COMPUTER AIDED MANAGEMENT DECISION MAKING;

A PRACTICAL AND THEORETICAL STUDY

A Thesis Presented to

THE UNIVERSITY OF ASTON IN BIRMINGHAM

for

THE DEGREE OF DOCTOR OF PHILOSOPHY

by

DAVID MICHAEL SLINGSBY-PEACE

Interdisciplinary Higher Degrees Scheme
University of Aston in Birmingham

June, 1974

SUMMARY

The present research attempted to study some of the problems of achieving successful implementation of new computer-based system designs. In particular it considered the implementation of management information systems having a limited number of users. An examination of the literature led to the conclusion that an important requirement for successful implementation was that the expectations of the ultimate user should be taken into account.

Initially the research was delayed because of the lack of existing techniques that could adequately reveal such expectations and so enable them to be incorporated into the design process. After some time however, the technique of repertory grid testing, which is based upon personal construct theory, was discovered. This test was originally devised for use in psychotherapy but in the present study a test form was developed that was suitable for investigating managers' expectations. A preferred method of analysis was also developed.

In order that implementation could be studied in a realistic way, the research was carried out in a sponsoring Company for whom a small computer-based management information system was developed and implemented. The final form of the repertory grid test was used upon the managers who would finally use the management information system and it was found to be efficient in revealing their attitudes to, and expectations of, computer systems in the context of their own jobs.

More detailed investigation and development of the test was curtailed (a) by the considerable time requirements associated with developing a worthwhile and practical computer-based management information system having utility for its intended user(s) and (b) by the considerable changes that took place during the research period in both the Company's products with which the management information system was concerned, and in the managers associated with the system.

ACKNOWLEDGEMENTS

The author wishes to express his thanks to his supervisors Mr R. Acton, Mr. K. Bowcock, Mr. J. C. Watt and in particular to Mr. R. S. Easterby for their encouragement and assistance during the research and in the preparation of this thesis. In this context thanks are also due to Prof. S. L. Cook and Dr. D. J. van Rest, I.H.D. Tutor.

He also wishes to thank the subjects of the repertory grid tests for their co-operation in the research and to thank Mr. K. Eason, Department of Ergonomics and Cybernetics, Loughborough University, for suggesting in the first instance that repertory grid methodology might provide the means the author was seeking for investigating the problems of implementation of computer-based management aids.

Individual thanks must also go to Dr. D. A. L. Wilson, Mathematics Dept., University of Aston, both for his tutorial assistance in the analysis of time series and for his lecture series on the same subject, which were invaluable in carrying out the forecasting work described in Chapter 4.

The author wishes to express his gratitude for the many discussions with his former colleagues at the sponsoring Company and the general interest shown by them. In particular, thanks are due to Dr. D. B. Gilding, formerly Computer Systems Manager at the Company and also industrial supervisor to the research.

Finally the author wishes to thank the Directors of the sponsoring Company for financially supporting him during the course of his research.

CONTENTS

Page No.

PART I: O.R./SYSTEMS WORK IN THE SPONSORING ORGANISATION

Part I is concerned with O.R. and systems work carried out at the sponsoring Company. After a brief introductory section describing the Company and the nature of the scheme under whose provisions the research was carried out, the problem is described and the means by which a solution is currently achieved outlined. The status of demand as a major variable is described and the work carried out in forecasting demand is indicated, together with the results and their relevance to the present manual stock control system. The latter is described and details of a computer based system intended to achieve the same objectives more efficiently, are given. The results of the computer based model using real data are presented, followed by a critical comparison of the two systems.

1. INTRODUCTION

In section (1.1) the sponsoring organisation is described; its history, its organisation and its products. In section (1.2.) the I.H.D. scheme is briefly described, and in particular its basis and objectives are outlined. The latter are important because they have significantly affected the way in which the research programme was organised. Finally the research objectives are given and discussed (1.3.).

