replaced by a single value $1.x$ from Table 1. Partial scans can be performed.

Single value operations

11–20. As 1–10, i.e. $D.d = A.a + B.b$, etc. The result may be tested in various ways.

Code operations

21. $D_F = D_F + (1)_C$	23. $D_C = A_F + B_C$
22. $D_F = D_F + B_C$	24. $D_C = -A_F + B_C$

Tables with suffix F contain the factors of the code; tables with suffix C have a one-to-one correspondence (apart from possible margins) with the code (Table C). Table C may contain multiple (packed) codes. In 21 and 22, D is set to zero at start. 21 gives counts, 22 gives means or totals. 24 can be limited to entries $D.x$ and $B.x$, where the x values are listed in a table X.

Sums of squares or products

27. $D.d = SS(A)$ 28. $D.d = SP(A, B)$

SSP for the body of a table (or pair of tables) or a specified margin. Also complete SSP for the effects and interactions of a multi-factor table. The components of the multipliers for SSP which depend on the dimensions of the A, B tables are supplied automatically.

Marginal totals or means

26. Over a specified factor of A in D (space for margin not provided in A).

30. Over one or more specified factors of D in D.

Miscellaneous

25. $D.x = D.x - B.x$ for all values of x contained in Table X, with test of $SS(B.x)$.

29. Effects of nominated two-level factors of D in D.

34. Print of maximum, minimum and mean of values in D, with plant of mean.

35. Linear combination of values of A, starting at $A.a$, in $D.d$. (Up to 16 consecutive values, each added, subtracted or omitted.) Result may be tested.

0. Initial set-up for preliminary analysis, variate analysis, covariance analysis.

−1, 50. Stops.

Input

31. Numerical data. 32. Codes, etc. 33. Names.

In 31 the values can be replaced by their deviations from their mean, and the mean can be planted.

Output

40. Table.	42. Analysis of variance, etc.
41. Single value.	43. Missing values.

In 40, output may be restricted to parts of a table; the factors appertaining to the table or part of it are printed at the head of the table. 42 prints up to six one-way tables in columns, with side headings. Standard modes (for integers, values, values plus mean, squares, products) which give correct positioning of the decimal points, etc., can be used, or scaling, etc., can be specified.

levels (including margins, if present) of 31, 15, 15, 15, 15, 15.

The table operations of Table 3 are cell-by-cell operations similar in form to those of the survey program, but with the additional feature that parts of tables can be operated on. This is effected by (*a*) providing for the addition of quantities, specified in the instruction, to the reference addresses generated by the scan routine for each of the D, A, B tables separately, and (*b*) reducing the ranges of scan, which are normally those of the D table. The restriction that the A or B table must contain all the factors occurring in the D table has also been removed, and replaced by the restriction (for programming convenience) that any one of the three tables must contain all the factors occurring in either of the other two.

Provision has been made for printed comments, headings, etc. Any instruction (other than a stored instruction) may be preceded by a "comment" which is transferred direct to the output tape, and test instructions can be followed by a "comment" which is transferred if the test is satisfied. There is also a name index, which permits the insertion of special names in the comments and table headings of standard sets of instructions.

Monitoring facilities are provided whereby the result of any operation (table or single value) can be printed out in the required mode on completion of the operation. Control is by the hand-switches.

It will be noted that the preliminary conversions and functions of the data, that are required before the analysis proper, are taken care of by the table-operation instructions, and no special instructions are required.

Table 4

Randomized block experiment: factors and tables

Factors	NO.	LEVELS	*Tables*	NO.	FS	DS
Treatments	1	P	Treatment code	2	1, 2	P, Q
Blocks	2	Q	Observed values	3	1, 2	P, Q
A of V	3	4	Block means	4	2	Q
			Treatment means	5	1	$P + 1$
			Analysis of variance:			
			Degrees of freedom	6	3	4
			Sums of squares	7	3	4
			Mean squares	8	3	4

Table 5

Randomized block experiment: instructions

Nature of operation	No.	D	A	B	C	Other conditions
Check on design(*)	21	5	—	—	2	
Block means(*)	26	4	3	—	—	Over factor 1
Treatment means	22	5	—	3	2	
Mean of treatment means(*)	30	5	—	—	—	
Total SS	27	7.3	3	—	—	
Block SS	27	7.0	4	—	—	$\times P$,
Treatment SS	27	7.1	5	—	—	$\times Q$,
Error SS	35	7.2	7.0	—	—	subtr., subtr., omit, add.
Mean squares(*)	4	8	7	6	—	Integer division
Treatment S.E.(*)	18	1.4	8.2	—	—	$k = 1,\ k' = Q$
Deviations from block means	2	3	4	3	—	
Residuals(*)	24	3	5	3	2	
Residuals SS(*)	27	1.5	3	—	—	

A simple example

The need for and uses of the various types of instruction will be apparent from the outline already given of the various operations required in the analysis of experiments. The general procedure may be illustrated by the instructions required to deal with a single variate of a randomized block experiment, for which the form of analysis is given above.

