



COMPUTER  
CENTRE  
**BULLETIN**

*Volume 1, Number 3.  
2nd September, 1968.*

*Editor:  
J. S. Williams.*



## THIS EDITION

This month's Bulletin includes a progress report on the PDP 10 system and a further article in the series on Computer Centre Staff. It also contains sections on programming advice, systems modifications and seminars. The programming language MAC (MAtrix Compiler) is discussed and a member of staff takes an objective look at the perils of "least squares" methods.

## STAFF OF THE COMPUTER CENTRE

The second in this series of articles detailing the position, duties and responsibilities of the Computer Centre staff deals with the Maintenance Technicians and the Data Preparation Assistants.

### MAINTENANCE TECHNICIANS

The Computer Centre employs two Maintenance Technicians, Graham Jerrard and Doug Brunner. As their position implies, their prime function is to maintain the computer in all its aspects. This ranges from the overhaul and repair of a 40 h.p. motor and alternator set to troubleshooting and repair of highly sophisticated electronic circuitry.

Maintenance is divided into two categories,

- (i) preventive maintenance, and
- (ii) corrective maintenance.

Preventive or scheduled maintenance is directed towards keeping the whole system in perfect operating condition and is carried out daily between the hours of 7 a.m. and 9 a.m. The greater part of preventive maintenance is carried out on the mechanical and electro-mechanical devices such as the card reader, card punch, magnetic tape handlers, etc. Numerous adjustments on these devices have to be kept within close tolerances; motors and other moving parts have to be lubricated; brushes examined and replaced and all reading/writing areas must be kept clean. Diagnostic programs which are designed to test all sections of the computer system are run during the preventive maintenance period.

Corrective or unscheduled maintenance is carried out to correct some malfunction of the system. These malfunctions occur from time to time and must be corrected as soon as possible. Consequently the maintenance staff are present for two of the three shifts during each day.

The electronic circuitry of the GE 225 computer system has been extremely reliable since its installation in 1962. Apart from a few adjustments such as memory drive currents, photo electric pulse amplitudes and some critical pulse widths, preventive maintenance of this circuitry is limited and corrective maintenance is only occasionally required. When a printed circuit board does fail, the faulty board must be located, replaced and later repaired. In most instances, it is much more difficult to locate the faulty board than to repair the fault. Thus it is essential for the maintenance technician to have a detailed knowledge of the operation of the whole computer system.

As with most maintenance of electronic equipment, the real "nightmares" are intermittent faults - those which occur once and then disappear for several days. In these instances the maintenance technician may have to write a special program to repeatedly test the section of equipment which is suspect.

Lest the impression be created that computers are unreliable, it should be stressed that electronic faults, either intermittent or solid, have occurred very rarely on the GE 225 (approximately 1 fault every 3 months). Because of the error detection circuitry which checks all data transfers within the GE 225 machine malfunctions are detected within a short time.

Some of the not so obvious responsibilities of the maintenance technicians include the maintenance of a comprehensive spare parts system. Although it is impractical to carry spares of every item used in the computer, a \$400,000 computer should not lie idle waiting for a replacement part. The spares holdings at the Computer Centre are backed up by larger holdings by Australian General Electric in Sydney. Expensive and unusual parts can be obtained within two days from the United States.

It is not always appreciated that magnetic tapes must be in first class condition at all times. A slight crease can cause a tape to move away

from the writing/reading head sufficiently to produce an error. Normal wear and tear can also result in oxide being shed from the surface of the tape such that errors occur. The maintenance technicians are responsible for the testing and repair of such tapes.

The maintenance staff also work very closely with the engineering staff in the design and implementation of engineering changes thus ensuring that the system contains the most important improvements. Two local changes have been the "stop at top of page" function for the high speed printer and the "straight-8" system for punching and reading paper tape.

The maintenance staff are also responsible for all paper and card supplies. Although, this may seem a trivial task, such things as limited storage space at the centre, the high volume of card and paper usage, and variable delivery times from interstate, must be taken into account. There remains little margin for error or "forgetting to order".

Finally, the work of a maintenance technician must be painstaking, careful and thorough. He must be capable of writing programs and be capable of resolving conflicts between program errors and actual hardware malfunctions.

