Experience in using a Deuce Computer for the Family Expenditure Survey

by Philip Redfern

Summary: A talk, given to The British Computer Society in London on 4 May 1959, reviewed some of the lessons learnt in using a computer for a statistical survey. After a short description of the origin of the survey, and of its statistical aspects, an account was given of the computer experience under five headings:

The choice of computer
The computer input and output
The program and computer operation
Programming and programmers
Costs.

The text below has since been provided by the author. The experience was gained and the talk prepared, before a significant amount of work on British automatic coding systems had been published.

THE SURVEY

For purposes of economic and social policy, the Government requires information on the expenditure and income pattern of consumers—that is of private individuals and households. There are several ways of getting such expenditure information: one can collect statistics of retail sales of such items as clothing, radios, and so on; a disadvantage of this method is that expenditure cannot normally be analysed between different groups of consumers, e.g. different income groups. Alternatively, one can approach a sample of consumers and ask them to give details of their expenditure during a period of time. Such family budget surveys were carried out by the Ministry of Labour before the last war in 1938, and again in 1953, in connection with the revision of the cost-of-living index.

More recently the Ministry of Labour, in consultation with the Central Statistical Office and other departments, started an annual sample survey of family budgets, the first being in 1957. In each year about 5,000 households located throughout Great Britain were approached, and they were invited to give details of income and expenditure for a two-week period. The surveys were spread over the year in order to detect seasonal patterns.

The selection of the sample and the actual collection of information was undertaken by the Social Survey, a Government organization coming under the Central Office of Information. Their field-workers collected some material (e.g. on the composition of the household) by means of interview, but most of the expenditure information was recorded by the responding households on questionnaires. Each member of a household was asked to enter his or her daily expenditure throughout the fortnight period on these forms. For the most part, the entries on the forms were in the order in which the expenses were incurred, and the descriptions of expenses were in the respondent's own words.

Of the 5,000 households approached each year, about 3,000 gave the full information asked of them, and each member of the co-operating households thereby earned £1 as some reward for his trouble.

The descriptions of items of expenditure and income given on the forms were coded by the Social Survey under about 330 headings, examples of which are

129	Optical goods
130	Photographic goods
131	Sports goods
132	Leather goods
133	Jewellery.

An example of an income item is

310 Income from sub-letting.

All the data collected were punched by the Ministry of Labour into Powers 65-column punched cards, because the analyses required by the Ministry were being prepared on its own orthodox punched-card equipment.

THE STATISTICAL ANALYSIS

Below are two examples of the kind of statistical analysis required by the Central Statistical Office. Before proceeding to these examples, it is necessary to explain that households have been classified in three ways:

(*a*) Firstly, by household composition, e.g. single persons have been distinguished from married couples with no children, which have again been distinguished from married couples with one child, and so on. Fifteen household compositions have been distinguished in all.

(*b*) Secondly, by social class: two classes have been distinguished on the basis of the occupation of the principal wage-earner. A household in which the principal wage-earner was, say, an actuary would

TABLE 1

EXPENDITURE OF HOUSEHOLDS CONSISTING OF A MARRIED COUPLE AND 1 CHILD

Items of expenditure (code numbers)	*Income of household (shillings per week)*						
	Under 75/– per week	*75/– and under 100/– per week*				*1,000/– per week and over*	*All incomes*
199–201, 238–240, 257–258.				average expenditure; standard deviation of expenditure			
205, 241.							
. .							
. .							
. .							
. .							
9–59, 203.							
. .							

TABLE 2

NUMBER OF HOUSEHOLDS CONSISTING OF A MARRIED COUPLE WITH NO CHILDREN, CLASSIFIED BY GROSS INCOME BEFORE TAX AND BY NET INCOME AFTER PAYING TAXES AND RECEIVING BENEFITS

Net income after taxes and benefits	*Gross income before tax*						
	Under 75/– per week	*75/– and under 100/– per week*				*1,000/– per week and over*	*All incomes*
Under 75/–				Numbers of households			
75/– and under 100/–							
. .							
. .							
. .							
1,000/– and over							
All incomes							

have been coded as middle class, and a household in which the principal wage-earner was a farm worker would have been coded as working class.

(*c*) Thirdly, by income ranges: 14 income ranges have been distinguished.

In addition, it has been possible to classify households according to whether their returns related to expenditure in the first, second, third or fourth quarter of the year. And, as I have said, expenditure and income have been classified under some 330 headings. All this means that a particular item of expenditure or income recorded in the survey could be classified into 1 out of 15 × 2 × 14 × 4 × 330 cells.

