COSMOS—Courtaulds' Own System for Matrix Operations and Statistics

By W. O'Brien*

This paper describes the software designed and written at Courtaulds for the statistical analysis of data and standard matrix routines. Input to the statistical routines is by means of a program which can form mathematical models of the user's choice, and the matrix suite is designed so that a program may be written by the user using normal matrix terminology.

COSMOS is an integrated suite of programs which is designed to provide access to some of the standard techniques of statistics and matrix analyses by means of a general and flexible input system.

Although the programs are grouped as a set, it is convenient to consider them as separate entities.

Statistical suite

This is designed to assist the engineer or chemist who has a mass of data to analyse. Some of the factors, he may decide, are dependent on others which between themselves are independent; or he may wish to investigate whether the variables in a set are related to other members of the same set. By grouping the variables into dependent and independent sets and perhaps after intermediate results, regrouping the data, the suite enables the scientist to arrange his experimentation in a more logical manner.

Input

The input handles up to 10,000 sets of variables, each set containing up to 96 members. Data is recorded conveniently on the forms illustrated in **Fig. 1**. Each column contains space for a descriptive heading, immediately above the card column number, while below this number is a further space for the variable identification number. This number consists of a letter *I* followed by a two-digit number, thus the first variable in a row would be *I*01, the second *I*02 and so on, the numbers being in strict numerical sequence. Columns 75–78 contain the row and 79–80 the card number on which a sequential check is maintained at reading-in time.

Since each sheet takes up to 9 variables, a set of, say, 20 variables would be recorded on 3 cards. Data is punched in fields of up to 9 characters including the decimal point, and gaps in the data are accepted as blank fields.

Control

It is desirable both from the point of view of the user and the computer department that a virtually unlimited selection of models of the data should be available, and this is achieved by three types of card called CONT, PROG and CONS respectively. The data can be held on tape and a selection of the control cards can then give any mathematical model required within the limits of the machine. Each of these cards will be considered separately.

CONT cards

A control card (**Fig. 2**) is the initial card of each program. Its function is to specify to the machine the number of variates along with the number of independent variables x and dependent variables y which

COMPUTER DEMAND FOR STATISTICAL ANALYSIS

TITLE DATE SENDER

								ROW No.	CARD No.
1–8	10–18	19–27	28–36	37–45	46–54	55–63	64–72	75–78	79–80

Fig. 1

* *Courtaulds Ltd., Matlock Road, Coventry.*

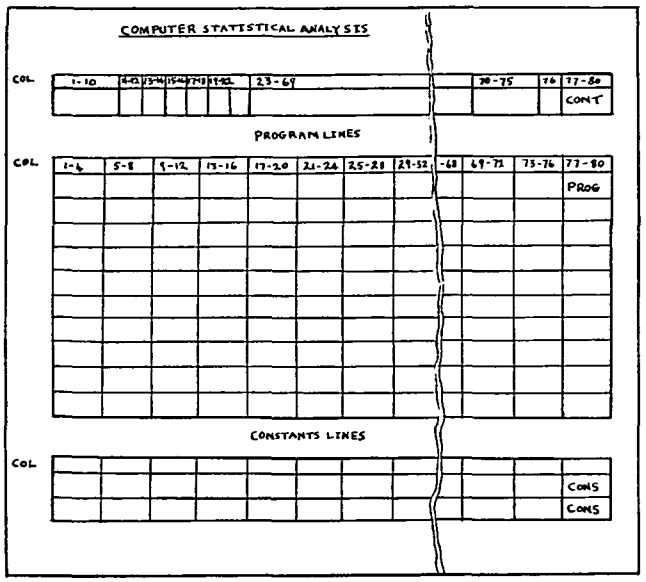


Fig. 2

are to be produced from the data. Since constants may be specified, the number of these which are used is also stated here and the total number of rows (sets) of variables. Supplementary information such as run identification and descriptive headings can be entered via this card as well as a tape control character for choosing the desired program on completion of the input segment. With the CONT card the machine knows the exact numbers of variables and sets it has to work with and should the following data be inconsistent an error diagnostic will be printed, although not necessarily a fatal one.