1.1 THE SPONSORING ORGANISATION

1.1.1 Background

The original Company was founded in 1909, specialising in copper tube production and employing about 100 people. It subsequently began to produce tubes made from non-ferrous alloys and by the end of the 1939-45 war had diversified its products considerably, supplying the aircraft, shipping and munitions industries. These products continued to be based on non-ferrous metals and their alloys. During the late 1950's the Company merged its interests with a similar organisation and at the time of the research 10 years later (1969-71), the new Company had become a wholly owned subsidiary of a large metal producing and manufacturing organisation.

1.1.2 Company Organisation

At the time of the research the affairs of the Company were controlled by a board of directors, the managing director also having a seat on the board of the controlling company. Reporting to the board were a number of executives who each held divisional responsibilities. A number of senior managers reported to each executive. The Company as a whole employed approximately 7,000 people.

Geographically the Company had four major manufacturing sites situated respectively in the midlands, the north west, Scotland and Yorkshire where the headquarters were established. These sites supplied a network of warehouses within the U.K. Additionally a number of subsidiary companies were owned which were fairly autonomous in their operation. Companies established on the Continent of Europe and in Australia again operated virtually independently of their U.K. parent.

1.1.3 Company Products

The company manufactured four major product groups, described below.

(i) <u>Domestic Tubes</u>

This group contained all those products used in domestic water supplies and services. All the products were made from copper (rather than alloys of copper) and they were made to stock sizes described in British Standard Specifications.

(ii) <u>(a) Domestic and (b) Engineering Fittings</u>

(a) The products in the domestic sub-group again found their major outlet in domestic water supplies and services. The fittings were used to join separate lengths of tube, or typically, they were the component parts of stopcocks, radiator valves and drainage systems. Copper and alloys of copper were used in the manufacture of these items.

(b) Engineering fittings had a similar function to their domestic counterparts but were usually larger, stronger and were made from copper alloys. They were used in large engineering plant (eg power station water cooling systems) and process plant (eg sugar evaporators).

(iii) <u>Engineering tubes</u>

As might be inferred, these products found a diversity of applications in the engineering world. Two major outlets were marine engineering and electric power generation. In both cases the tubes were mainly used in water cooling systems. They were usually made from copper alloys for better corrosion resistance and greater strength.

(iv) Plates

This product group comprised hot-rolled copper and copper alloy plates with thicknesses in the range of approx. $\frac{1}{4}$ inch to several inches, depending on the properties, overall dimensions and total weight required. After machining and drilling most plates were used as end-plates in tubular heat exchangers.

The production of plates usually made a slight financial loss. However it was generally considered to make an indirect contribution to profits because it enabled the Company to offer a complete materials supply service to heat exchanger manufacturers.

A financial picture of the relative importance of each of the above product groups may be obtained from Table (1.1.).

1.2 THE I.H.D. SCHEME

The research described in this thesis was one of the first to be registered under the above scheme. Post graduate interdisciplinary research as defined in the I.H.D. scheme has significantly different requirements to those expected from traditional postgraduate research. These have materially affected the way the present work was carried out.

Prof B. H. Flowers (1969), late chairman of the Science Research Council, referred to so-called "Swann-awards" made by S.R.C. (Swann-awards supported the early I.H.D. students) in the following way; "We are keen, in particular, to see whether it is possible to develop a training in science and technology as intellectually challenging to the most able students as the Ph.D. The traditional Ph.D. as a training for research must show a student how to penetrate in depth at a particular point in his subject albeit within a wider context. But

TABLE (1.1) TYPICAL ANNUAL PRODUCTION STATISTICS

Product Group	Turnover (£m)	R.O.I. %	Production (tons)
Domestic Tubes	16.7	14.5	23,760
Fittings	9.8	23.5	60.1 million fittings
Engineering Tubes	18.1	5.5	23,629
Plates	1.4	-	2,018

to deal with many problems in industry or administration, what is needed is a capacity to grasp essential facts and ideas over a very wide area, and by seeing in depth the inter-relations between them to plan the right action". On the basis of this analysis, the traditional Ph.D. student is concerned with a study in depth at a particular point, although he is expected to set his work within a broader context. On the other hand a student following a new type Ph.D. has to abstract facts and ideas over a wide area and obtain a fundamental understanding of inter-relations between them. In this research the mathematical/logical procedures of O.R. and system analysis were used to resolve certain technical problems before further investigation led to the fields of Applied Psychology and Psychotherapy, where a theoretical basis was found from which to attack a more fundamental difficulty remaining in the original problem area, that of allowing for the man in the system.