#### DATA PREPARATION ASSISTANTS

Two Data Preparation Assistants, Dianne Ball and Angela Vidanovic provide a data preparation service at the Computer Centre. This includes the punching and verifying of programs and data for clients and the Computer Centre itself. At present they operate two IBM 029 Card Punches and an IBM 059 Card Verifier.

Accuracy is extremely important in preparing punched cards and therefore a high level of concentration is necessary. For this reason, any inquiries relating to data preparation must be addressed to the Administrative Officer, extension 8471.

#### PDP 10 COMPUTER SYSTEM ARRIVES

The first units of the PDP 10 computer system were delivered in July and installation of the equipment is underway. This dual processor timeshared system will be one of the fastest computers in the country.

The PDP 10 was manufactured at the plant of Digital Equipment Corporation at Maynard, Massachusetts. Some 19 packages totalling about 15,000 lbs. weight were air freighted from Boston to Brisbane. Messrs Peter Bennett and Bruce Tannock, of Digital Equipment Australia, are responsible for the installation of the PDP 10 and its maintenance over the next 12 months. Assisted by members of the Computer Centre staff, they connected power to the system and ran the first programs within two days.

The PDP 10 is currently undergoing comprehensive checks on the hardware and software. Acceptance Tests are expected to commence early in September to ensure that the equipment performs reliably and according to specifications.

After the system is accepted, the programming staff at the Computer Centre will commence development of system software to allow for batch-processing of FORTRAN programs initially, and subsequently time-shared remote terminals. A wide range of system software for job control, file management, and automatic accounting is planned, and numerous languages, subroutines and package programs will be gradually integrated with the system.

#### CHANGES IN CARD PUNCHES AT THE COMPUTER CENTRE

In preparation for the new PDP 10 computer system, the IBM Model 026 key punches which were in use at the Computer Centre have been replaced by Model 029 key punches. Unfortunately, the card codes produced by these two models differ in some respects.

In general, cards prepared on an 026 punch are suitable only for the GE 225. Cards prepared on an 029 punch are suitable only for the PDP 10.

However, because of the large percentage of programs written in FORTRAN, the compilers and execution packages on both the GE 225 and PDP 10 have been modified to accept cards prepared on either type of key punch. Other programs require that cards be prepared using the correct key punch.

#### CHARACTER CODES

The IBM Model 026 key punch has 48 different characters -



## CLIENTS ROOM

There is now one 026 key punch and one 029 key punch located in the Clients Room. FORTRAN programs may be punched on either key punch. Other cards for the GE 225 should be punched on the 026. Cards for the PDP 10 should be punched only on the 029.

## SEMINARS AND PROGRAMMING COURSES

### SEMINAR: *Multiprocessing Concepts*

The speaker will be Mr. Ian Oliver, Lecturer in Computing at the Computer Centre. The seminar will be between 12 and 1 p.m. on Wednesday, 2nd October, 1968, in Room B18 of the Engineering School, Circular Drive.

Many unique problems arise in a computer system which has more than one processor. The problems of scheduling and allocation of system resources will be discussed. The terms multiprocessing, multiprogramming, time-sharing and time-slicing will be defined and some of their ramifications considered.

For further particulars concerning this seminar, please contact the Secretary, Computer Centre, extension 688.

## PROGRAMMING ADVICE

The following points refer to FORTRAN IV, for the GE 225:

### DATA STATEMENT

The combination of characters "./" is unacceptable in a DATA statement, i.e. never use a statement such as:

```
DATA A,B,C/4.1,3.67,5./
```

Correct usage would be:

```
DATA A,B,C/4.1,3.67,5.0/
```

### PUNCHING CARDS

When running a program that punches cards, avoid possible mispunching of information on the last card by using the library subroutine EXIT to end the program, i.e. CALL EXIT should be used as the last executable statement.

## DØ LOOPS

If an inexplicable loop occurs in loading a program, then possibly there is a DØ loop in the program which ends on a FORMAT statement.

## "TOO MANY LOOPS" MESSAGES

When a bracket is left off an implied DØ loop in a READ or PRINT statement, all implied DØ loops and DØ statements in the rest of the program will have the error messages 30 (Too many implied DØ's in an I/O list) and 61 (There are too many DØ's) respectively, although they are probably not in error. There may or may not be an error on the incorrect statement.