PROG cards

Control over the selection of variables is achieved by the use of program cards which are identified by the letters PROG in the last four columns. The remaining 76 columns are divided into groups of four characters, the first symbol of each group being an operator (**Table 1**), the remaining three a variable identity, e.g. $I01$, $X03$, $C07$, etc. A full stop immediately after the last group signifies the end of the program lines. In the examples which follow Δ signifies a blank character.

The simplest operation is of the form

$$\Delta I07 = X01$$

When the arithmetic operators are used the hierarchy of operation is from left to right so that

$$\Delta I15 + I16 * I18 \text{ means } (I15 + I16) \times I18$$

For more complex operations the combination operator A is used.

$$\Delta I01 + I03 * I11 LA01 = X01$$

means

$$\log e(I01 + I03)I11 = X01$$

Table 1

Control operators for the input segment

SYMBOL	INTERPRETATION
Δ	TAKE THE FOLLOWING VARIABLE
(Blank)	
$=$	EQUATE TO
T	TRANSFER (e.g. $TI01$ means transfer I_{ij} to X_{ij})
A	THE WHOLE EXPRESSION TO THE LEFT AFTER THE SIGN Δ OR $=$
.	END OF THE PROGRAM
$+$	ADDITION
$-$	SUBTRACTION
$*$	MULTIPLICATION
$/$	DIVISION
L	NATURAL LOGARITHM
S	SQUARE ROOT
E	EXPONENTIAL
P	PRINT THE VARIABLE SHOWN

Furthermore, each variable obtained is stored so that it can be operated on by a subsequent instruction. For example

$$I01 \log (aI02^2 + C)$$

can be programmed as

$$\Delta I02*I02 = X01*C01 = X01 + C02LA01*I01 = X01$$

where $C01 = a$
$C02 = C$

During the construction of the model a blank data field may be encountered, in which case the row is ignored. This of course does not preclude the incorporation of this particular row in subsequent models which did not contain the variates with blank fields.

CONS cards

A final card completes the triumvirate of control. Although normally used for the supply of constants to the program, the constant lines, since they are continuously in core, can be used as variable storage and thus perform a more constructive role. A particular example of this is to stagger the readings since, on a continuous plant, a technological time lag may exist between readings taken at the same time. This can best be illustrated by the following example. In the set of variables

$I01$	$I02$	$I03$	$I04$	$I05$
i_{11}	i_{12}	$\underline{i_{13}}$	i_{14}	i_{15}
i_{21}	i_{22}	i_{23}	i_{24}	$\underline{i_{25}}$
i_{31}	$\underline{i_{32}}$	i_{33}	i_{34}	i_{35}
$\underline{i_{41}}$	i_{42}	i_{43}	$\underline{i_{44}}$	i_{45}

a relationship (Y ; $I01$, $I02$, . . ., $I05$) is required. However, when the data are recorded time lags exist so that the readings underlined must be taken together, *et seq.*

The PROG control card

$$TI01\Delta C04 = X02\Delta C01 = X03TI04\Delta C03 = X05\Delta C02 = C01\Delta C06 = C03\Delta C05 = C02\Delta I02 = C04\Delta I03 = C05\Delta I05 = C06$$

will produce the sequence of transformations

	$X01$	$X02$	$X03$	$X04$	$X05$	
1	i_{11}	Δ	Δ	i_{14}	Δ	Reject
2	i_{21}	i_{12}	Δ	i_{24}	Δ	,,
3	i_{31}	i_{22}	Δ	i_{34}	i_{15}	,,
4	i_{41}	i_{32}	i_{13}	i_{44}	i_{25}	First line

	$C01$	$C02$	$C03$	$C04$	$C05$	$C06$
	Δ	Δ	Δ	Δ	Δ	Δ
1	Δ	Δ	Δ	i_{12}	i_{13}	i_{15}
2	Δ	i_{13}	i_{15}	i_{22}	i_{23}	i_{25}
3	i_{13}	i_{23}	i_{25}	i_{32}	i_{33}	i_{35}
4	i_{23}	i_{33}	i_{35}	i_{42}	i_{43}	i_{45}

An obvious extension to this is the computation of cross correlation functions. Also a pseudo random sequence of numbers could be generated by a cunning use of the *C*s, although E.C.S.L. (see Clementson, 1966) is the natural vehicle for this.