The first example of the analysis required is a table similar to Table 1. The entries in the table would be firstly average expenditure per household, and secondly standard deviations* of expenditure. Note that each line of the table refers in general not to a single item of expenditure but to a *group* of items. Each such group has to be defined, both outside and inside the computer, by an index of item code numbers.

The second example of a tabulation required is what statisticians would call a two-way frequency distribution, in the form of Table 2.

The gross income before tax of each household is calculated by the computer as the aggregate of such of the 330 headings of expenditure/income as represent items of income from salaries, wages, interest, dividends, etc. The household's net income, after taking account

* Very briefly, for those who are not statisticians, the standard deviation of expenditure measures how far expenditure varies from household to household around the average—it is a measure of "dispersion."

of all taxes and benefits, is then obtained by (i) adding to gross income certain other of the 330 headings (e.g. national insurance benefits), and then (ii) deducting other of the 330 headings (e.g. income tax and national insurance contributions) and a specified proportion of yet other headings. Thus, if we reckon that 18% of the price of soap is purchase tax, then we must deduct 18% of the particular household's expenditure on soap in order to arrive at its net income after both direct and indirect taxes. Table 2 is designed to give information on the redistributive effects of all forms of taxation and subsidies and social service benefits.

The statistics given in some of the tabulations need to be corrected for differential response rates, i.e. to allow for the fact that the percentage of households co-operating in the inquiry varies from one stratum of society to another.

The initial proposal for meeting these statistical requirements was to use punched-card machinery, but adequate capacity was not available at the right time. Moreover, it would have been very difficult, if not impossible, to organize punched-card machines to do all the mathematical operations required. At the time the problem was under discussion, we were hearing a lot about electronic computers and it seemed to us that here was the obvious answer. The computer could do all our complicated mathematics, the whole job would be done in a fraction of the time and, we hoped, at a fraction of the cost of the punched-card methods.

THE CHOICE OF COMPUTER

At the time when use of a computer was first discussed, no Government department outside the scientific civil service had a computer installed and working. Instead of buying time on a commercial installation, we found that time could be made available on the twin Deuce installation at the Royal Aircraft Establishment, Farnborough, Hampshire. We did not consciously choose Deuce from a range of possibilities. After learning of the Deuce capacity available for our job, the National Physical Laboratory staff were called in as computer experts, and they satisfied themselves that it was a reasonable proposition to analyse the family expenditure survey on Deuce. They made some estimates of time—both programming time and computer running time—and of costs; these estimates suggested that we should get our job done more quickly and cheaply than by punched-card methods. Looking back, we now realize that their figures of time and costs were seriously underestimated. The underestimation occurred partly because the extent of the statistical requirements was not fully appreciated, and in part because the programming team has been inexperienced and the team's composition has changed several times.

It will be generally appreciated that Deuce uses a 32-bit word and the high-speed store consists of 12 mercury delay lines, each holding 32 words, together with a small number of shorter delay lines each holding 1, 2 or 4 words. A magnetic drum holds a further 8,192 words. The basic bit frequency is 1 megacycle per second, so that the word-time is 32 microseconds. Addition and simple logical operations take 2 word-times (64 microseconds); multiplication and division take about 65 word-times (2 milliseconds). Transfers to or from the drum of blocks of 32 words occupy 13 milliseconds, with an additional 40 milliseconds if a drum head-shift is required. It is possible for multiplication or division and drum transfers to take place concurrently with other instructions. Likewise, calculations may proceed between reading or punching consecutive rows of a card. The program currently being operated is held in the long mercury delay lines. To get full benefit from the fairly high speed of the Deuce machine, it is therefore necessary to adopt optimum timing techniques, i.e. so to arrange the instruction words in the long delay lines as to minimize operating time. In my opinion, this need for optimum programming is a major disadvantage of Deuce from the programmer's point of view.

COMPUTER INPUT AND OUTPUT

The input and output media to the Deuce computers at Farnborough are Hollerith cards (input at 200 cards a minute, output at 100 cards a minute); reading and punching can now be on 32 or 64 columns of the 80-column cards, according to the setting of a switch. For the family expenditure survey, we used the original Deuce input and output arrangements, which covered only 32 columns of the card. For the large mass of data which we had to handle, it is likely that magnetic tape would in future be a more appropriate input and output medium because of its speed, given suitable off-line conversion facilities. It could also be used as a backing-store, to hold summaries of the basic data after the first computer run. The English Electric Company now offers magnetic tape supplementary storage for Deuce.