The tables required, and *FS* and *DS*, are set out in Table 4. Factor 1 is used as a pseudo-factor for Tables 1 and 2. An extra cell is included in Table 5 to enable the general mean to be printed alongside the treatment means.

Table 5 gives the instructions required to furnish the block and treatment means, the analysis of variance, the standard error, and the residuals and their sum of squares. The first instruction gives a simple check on the design, i.e. that the number of replicates is the same for each treatment. It is assumed that the original values have been replaced by their deviations from the general mean on input. This reduces risk of overflow. The mean is added back where necessary in print operations. To save space, print instructions have been omitted; they follow instructions with asterisks. Storage has been economized by writing the residuals over the original yields.

In the analysis as shown the checks, e.g. that the residual *SS* equals the error *SS*, have to be made visually from the printed results. If desired, tests can be incorporated in the instructions, with suitable printed comment. It is also assumed that the degrees of freedom are supplied with the data; instructions can, of course, be included for their calculation from the basic parameters P and Q.

Stored loops of instructions

To economize storage the autocode instructions are read one by one from the tape. It would obviously be more convenient if all instructions were stored, and as a compromise provision is made to store loops of instructions, one loop at a time. When all the instructions of the loop have been obeyed control is returned to the beginning of the loop. Exit from the loop to the next instruction on the tape is made either (*a*) after the loop has been obeyed a specified number of times, or (*b*) by means of a test in some instruction in the loop.

The utility of stored loops is greatly increased by a loop modification facility, which increases the first six parameters of the instruction proper (*D*, *A*, *B*, and in single value operations *d*, *a*, *b*) by specified amounts ($\leqslant 13$) each time the stored instruction is obeyed or skipped.

The stored loop facility simplifies repeated iterations for the evaluation of missing values, and economizes instructions in repetitive processes such as the evaluation of the four sets of confounded degrees of freedom in a $3 \times 3 \times 3$ experiment.

Analysis of successive variates and covariance analyses

If only analyses of variance are required the analysis of a number of variates from the same experiment can be dealt with by completing the analysis of each variate before reading in the data for the next variate, using the same storage and same set of table numbers for each variate. If, however, covariance analyses are required it is necessary to preserve most of the tables relating to a particular variate until all the covariance analyses involving that variate have been completed. To avoid the necessity of writing different sets of instructions for each variate analysis and each covariance analysis, a system of modification has been provided in which the modifiers depend on the number of the variate being analysed, or, in the case of a covariance analysis, the numbers of the pair of variates entering into the analysis. In the case of a variate analysis each of the first six parameters of an instruction can be modified by the addition of (*a*) the variate number, or (*b*) a value corresponding to the variate number given in an initially supplied table. In the case of a covariance analysis the modification can be by either variate number, or by the tabular value corresponding to either variate number.

These modifications are called for automatically by a control system which advances the variate number by one each time a variate analysis is performed, and similarly advances a covariance analysis number each time a covariance analysis is performed. The covariance analysis number provides a reference to a table containing a list of the pairs of variates for which covariance analyses are required.

Standard sets of instructions

With these provisions it is possible to write standard sets of instructions to cover the analysis of all designs of a given type, e.g. $3 \times 3 \times 3$ designs in blocks of nine plots with more than one replicate. The instruction tape is divided into three sections (*a*) preliminary analysis, containing a suitable check on the design and a determination of the extent to which different sets of degrees of freedom are confounded, (*b*) variate analysis, including the evaluation of missing values, if any, and (*c*) covariance analysis. All that is required of the operator is to feed in the data at the correct points, and to present the section (*a*), (*b*), or (*c*) of the instruction tape required for the next analysis.

Standard sets of instructions are of course very necessary if any large volume of experimental analysis is to be efficiently dealt with. It would be intolerable to have to write a separate set of instructions for every experiment or even for every minor variant in a design. It is also important that standard sets of instructions should be so written that there is no need to specify details of the experiment, in particular the exact nature of the confounding, which cannot be readily determined by cursory inspection.

Experience with the program

It has been found that the 401 program is quite powerful, and is capable of dealing not only with very varied types of experimental analysis, but also with many miscellaneous problems requiring table manipulation. Thus, for example, it has been used for the analysis of uniformity-trial data in an investigation of the errors associated with plots of different shapes and sizes and with and without guard plants. It can provide a table of correlation coefficients from a table of crude sums of squares and products $S(x_i x_j)$ and sums $S(x_i)$ of *n* sets of values. And it can perform all the operations (except transpose) included in Part 2 of the survey program.

The instructions are, however, more tiresome to write than they should be, particularly if a standard set of instructions is required to cover all designs of a given type with provision for missing plots and covariance.

The program is also slow compared with the speed that can be attained with a specially written program limited to designs of a given type. The causes of this have not been analysed in detail, but the main ones are probably (*a*) the undisciplined use of the relay-switched tracks of the drum store for data and working space, (*b*) the additional time taken by the general scan routine, and (*c*) the use of means instead of totals. (The 401 has no built-in division, so that division is inevitably slow.)