The output segment of the program supplies as its first two data records

(1) the contents of the control card
(2) the mean values of the sets of data.

These are followed by records consisting of the set of *X*s and *Y*s.

Statistics

The main purpose of the suite is to provide a set of regression equations (y; βx) which are estimated by minimizing the sums of squares

$$(y - \beta x)'(y - \beta x)$$

of the residual terms $(y - \beta x)$.

During the process of obtaining these estimates several functions of statistical interest are obtained, which can be printed out by option using sense switch settings.

Initially, the transformed variables $(x;y)$ are read from the magnetic tape constructed by the output segment and the sets of mean deviates $(x–\bar{x}; y–\bar{y})$ are formed. As each row is processed this is partitioned into subsets $(x–\bar{x})$, $(y–\bar{y})$ which are written to tapes 3 and 4 respectively; these are rewound at the completion of the segment preparatory to the formation of the matrix of products and cross products $X'X = (x - \bar{x})'(x - \bar{x})$.

In order to carry out subsequent matrix operations entirely within core, use is made of the fact that the matrix $X'X$ is symmetric and it is accordingly stored as an upper triangular matrix. As each row i is read, the product $x_{ij}x_{ik}$ is formed for each value of j and added to the ikth element of the matrix. It is now possible to compute and print the correlation matrix C whose terms are

$$C_{ij} = \Sigma x_{ki}x_{kj}/(\Sigma x_{ik}^2 \Sigma x_k^2 j)^{1/2},\ k = 1, n$$

that is

$$C_{ij} = \frac{X_{ij}}{\sqrt{(X_{ii}X_{jj})}}$$

or, in words, by dividing the terms of the information matrix by the square root of the leading diagonal elements in the same row and column as the element concerned.

If a set of correlation coefficients only has been requested, the program is now terminated. Alternatively, the correlation coefficients provide a measure of the relationship of the members of the independent set within themselves, and therefore an indication of the conditioning of the matrix to be inverted; thus a printout of the correlation would be valuable in indicating whether a regrouping of the variables is necessary.

The program continues with the inversion of the information matrix ($X'X$). A compact method is used (see Appendix) in which the array elements are transformed by a set of linear relationships. At the end of the inversion the matrix is stored on magnetic tape immediately following the set of mean deviates $(x–\bar{x})$, and is thus available when evaluation of the cross products, which is the next stage, has been completed. The products and cross products $Y'Y$ and $X'Y$ are then calculated, and $X'Y$ written in columns on to magnetic tape. With the matrix inversion and cross products evaluated the bulk of the work has been completed and the regressions are now estimated.

In order to cater for the maximum size of problem consistent with the most efficient usage of core and peripheral handling, the regressions are calculated in blocks of up to 8 equations at a time. Simultaneity with the peripheral handling is obtained by starting the calculations when the first records of the matrix and the block of cross products are in core, so that the calculation is proceeding whilst the remainder of the blocks are being read in.

The estimate of the regression coefficients β is based on the equation

$$\beta = (X'X)^{-1}X'Y$$

where

$$\beta = (\beta_1, \beta_2, \beta_3 \ldots)$$

and

$$\beta_0 = \bar{y} - \sum_{i=0}^{m} \beta_i \bar{x}_i.$$

A print out of each of the regression equations is now executed, during which the regression sums of squares $\beta'X'Y$ and variance $\beta'X'Y/(m - 1)$ are calculated. When the print of each individual equation is completed the total variance $Y'Y/(n - 1)$ and residual