Although the Deuce machine used Hollerith 80-column cards as input, the Ministry of Labour had punched the results of the survey on to Powers 65-column cards, on the basis of one card per item of expenditure; as already explained, they did it this way to enable them to do their own analyses on their own Powers punched-card machinery. Thus we had the massive job of reproducing the 700,000 Powers cards into 700,000 Hollerith cards. From the computer point of view, the 700,000 cards represented an inefficient form of input, because, apart from certain master cards carrying details of the household (e.g. composition, income), each card effectively recorded only one item of expenditure—the amount and its item code: this information was decimally punched in 11 columns of the card. Merely to read this mass of 700,000 cards required something like 60 hours of computer time. It was necessary to read the basic material about eight times, because of the variety of statistical analyses required. So, to avoid reading in three-quarters of a million cards more than once, we decided that the first task must be to feed into the computer the 700,000 cards and program it to punch out 60,000 binary-punched cards; each of these cards carried details of not one, but 12 items of expenditure. All the subsequent statistical analyses were carried out by feeding in this condensed pack.

I would suggest that the moral of this experience is that we should make sure that the basic data is punched, or otherwise recorded, in a form which is both suitable and economic for the computer, and punched at as early a stage as possible.

THE PROGRAM AND COMPUTER OPERATION

The computer operation falls broadly into three parts.

(A) The condensation run, already described, in which all the primary (decimally-punched) material is converted into a set of binary-punched cards. This occupies about 125 hours, allowing for card handling and time to check queries thrown up by credibility tests.

(B) A series of about 8 runs, in each of which the condensed binary-punched pack is read; during each run aggregates of various statistics (e.g. aggregates of expenditure and aggregates of squares of expenditure) are built up with the requisite classification (e.g. by income) and these aggregates are punched out in binary at the end of the run. Each of these runs has occupied 20 to 30 hours of computer time.

(C) A series of programs, which compute the arithmetic means, standard deviations, standard errors of means, and so on, from the aggregates built up in (B); these programs punch ou^t the results decimally, in a form suitable for input to the Hollerith tabulator. Most of these runs are of relatively short duration.

I do not propose to go into detail about these programs, though I might say, in passing, that we introduced floating-point routines at one point to calculate standard deviations. This we did because of the great variability in the magnitudes of expenditure on different items and in the numbers of households to which the standard deviations related.

I will, however, make a general observation on errors. Most errors are due to one of three causes:

(*a*) faulty data, or data outside the limits for which the program has been designed;
(*b*) faulty operating; and
(*c*) machine faults;

and a well-designed program should incorporate checks on each of these. We have found this particularly important on some of the long runs, for example when reading our condensed pack of binary cards in order to form aggregates of expenditure [(B) above]. Wherever possible the error routines include re-entry routes which enable the fault, whether of data, operating, or machine, to be corrected.

With some errors, however, particularly machine errors such as drum writing failures, the results of a run are irretrievably lost. To minimize losses of this sort, our long-playing programs contain provisions for intermediate punching out of all aggregates accumulated to date, as often as required. If an irretrievable error occurs, it is not necessary to go back to the start of, say, a 5-hour run, but only to the last intermediate punch-out, which might mean re-doing at most 60 minutes of work. When, as in our case, the punching out of the intermediate aggregates occupies up to 10 minutes, there is an obvious upper limit to the frequency of such punch-outs. Consequently a frequency of arithmetical or drum errors of once every 30 minutes, which might be acceptable for some programs, becomes impossible with some of the runs in the family expenditure analysis. The conclusion I draw is that, for this kind of statistical work, a high standard of machine reliability is essential for efficient operation. If we had used magnetic tape as a backing-store, the problem of machine reliability would have been less acute, because it would have been possible to write away the intermediate aggregates on to tape much more frequently, without using up an unreasonable amount of computer time on such "unproductive" work.

Some of our programs incorporate other features aimed at minimizing the disorganization and loss of time from errors, two of which are worth mentioning.

One of the causes of lost time which we have encountered is faulty punching of the computer output. Consequently, some of our programs now require the validity of the punched-out packs to be checked, by passing these packs through the reader. Only if the check sums in these punch-outs tally do we proceed to the next part of the program. If the check sums do not tally, there are facilities for, alternatively, punching out a duplicate pack, or re-reading the punched-out pack; this latter facility is introduced to deal with reading errors.

Some machine errors can be corrected without intervention by the operator—even without the operator knowing anything about it. This is possible if there is a re-entry route into the program, and if no mechanical manipulation of packs of cards is involved. The flow diagram we have adopted in several cases is on the lines of Fig. 1.