Improvements for the Orion

The Orion program is now being written. There will be one general table-manipulation program, which will serve the purposes of the present experiments program and the survey program. It is too early yet for a full description, but the following notes will indicate the lines on which we are working.

There will be an improved and simpler notation for instructions and for the specification of tables, including permissive symbolic representation of factors, e.g. BLOCKS, *A*, *B*, *C*, and automatic compilation of *FS* and *DS* lists.

For certain purposes more comprehensive instructions would be of value, and we may adopt the compiler technique, using the machine to compile and store sets of basic autocode instructions from more comprehensive instructions in the user code.

A few further instructions will be added to increase versatility, such as extension of the double scan concept of marginal means to provide linear functions of rows, columns, etc. The full repertoire of matrix operations will also be included.

Provision will be made for the optional introduction of headings to rows and columns of tables, and for more flexible output generally.

Input facilities will be extended to provide for the facilities provided in Part 1 of the present survey program, i.e. for the input of data unit by unit, calculation of functions of the data, and compilation of tables.

In many ways machine strategy is simpler on Orion. All autocode instructions can be stored, thus permitting full exploitation of loops. Tables will be held in the machine in standard factor order, the factors of a table being indicated by the digits of a single word. Transpositions will be required on output, and occasionally on input.

Scan has been reorganized, the counts for the different factor levels being held in separate words, with addition (not multiplication) for the generation of the addresses.

Tables will be held on the drum and worked on in the core store (4K), with the possible option of retaining tables in the core store for small problems or parts of larger problems.

The program will be stored permanently on magnetic tape (together with other programs), and standard sets of instructions will be similarly stored and called for by number.

References

GOWER, J. C. (1962). "The Handling of Multiway Tables on Computers," *The Computer Journal*, Vol. 4, p. 280.

YATES, F. (1962). "Computers in Research—Promise and Performance," *The Computer Journal*, Vol. 4, p. 273.

YATES, F., and SIMPSON, H. R. (1960). "A General Program for the Analysis of Surveys," *The Computer Journal*, Vol. 3, p. 136.

YATES, F., and SIMPSON, H. R. (1961). "The Analysis of Surveys: Processing and Printing the Basic Tables," *The Computer Journal*, Vol. 4, p. 20.

Summary of discussion

Mr. O. B. Chedzoy (*Bristol College of Science and Technology*): Is there any useful comparison in times between a specific program and the general program for replicated experiments, which could indicate a cost of flexibility in programming?

There appears to be a tendency for the amount of analyses per experiment to rise appreciably—how far is this due to a tendency to ask for information because it is easy to obtain, and what measures may be taken against it?

Mr. W. C. Henshaw (*Bristol College of Commerce*): (1) For how many hours per day, on the average, is the computer in use?

(2) What methods are employed to transmit information for analysis from out-stations to the computer? Are there any important problems to be solved in computing for out-stations?

Dr. F. Yates: In reply to Mr. Chedzoy, the only available comparison between times taken by specific programs and those taken by the general program for the same analyses, that for randomized blocks, suggests that the ratio is about 12 to 1, but the general program provides an improved layout. Reasons for the slowness of the general program are suggested in the paper, and should not be taken as typical of what will occur with modern machines with adequate immediate access storage. Had we started with this program we should undoubtedly have had to write special programs for the most commonly occurring types of analysis, in order to speed up the work as the load increased. With the Orion, we are starting with the general program and will only write special programs if it is apparent that useful economies of time will result. I would stress, however, that flexibility is essential if a research department is to provide the service that is required of it. Machine costs are secondary.

It is of course true that certain workers tend to ask for unnecessary analyses. We have recently instituted a job request system by which requests which involve more than five hours of machine time are reviewed, and this has resulted in the elimination of a considerable amount of work which we think is likely to prove unprofitable. On the other hand, it is sometimes necessary to do what we believe are useless computations in order to convince those who ask for them that they are in fact useless. It is also possible to be wrong in one's judgement and therefore too strict a censorship of this kind should not be practised.

In reply to Mr. Henshaw, in 1961 the computer was in operation for 3,688 hours, equivalent to 14 hours per day on a 5-day week basis, and to 186% of normal laboratory hours. The sub-division of this time was: 72% for productive work, 12% for program development, and 16% for maintenance, etc.

For the 401 practically all the experimental data are sent in manuscript and transferred to paper tape at Rothamsted. Large bodies of data, such as occur in survey work, are transmitted by means of punched cards, the preparation of which is the responsibility of the institute requesting the work. For the Orion, however, with its much greater capacity, we are encouraging institutes who expect to make considerable use of the machine to install their own equipment for tape preparation. They will also be expected to do much more in the way of specifying the exact analyses required, using specialized autocode instructions, or for special problems writing their own programs in Ferranti autocode. In the case of the general experiment program, standard sets of instructions will be available for the standard designs so that the specification of the analyses required for such designs will be extremely simple.