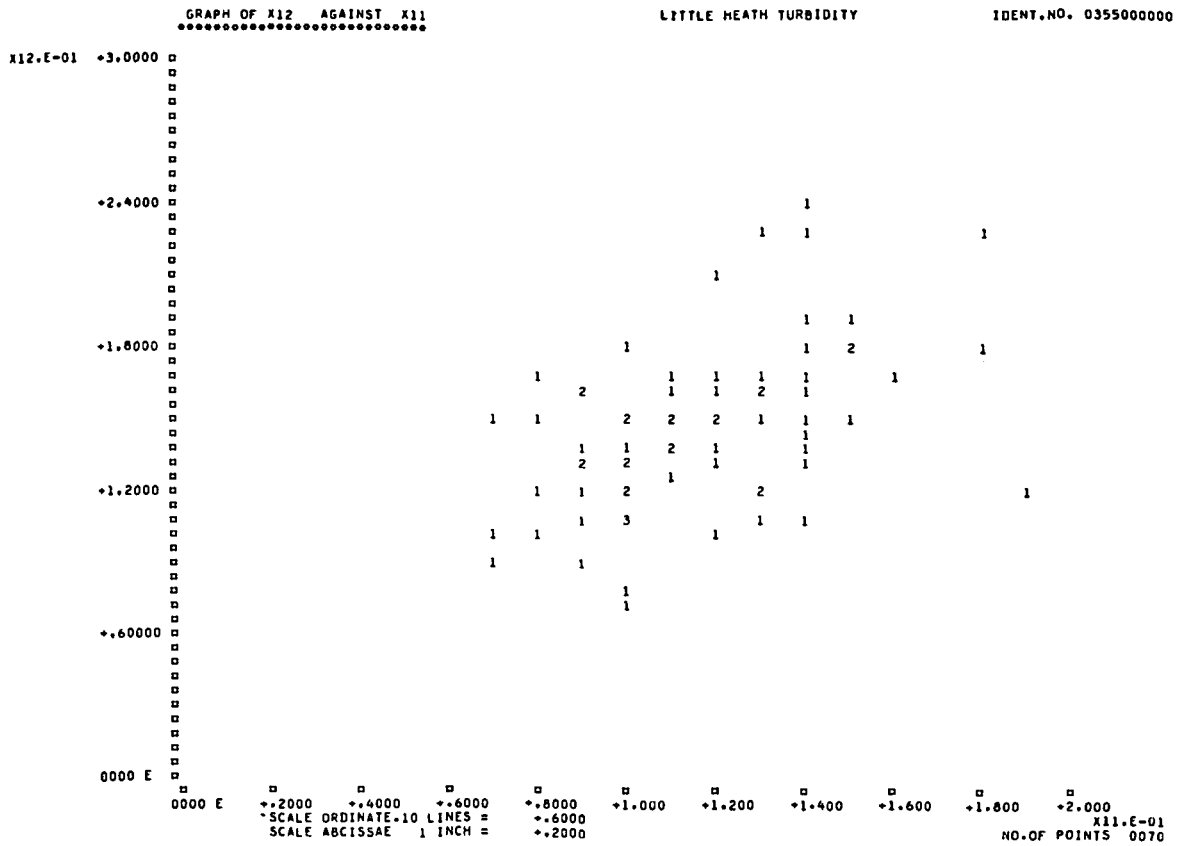


Fig. 3

$(Y'Y - \beta'X'Y)/(m - n)$ is computed, and the analysis of variance table printed out with the F-ratio

$$F = \frac{\text{Regression variance}}{\text{Residual variance}}.$$

To complete the picture the confidence limits of the set (β_i) are printed based on $s\sqrt{a_{ii}}$

where $A = (X'X)^{-1} = (a_{ij})$

and s is the square root of the residual variance.

Scatter diagrams

An ancillary to the multiple regression program—which can of course be used independently—is the provision of a program for drawing scatter diagrams. Input is normally through the input program and the use of CONT and PROG cards prior to the data. The program will print out, as illustrated in **Fig. 3**, a plot of $X_{i+1} : X_i$, where $X_i = X01, X03, X05$, etc.

Thus it is possible to plot up to 32 different diagrams on any one occasion from the same data. Visual scaling of the diagrams can be improved by setting of sense switches, e.g.

SS 1 will scale the X axis by 10
SS 2 „ „ „ Y „ „ 10
SS 3 „ „ both axes by 10

and used in combination other scalings can be obtained. Individual graphs can be selected by typing in the number of the graph to be modified so that if in a batch of 32 graphs one or two of them needed modification of the scale, these graphs could be modified without repeating the entire run. Numbers of points with the same co-ordinates are represented up to 9 by numbers, then 10–19 by A, 20–29 by B, etc. An example of the use of the facility would be in the plot of the residuals following the regression analysis.