Since the purpose of interdisciplinary research is to provide training that is subsequently of relevance to industry and administration, provision is made for students to work closely with a sponsoring organisation. He is required, in fact, to spend a minimum of 30% - and up to 70% - of his time with the sponsoring firm, or organisation (I.H.D. Scheme: Notes for Guidance, 1971). Compared with traditional research, the broader approach of an interdisciplinary study clearly has implications for the kind of work carried out; nevertheless the provision that a significant amount of time be spent within the sponsoring organisation had greatest effect in much of the planning and execution of the research programme. As will be seen, the fortunes of the sponsoring organisation can become inextricably intertwined with the research to the extent that events occuring within the organisation, which themselves have no direct bearing on the research, can seriously affect the work taking place.

1.3 RESEARCH OBJECTIVES

At the time the broad research objectives were being discussed, a number of new computer systems had just been introduced in the sponsoring company. However, without exception these were large scale data processing applications such as payroll, sales order processing, stock control etc, and the Computer Systems Manager felt that some consideration should be given to providing computer-based support to decision making at senior management level. The idea of corporate modelling was examined for some time, but a number of difficulties were evident, not least of which was the magnitude of such a task, given the limited resources available. However a more fundamental problem was foreseen. It was felt that the general approach of corporate modelling would not be compatible with the existing management style within the Company.

This was based on the fact that there had been attempts to introduce such schemes as simulation for more efficient foundry scheduling, linear programming for optimal draw-bench loading, statistically controlled customer service levels in stock control and exponentially smoothed forecasts of machined components for small assemblies. Thus there was little to suggest that the Company lacked either models or methodologies with which to tackle its problems. However these had all received a lukewarm reception at best from senior management, despite significant advantages being demonstrated for each of the suggested schemes, and none had been integrated into production procedures. Thus a situation existed where a number of schemes having demonstrable benefits for the Company had not been adopted - or even implemented on a trial basis in some cases. (It might be thought that some practical difficulty or high capital cost prevented implementation in these examples. However in each case sufficient was known of the circumstances to conclude that the main cause of failure to implement centred upon lack of management enthusiasm to do so).

The foregoing suggested that one useful line of research would be to consider just how successful implementation of schemes based on new principles and methods could be achieved.

Although there appeared to be resistance to the introduction of management science methodology among senior management, equally there was a clear requirement for the use of modelling and other techniques of operations research at that level. It was decided therefore that some operations research work would also be carried out with the intention of aiding decision making at senior management level, but that effort should ultimately concentrate on the problems of implementation.

However, preliminary searches of the literature showed that,at that time, virtually no work had been published concerning implementation, with the exception of a limited amount of case study. Therefore, from a practical point of view, it seemed expedient to commence the overall research by carrying out operationsresearch on decision making. Concurrently, the problems of implementation would be considered, leading hopefully to the development of some technique of systematically investigating the factors concerned in introducing new methods. If time permitted the results of this latter work could be compared with the observed reactions of senior management to the straightforward operations research work carried out earlier.

2. A PROBLEM AREA WITHIN THE ORGANISATION

The practical outcome of O.R. and systems analysis within the sponsoring company was a simple system that, in the words of the executive involved, "begins to tackle a long outstanding problem" in planning production resources and balancing stocks of copper tubes. The purpose of the present chapter is to introduce this part of the work.