## USE OF \$BINARY

A \$BINARY control card must always be used in front of a binary deck (or group of decks) even if the NOSOURCE option is used on the \$JOB control card.

## SYSTEM MODIFICATIONS

### OVERFLOW SUBROUTINE - FORTRAN IV

An error in the FORTRAN IV library subroutine OVERFL has been discovered and corrected. Previously a call to OVERFL did not reset the Overflow Hold Indicator, if it was set. Thus once the indicator was set it would be detected as set on all subsequent calls to OVERFL. This error has now been rectified. If the indicator is on, the routine returns a value of 1 and resets the indicator, if not, a value of 2 is returned.

## LIBRARY PROGRAMS

### NEW

Two new FORTRAN IV subroutines are available for finding the eigenvalues and eigenvectors of any real matrix. Both of these routines use variable dimensions.

EIGEN - Eigenvalues, Eigenvectors (D4.269)

This subroutine calculates the eigenvalues and, optionally, the

eigenvectors of any real matrix. The routine leaves the eigenvalues and eigenvectors stored in a format rather difficult to access so, unless storage limitations dictate otherwise, the user may prefer to use the subroutine ZEIGEN in conjunction with this routine.

ZEIGEN - Eigenvalues, Eigenvectors (D4.270)

This subroutine uses EIGEN and returns the eigenvalues and, optionally, the eigenvectors of any real matrix stored in a more convenient form than that of EIGEN.

If the matrix is symmetric, the user may find the subroutine HDIAG (D4.269) a little simpler to use.

### MAC, THE MATRIX COMPILER

The programming language MAC was designed and implemented on the GE 225 by Messrs. I. Oliver and E.J. Sokoll in 1964. It greatly facilitates the programming of problems involving such matrix operations as inversion, multiplication, and transposition. MAC is an extremely easy language to learn and the programming effort for matrix calculations is commonly reduced by as much as 95%.

Example 1 shows a sample MAC program which reads from cards two matrices, A and T, and prints the matrix  $T^{-1}AT$ .

```
READ MATRIX A,T
I = / T
J = I*A
P = J*T
PRINT MATRIX P
END
```

#### Example 1 Sample MAC Program

A useful feature of the MAC language is that as soon as a matrix has been referenced for the last time in a program, the portion of memory which it occupied is made available for other arrays.

Complete documentation on the MAC system can be obtained from the Administrative Officer.

### THE PERILS OF LEAST SQUARES - I. Oliver

The technique of "least squares" is used extensively in a wide variety of problems where it is desired to "fit" a mathematical model to a physical situation. Suppose, for example, it is required to find a subset of the variables  $x_1$ ,  $x_2$ , and  $x_3$  which "predict" a dependent variable  $y$ . A common procedure is to assume a model of the form

$$y = a_0 + a_1x_1 + a_2x_2 + a_3x_3 \quad (1)$$

and to apply least squares to find the coefficients  $a_0$ ,  $a_1$ ,  $a_2$ , and  $a_3$  which give the "best" equation for a set of experimental data.

Nearly every computer centre has a multiple linear regression program for handling this sort of problem but a user must be very careful in interpreting the results from such an analysis. It is particularly easy to be misled by the masses of figures which typically pour out from these programs. The user should very carefully check the reasonableness of all output from a least squares program.

There are two common criteria by which an equation fitted by least squares may be judged. The two criteria are (i) explanation of as much of the variation in  $y$  as possible, and (ii) obtaining a set of coefficients which are all statistically significant.

The simplest way to obtain an equation meeting the first criterion is merely to use all of the independent variables upon which measurements are available, but this is computationally a very dangerous procedure. Many programs simply invert a matrix (the matrix of sums of squares and products, or the correlation matrix) to achieve this result and such programs should be avoided, unless the user is certain that he has a well-conditioned problem. Matrix inversion is a tricky business and many standard subroutines will sometimes produce meaningless results which look almost reasonable.