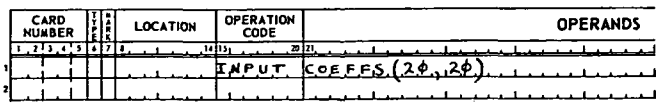
Matrix suite

Standard matrix routines are provided in the program by use of operators, which are listed in **Table 2,** and operands consisting of matrix or scalar sets identified by unique 6-character names. A form of program statement is used which is chosen to conform with Honeywell Easycoder practice. Thus, one would merely have to write

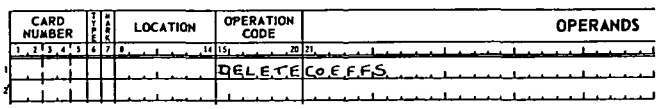
CARD NUMBER	LOCATION	OPERATION CODE	OPERANDS
1		INVERT	COEFFS, RECIP

for the inverse of the first matrix to be found and given the name RECIP.

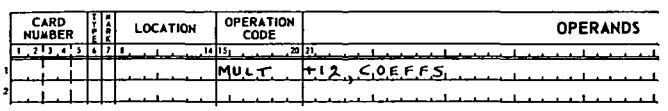
The size of each matrix is obtained on input as



and on input the original matrix is stored on a library tape until deleted by a subsequent instruction



Constants are introduced by the use of a signed number.



would multiply every element of COEFFS by 12.

Apart from the trivial operations of addition, subtraction and multiplication, the main part of any matrix program lies in the inversion routine, and in the determination of latent roots and their associated vectors. One is rarely faced with a problem where the single determination of these arrays is the sole object of the exercise, and the suite is written so that a user who can frame his problem in matrix algebra can write limited programs using a sequence of operators.

Consider an elementary problem in the theory of elasticity of a set of particles connected by a series of elastic strings. The equation of equilibrium can be written

$$M\ddot{Z} + KZ = 0$$

or $$\ddot{Z} + \Lambda Z = 0 \quad \text{where } \Lambda = M^{-1}K$$

It can be shown by assuming Z to be of the form

$$Z_i = a_i \cos p_i t$$

that the latent roots of Λ are the fundamental frequencies of the system and the set of orthogonal vectors associated with these roots are the normal modes of vibration. A program to determine these could be written

```
INPUT     MASS (10,10)
DIVIDE    MASS, STIFF, LAMBDA
ROOT      LAMBDA, FREQ
REMARK    FUNDAMENTAL FREQUENCIES
PRINT     FREQ
VECTOR    LAMBDA, MODES
REMARK    NORMAL MODES
PRINT     MODES
DELETE    STIFF
END
```

The program would read the matrix MASS into core and at the same time write it to the library tape. MASS would be inverted, the matrix STIFF read from the library tape and premultiplied by the invert to give a new matrix LAMBDA. Calculations of the latent roots, which would be stored as a vector called FREQ, would then take place.

REMARK is a code which tells the machine to print anything punched on that card, so the heading FUNDAMENTAL FREQUENCIES would be printed followed by the vector FREQ. VECTOR would then produce the latent vectors which would be printed under the heading NORMAL MODES. STIFF would be deleted from the library tape.

Table 2

Operators for the matrix segment of COSMOS

OPERATOR	INTERPRETATION
INPUT	Read the matrix from cards
ADD	Perform $A + B$ where A or B can be matrix, vector or scalar
SUB	Perform $A - B$ where A and B as for ADD
MULT	Perform $A \times B$
POWER	Raise the square matrix A to the power B
INVERT	Invert the matrix, which must be square
DIVIDE	Find the inverse of A and premultiply B
DET	Calculate the determinant of the matrix
TRANS	Find the transpose of the matrix
PRINT	Print the array requested
DELETE	Delete the matrix from tape
REMARK	Print the contents of this card
EXIT	End of the program
COMP	FORM $A'A$
ROOTS	Find the latent roots of A
VECTOR	Find the latent vectors of A
TRACE	Calculate the spur or trace of A

Methods

The inverse is carried out by a compact method (see Appendix) which enables the maximum use to be made of core storage. In addition the product of the leading elements of the upper triangular matrix U gives the value of the determinant. Should any one of these elements be zero, this would indicate that the matrix is singular at a fairly early stage of the program.