In 2.1. the criteria used in selecting the problem area are presented. The problem is described and is placed in the context of the total picture of Company production activities. Its essence lay in balancing production against demand for various products in order to achieve reasonably stable production levels without holding excessive stocks. The associated organisation structure is given in 2.2., which contains sub-sections briefly describing the amount and quality of marketing information available concerning future demand (2.2.1), the individual responsibilities held and the management controls available to carry out the function described in 2.1. (2.2.2.), and the Company stock holding policy (2.2.3.).

2.1 SELECTION AND DESCRIPTION OF PROBLEM AREA

After discussion with the Computer Systems Manager at the Company it was decided that the most flexible way to provide an aid to management decision making would be to build a "what-if" simulation model of some relevant part of the company. This would enable the manager in charge to explore the consequences of his decisions in some detail before actually putting them into effect. The following criteria were applied in selecting a problem area;

(a) the area selected should have importance within the scope of the company's activities.

(b) a significant amount of historical data should be available on which to test the model

(c) the manager of the area selected should be interested enough to actively collaborate

(d) the manager should be senior enough to implement the results of the study.

On the basis of the above, the area of production planning and allocation was selected. Thus the problem was concerned with a very important part of the Company's activities, that of domestic tubes (see Table 1.1.). Furthermore there was a reasonable amount of historical data available. The manager of the Central Allocation Department (C.A.D.) met criteria (c) and (d) since he had already given assistance when the notion of a company model was being discussed. In terms of seniority, he was promoted to executive status during the course of the research.

The purpose of C.A.D. was to allocate production facilities to orders. Orders could be raised in two ways, by a customer or internally by C.A.D. itself. Orders raised by customers were designated 'bespoke' orders and were allocated production plant as it became available. Bespoke work was often related to large engineering projects and orders were frequently placed for delivery at various intervals over a period of time, so giving rise to a 'forward order book'. Although forward orders often accounted for a large proportion of production capacity, it is interesting to note that it had never been company practice to accumulate such orders as they were received, for the purpose of production planning. In fact, such a procedure was being devised at the time the research ended.

In contrast, domestic tubes were made for stock as opposed to customers' orders. Since domestic tubes shared a common production route with much of the bespoke engineering tube work, it competed with the latter when production runs were being scheduled. However this was turned to advantage by using the volume of domestic tubes produced as a regulator for the level of overall production activity. Thus when bespoke work was in short supply more domestic tubes than required in the short term were made for stock. When demand for bespoke work was high only part of demand was met by production, the remainder coming from stock. This process and the reasons for it are described in Chapter 5.

In principle the job of C.A.D. was a simple one. Forecasts of future demand for domestic tubes were supplied by Marketing Department. These had to be balanced against spare production capacity and a decision taken to place a production order with one or more of the manufacturing sites. Deviations from the forecast, and failure to achieve requested production levels were monitored weekly and if the deviations became significant, new production orders were arranged.

In reality, however, a number of serious practical difficulties existed.

The C.A.D. manager did not have full control over the situation for which he was responsible. In the first place, the forecasts of demand he received were not reliable and secondly he could not direct the level at which production was pitched. Finally he was not responsible for setting stock levels or controlling stocks in financial terms. These aspects are amplified below.

2.2 ORGANISATION STRUCTURE WITHIN PROBLEM AREA

2.2.1. Marketing Information

Marketing Department were able to produce forecasts based on two different approaches. The first of these, described in 4.1., was an econometric method and used the fact that most of domestic tube production was used in new house construction. It provided a forecast of quarterly demand for the next 18 months. The second was a "guesstimate" which depended on the experience of the forecaster and was based on the information he gleaned from the building industry. This latter prediction was available approximately monthly and attempted to indicate likely demand for the next 5-6 weeks.

In fact the econometric approach was not used for reasons that were quite independent of the validity of the method. (The originator of the econometric forecast undertook new responsibilities and his successor did not continue the system). Thus the formal basis for production planning was the Marketing Department prediction.

2.2.2 Managerial Controls and Responsibilities

Although the prediction of demand tended to be unstable (it was often necessary to seriously modify it part way through a planning period — three months), superficially at least, this was not too serious because the C.A.D. manager was not constrained to rely on it.