The conditioning of data for a least squares program depends on the correlations among the variables. In general, if the independent variables

are highly intercorrelated then difficulties will occur in the matrix inversion. But this is not always the only troublesome case and it is generally impossible to predict trouble by mere inspection of the correlation matrix. One good way in which to test the accuracy of the inverse of a matrix is to invert the inverse, hopefully producing the original matrix. One standard program tested was so bad that the inverse of the inverse did not have even the sign of any element correct. The equation obtained from this run predicted even the signs of the dependent observations incorrectly. This test was on a problem with only three independent variables.

In practical terms the difficulty experienced in the above test was caused by the non-uniqueness of the predicting equation. An equation involving just the first two or in fact, any two of the independent variables would have been just as good as an equation involving all three variables. In other words the third variable contained no additional information over the information contained in the first two variables.

In mathematical terms the matrix of sums of squares and products in the above test was exactly singular with rank 2. But the first two pivots (using Gaussian elimination) disturbed the singularity and enabled a third pivot to be carried out resulting in a meaningless "inverse". Grossly inaccurate results can also be obtained with nearly singular matrices.

The user is therefore recommended to use a stepwise regression program. Such programs usually check fairly closely on the problem of singularity and only fit as many of the independent variables as can be done safely. The user is cautioned that the resulting equation may not be the only possible equation. For example, if  $x_2$  and  $x_3$  are very highly correlated a stepwise program will produce an equation with  $x_1$  and  $x_2$ , or  $x_1$  and  $x_3$ , but not  $x_1$  and  $x_2$  and  $x_3$ . If the user expects, a priori, certain variables to be in the final equation he should use the option, which is usually available, to force the selection of these variables ahead of any others.

A second criterion which could be used as a basis for determining the final equation is the significance of the individual coefficients. This criterion is not the same as the previous one. Indeed, it could happen that in an equation selected according to the first criterion, none

of the coefficients may be significant at a given significance level.

In general, with the second criterion a smaller fraction of the variance in  $y$  will be explained since the user is protecting himself against coefficients which may be non-zero due solely to random fluctuations in  $y$ . This approach is, therefore, rather conservative. Again, however, the final equation may not be the only possible equation which could be derived from the data.

It is a sound practice to obtain the residuals, i.e. the deviations of the predicted  $y$  values obtained from the final equation, and the observed  $y$  values. If there are systematic variations in the residuals then the linear model does not fit the data too well. For example, there may be a number of positive residuals followed by a number of negative ones, etc. Autocorrelation may be present, or perhaps a curvilinear or non-linear model is required.

Finally a note of caution to users who write their own least squares programs. In matrix terms the general linear model is

$$Y = Xb + e \quad (2)$$

where  $Y$  is a vector of observations on the dependent variable,  $X$  is a matrix of observations on the independent variables,  $b$  is a vector of unknown coefficients, and  $e$  is a vector of errors or disturbance terms. The least squares vector  $\hat{b}$  is given by solution of the equations

$$X^T Y = X^T X \hat{b} \quad (3)$$

The solution of these equations may be a treacherous business due to loss of significance and other numerical problems. A matrix inversion routine should not be applied to  $X^T X$  unless it is known that the matrix is well-conditioned and of full rank, or unless the routine is designed to take special action on nearly singular matrices.

It is worthwhile to standardize equations (3) in the following manner. Let  $D$  be a diagonal matrix of the square roots of the diagonal terms of  $X^T X$ . Then

$$\begin{aligned} D^{-1} X^T Y &= D^{-1} X^T X D^{-1} D \hat{b} \\ &= D^{-1} X^T X D^{-1} b^* \end{aligned} \quad (4)$$

where

$$b^* = D \hat{b} \quad (5)$$

The diagonal terms of the correlation matrix  $D^{-1}X^T X D^{-1}$  are all unity. Equations (4) should be solved for  $b^*$  and the trivial equations (5) then solved for  $\hat{b}$ .

When applying Gaussian elimination to (4) a check should be made on the size of each diagonal element before it is selected as a pivot. The diagonal elements should always be in the range zero to one and a pivot less than a small number  $\epsilon$  (e.g.  $10^{-5}$ ) should never be chosen. In other words only a submatrix of the full matrix is actually inverted. This is equivalent to arbitrarily setting some of the coefficients in the original model (1) equal to zero.