Only if the machine is incapable of getting satisfactory results from the part of the program in question do the alarm signal and failure lights stay on; this happens if the machine gets trapped in the failure loop. On a single circuit of the failure loop, the machine may slip through unseen and unheard; and if one did not wish to encourage it in such stealthy practices, one could always instruct the machine to keep a record of its journeys through failure loops.

I should like to mention one problem of programming a statistical job such as the family expenditure survey, a problem which is also present in most so-called clerical processes. This is the problem of flexibility. It ought to be possible to modify the program to cope with

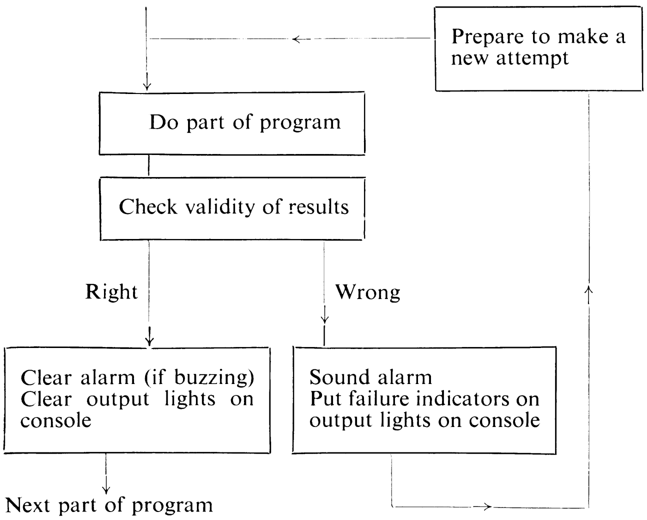
Prepare to make a new attempt
Do part of program
Check validity of results
Right
Wrong
Clear alarm (if buzzing) Clear output lights on console
Sound alarm Put failure indicators on output lights on console
Next part of program

FIG. 1.—Flow diagram for error correction routine.

small—or perhaps large—changes in requirements without extensive re-writing, and, moreover, it ought to be possible for someone not conversant with all the detail of the program to carry out this modification. With many mathematical problems, such as the linear-programming problem, the introduction of preset parameters gets round the difficulty, and the number of such preset parameters is not excessive. With a statistical job of the family expenditure survey type, the number of preset parameters is likely to be very much greater, and, moreover, many of the variants of the computational scheme cannot be envisaged in advance. Even where one can foresee that it would be desirable to treat some quantity—such as the number of income groups into which households are classified—as a preset parameter, to do so may unduly complicate an already complicated program of, say, 1,000 instructions.

Moreover, allowing for such preset parameters is more difficult in a machine such as Deuce, wherein the storage locations are arranged in blocks and sub-blocks of fixed dimensions, than would be the case with a machine whose N storage locations were arranged in a single block 1, 2, . . ., N. I am offering no solution to this problem. I can only say that I am conscious that the programs for the family expenditure survey are not as flexible as they ought to be. It might be fair to say that the computer itself is about as flexible a piece of machinery as exists, but the computer program is often most inflexible.

Perhaps this is the point to comment on automatic programming. Let me say that we have no experience of this on the family expenditure survey, but I look forward to automatic programming routines, designed for statistical and clerical-type processes, which will cut out a lot of the routine of programming and turn man-years into man-months. Perhaps it will help us on this problem of preset parameters—perhaps we can write our program in a Fortran-type autocode on the lines:

```
        . . . . . . . . . . . . . . . . .
        . . . . . . . . . . . . . . . . .
number of income groups $i = I$
classify aggregate $e_j$ by $i$ ($j$ = 1 through $J$)
                . . . . . . . . . .
                . . . . . . . . . .
                   $I = 14$
                . . . . . . . . . .
```

and then let the computer slave away with its conversion program to produce a set of program cards in machine language; and I would hope that the resulting computer program would not require substantially more computer operating time than a program prepared in the ordinary way. If, then, we want to change I from 14 to 17, only one trifling change in the list of autocode instructions will be required, and the conversion program can then be re-run to give us a new set of program cards in machine language. Perhaps I am being too optimistic. It does seem to me that manufacturers and major computer users (e.g. the Government) might with advantage get together and develop computers whose design and order codes were specially suitable for autocoding, and develop autocoding procedures to be used on these computers.*

PROGRAMMING AND PROGRAMMERS

We were fortunate in that the National Physical Laboratory not only undertook the studies of feasibility to which I have referred, but also undertook to supervise the first stages of programming. They provided what we have come to call "generalship" over the programming team.