In the determination of latent roots and vectors a transformation method (Fox, 1964) is used where the original matrix is reduced to tri-diagonal form.

Input

Input is by means of the card illustrated in **Fig. 4**. Columns 25–26 and 28–29 designate the number of row and column of the first element on that particular card. In this way a sparse matrix need not be punched in

entirety. For example, a tri-diagonal matrix could be input as

01	01	a_{11}	a_{12}	
02	01	a_{21}	a_{22}	a_{23}
03	02	a_{32}	a_{33}	a_{34}

and so on. The word END in the first three columns of the last card signifies the end of the data.

Multi factor analysis

From the routines established in the general program, we can now breach the dyke of multifactor analysis. Basically, the problem is in the multiplicity of variates which we wish to reduce. It is known that the latent vectors of the correlation matrix give a set of transformed variates which are statistically uncorrelated. Accordingly, we proceed in the main regression program to the point where the matrix of correlations has been constructed and printed along with the mean values and standard deviations of the variables, and then call on the programs ROOT and VECTOR to produce the set of latent roots and vectors required. Suppose we wish to reduce our set of n variates to p, then we find the first p latent roots from the correlation matrix, and thus obtain a matrix A of n rows, whose p columns are the variates required. The set of vectors so obtained are rotated in pairs maximizing the variance of the elements of $A'A$ until a point is reached where the overall variance is unchanged, and the p vectors are printed out.

The elements of these vectors represent the contributions made by each of the n variates and thus an indication of those which can be omitted on subsequent models.

Conclusion

The current regression program is not regarded as the final word, and at the moment improvements are being

COSMOS MATRIX INPUT SHEET

TITLE.....

DATE.....

SENDER .....

COL	15–17	25–26	28–29	36–46	47–57	58–68	69–79

Fig. 4

investigated. As a result of the analysis of a set of variates (x_i) it will inevitably be found that some of the regressions are statistically non-significant. By removing variables which are not significant a set $(x_j) \subset (x_i)$ should be found where removal of further terms has no significant effect on the sum of squares, and the program is being extended to do this. The set (x_j), however, may not be unique and further investigations in this direction could prove fruitful. Tests for serial correlation are being programmed and will also be included in the program at a later stage.

Finally, the program is intended to provide a service to non-specialist users and as an additional aid a printed comment on the results obtained is to be added in a later segment.

Where matrix algebra is concerned the problem lies in the education of the user, and so our investigations will be in the direction of finding applications for this type of work, an obvious example of course being in the field of linear control systems which we will be investigating in the near future.

References

FOX, L. (1964). *An Introduction to Numerical Linear Algebra*, Oxford University Press, pp. 247–9.
CLEMENTSON, A. T. (1966). "Extended Control and Simulation Language", *The Computer Journal*, Vol. 9, p. 215.

Appendix

In the statistical and matrix programs a compact method is used for the matrix inversion. In this the matrix is decomposed into two factors consisting of upper and lower triangular matrices U and L so that

$$A = LU \tag{1}$$

where $A = (a_{ij})$

$$L = (l_{ij}) \quad l_{ij} = 0, \quad i < j; \; l_{ij} = 1, \; i = j$$

$$U = (u_{ij}) \quad u_{ij} = 0, \; i > j$$

$$LL^{-1} = I; \; UU^{-1} = I \tag{2}$$

then $$A^{-1} = (LU)^{-1} = U^{-1}L^{-1}. \tag{3}$$

By expanding the matrices and equating elements on

both sides of the equations (1), (2) and (3) the following set of algorithms can be obtained