Because this particular piece of programming was regarded as an *ad hoc* once-for-all job, and because the Central Statistical Office did not itself have the necessary trained staff, we relied to a considerable extent on personnel loaned from a number of departments to do the detailed programming work. In return for the help given us it was hoped that these personnel would gain practical programming experience. None of them had had previous programming experience, and by the time we had trained them in the intricacies of Deuce and the family expenditure survey, the period of their loan had as often as not expired. I think that with some exceptions these trainees and the departments employing them probably got more from the experience gained than we got in terms of completed and efficient programs. Although it may be true that programming can be taught in a three-weeks' course, program output in terms of both quality and quantity can, I believe, increase by a factor of perhaps 10 in a period of a year.

My personal feeling is that the volume and complexity of programming work is often underestimated by those

* The British Computer Society has two groups of committee members studying automatic coding techniques for mathematical and business applications, respectively.—ED.

without practical experience of it; and that the greatest care needs to be exercised to select programming staff with the requisite logical abilities and qualities of meticulous accuracy.

For all these reasons we grossly underestimated the time required for programming the family expenditure survey, by a factor of about 3. This is, I believe, a fairly common experience.

COSTS

I would have liked to end by giving figures of cost which would demonstrate the economic advantages of using a computer for this sort of work: but no precise figures are available. All I can say is that the estimated costs of doing the job on punched cards (so far as punched-card procedures could cope with the task) exceeded the estimated computer costs by a substantial margin. The *actual* computer costs have also exceeded the *estimated* computer costs by a sizeable margin.

What I can also say is that, once the programs were written, the computer has given us results more quickly than would be possible with punched-card methods, and given us analyses which would barely be feasible by alternative methods. These perhaps are more important considerations in favour of using a computer on a statistical job, than small savings in costs.

REFERENCE

WRIGHT, M. A. (1959). "Techniques for Analysis of a Family Expenditure Survey on a Computer," *Business Computer Symposium.* London, Pitman.

Handbook for Automatic Computation

Preparation of a handbook for automatic computation, in five or more volumes, is now under way for publication by Springer-Verlag. It will appear in F. K. Schmidt's series, "Grundlehren der Mathematischen Wissenschaften." Editors are

F. L. Bauer, Mainz.
A. S. Householder, Oak Ridge.
F. W. J. Olver, N.P.L., Teddington.
H. Rutishauser, Zürich.
K. Samelson, Mainz.
R. Sauer, Munich.
E. Stiefel, Zürich.

The purpose of the handbook is to provide a collection of tested algorithms for mathematical computations of all sorts: the solution of finite and of functional equations, methods of approximating functions, the evaluation of special functions, etc. These algorithms are to be written in Algol, hence will be usable on any machine for which a suitable translator is available, and even without a translator can be used as a model for programming. It is evident that such a collection could have no general utility unless written in some common program language. The descriptive language will be English.

As plans now stand, the organization of the series will be as follows: Volume 1A will contain a description of the use of Algol, and Volume 1B a description of the structure of translators. These introductory volumes are the only ones that will not be made up primarily of actual algorithms. Volume 2 will be devoted to the solution of finite equations, linear and non-linear, including the determination of characteristic values and vectors of matrices. Volume 3 will be on functional equations, especially differential equations, ordinary and partial, and integral equations. Volume 4 is concerned with methods of approximation, and Volume 5 the evaluation of particular functions. It is possible that certain algorithms, such as those for solving inequalities, for mathematical programming, for statistical computations, and the like, that do not seem to fall naturally in any of these areas, may be reserved for a sixth volume. Each algorithm is to be accompanied by enough explanatory information to make it understandable, along with whatever information is available on speed, accuracy, range, or, more generally, for judging the effectiveness of the algorithm for a given type of problem. In any event, only pretested algorithms will be published.

Before the appearance of the volumes themselves, the algorithms will be prepublished in a series of supplements to the journal *Numerische Mathematik*. This is partly to make generally available each algorithm at the earliest possible time. But in addition to this, it provides the possibility for including in the handbook itself additional information, and even corrections, that might come in from users.

Contributions are earnestly solicited. For the present, at least, these must necessarily be in the form of actual algorithms, along with information as to the extent and mode of testing the algorithm, estimates of accuracy, and experience in using it. Untested algorithms will not necessarily be rejected *ipso facto*, but their use must necessarily await actual test. As algorithms are published, information relating to published algorithms also will be welcomed. Contributions may be sent to any of the editors named above.