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} u_{ik}u_{kj} \qquad i \leqslant j$$

$$u_{ij} = \frac{a_{ij} - \sum_{1}^{j-1} u_{ik}u_{kj}}{u_{jj}} \qquad i > j$$

where u_{ik} are the elements of the composite matrix LU

$$w_{ij} = -u_{ij} - \sum_{k=j+1}^{i-1} u_{ik}w_{kj} \qquad i > j$$

$$w_{ij} = \frac{1}{u_{ij}} \qquad i = j$$

$$w_{ij} = -\frac{\sum_{k=1}^{j-1} u_{ik}w_{kj}}{u_{jj}} \qquad i < j.$$

Finally

$$b_{ij} = \sum_{k=i}^{n} w_{ik}w_{kj} \qquad i > j$$

$$b_{ij} = w_{ij} + \sum_{k=j+1}^{n} w_{ik}w_{kj} \qquad i \leqslant j.$$

In the case of a symmetric matrix, the matrix can be stored as an upper triangular, and the above algorithms modified by introducing the relationship

$$a_{ji} = a_{ij}.$$

Book Review

An Audit Approach to Computers, by A. Pinkney, 1966; 159 pages. (London: *Institute of Chartered Accountants*, 20s.)

The book has been published by the Trustees of the *General Educational Trust of the Institute of Chartered Accountants in England and Wales* at the suggestion of the Council of the Institute. The views expressed are stated to be the author's own and not necessarily those of the Trustees or of the Council, but coupled with the fact that this is the first book published in the United Kingdom on the subject will undoubtedly tend to make Mr Pinkney's book at least the unofficial standard text book on auditing computers for some time to come.

Does it deserve a place in the accountant's bookcase? The answer is probably both yes and no, the latter in the sense that any accountant with "computer clients" will find the book more used as a constant companion than one lying on a shelf waiting for the pages to be turned if occasion demands. The book is written primarily for the external auditor but internal auditors, management and anyone concerned with establishing and reviewing systems of internal control will find a wealth of information and interest, including even a review of the elements of a computer system.

The first chapter deals with general audit considerations and covers the audit approaches, audit trails, internal control and the co-ordination of audit and management requirements. Then follow three chapters on controls, the first two dealing with those related to the manner in which the work of the data processing department should be organized, and thirdly those required by reason of the way in which the applications are planned to be processed in order to achieve reliable output records. Among the many aspects covered are:

Systems description documentation described in simple detail with references to program security, authorization, testing and amendments.

The division of responsibility on which the system of internal control is based is given considerable attention with particular reference to computer operators and control section—two very important features from the audit point of view.

The manner in which files containing the processed data of the business may be safeguarded: procedural controls are considered under the main headings of input, processing (including typical program checks), master file and output, with several easily understandable examples.

At the end of each of these chapters are a number of paragraphs on audit considerations which are in the nature of a "re-cap" of the previous subject matter regrouped from the audit viewpoint. A number of questions which are posed for the auditor to ask when faced with the problem of EDP audits will no doubt be most useful although it is felt that the author's views on the answers (as appropriate) could also have been most valuable, or at least the questions could have been referenced to the relative paragraph in the text. Brief references made to the dangers inherent in weak control serve to remind that even with EDP the body of fraud can still exist.

With all the results of the auditor's review of systems and controls in front of him there still remains the problem of evaluation of the complexity of the results and to this end Mr Pinkney has proposed the use of a control sheet (with example) as an aid to assimilating the information. He also suggests that it will normally be helpful for the auditor to prepare a special questionnaire or check list to cover each of the control aspects that have been featured in the book. It seems a pity that Mr Pinkney did not go so far as to provide one for his readers which would have been readily accepted as the standard work. Or perhaps it is at this stage that the reader, having been so well provided, is tending to expect everything to be "served up" for him.

The next two chapters deal with purely audit functions. The first describes, primarily, audit tests in relation to a computer application but sounding the warning that regard must also be paid to the controls which operate outside the data processing department. The second considers special audit techniques including the use of test packs and special computer programs. Mr Pinkney takes the view that conventional audit techniques will normally be applied wherever the system of processing allows this, but where the efficiency of the audit can be increased or the cost thereof reduced by the use of special techniques these should be considered. He recommends that the auditor's tests should be directed towards seeing that the internal controls, both procedural

(*Continued on p. 149*)