In fact the C.A.D. manager seldom, if ever, accepted the Marketing Department prediction. On the basis of their prediction and his own intuition he would make his own estimate of likely demand and use this as the basis of his production orders. Actual production orders were made on a quarterly basis in the form of "production budgets". Each manufacturing site would agree during discussions held before the start of a planning period to produce specified constant weekly amounts of certain products. These amounts constituted the weekly production budget. The size of the budget would depend upon expected demand for domestic tubes, existing stock, the amount of bespoke work scheduled and the likely availability of production plant and labour.

However the amounts and proportions of products made often deviated greatly from that budgetted for. Ultimate responsibility for deciding which items should be produced rested with factory (site) managers and their production staff. If starting stock or tools were not available for bespoke work, rather than let production equipment lay idle, domestic tubes would often be made for stock. It was also common practice to redeem spoiled bespoke work by processing it to domestic tube sizes. Equally, if bespoke work had been delayed, production of domestic tubes might be cancelled or severely cut back. While these practices were consistent with the Company's policy of using domestic tubes for 'regulatory' or 'load-balancing' purposes, the degree of action taken was not controlled by the C.A.D. manager, who was responsible for overall company domestic tube stock holding, but was decided by factory managers for their own local reasons.

Some of the results of this situation are presented later, but it may be imagined that the system easily went out of control and once this had happened there were no direct means whereby the C.A.D. manager could regain control. In fact during the course of the research, the domestic tube situation was always at or near crisis level. Just before the end of the research period a reorganisation took place and this may have improved the situation. It meant that the factory managers responsible for producing domestic tubes would report to the same director as the C.A.D. manager. (The C.A.D. manager, incidentally, had nominally similar status to the factory managers). This should enable some of the conflicts to be resolved.

It had been apparent from an early stage in the research that reorganisation of responsibilities of the sort mentioned above would ameliorate or remove many of the problems of C.A.D. However it was decided that these were beyond the scope of the project and that any improvements that were devised during the study would have to accommodate existing organisational difficulties.

2.2.3 Company Stock Holding Policy

As already indicated, the C.A.D. manager was also responsible for stock control of the domestic tube product group. Work carried out as part of this research to help in the tasks of stock control and production allocation, may best be appreciated by some knowledge of the systems of stock control existing in the company at that time. The procedures described below are those in force at the end of the project. Most of the computer systems mentioned were only being developed when the research began.

(i) Domestic Fittings

This was the most sophisticated system. Orders received at sales offices situated in the various regions of the U.K. were duplicated; one copy was sent to the nearest Company warehouse and the other was sent to headquarters and was entered on a daily run of the stock control computer program. The stock control program issued automatic replenishments for 3,800 items monitored. A list of supply commitments was compiled from the orders, the commitments being diminished by despatches. Although a daily run of the stock program was made, only certain warehouses were considered each day, this being done on a rota basis. In practice each warehouse was monitored once each week. In addition to known commitments, a smoothed (simple exponential) forecast without trend was made of demand. The list of commitments in combination with the forecast enabled weekly production orders to be raised. These orders were placed on the basis of Company stock levels. In practice little attention was paid to the tracking signal, manual scanning of the computer print-out of stock position apparently being preferred.

Domestic fittings were manufactured at headquarters who supplied stock to the warehouses and held a supply of non-warehouse stock items. Production lead times varied typically in the range 6-15 weeks.

(ii) Engineering Tubes

These were not stocked at depots, only at headquarters. Strictly speaking engineering tubes were only made to customers' order but often a small stock of common sizes of tube was held as a result of production over-runs or possibly because an order had been completed early. In the latter case the material might then be used for a subsequent order and the batch re-made to satisfy the original requirement.

A computer system for accumulating orders, similar in many respects to that described under (i) above, was introduced about the time the research ended.

(iii) Domestic Tubes

Fifteen warehouses were involved in this stock control system. However none of the stock was computer controlled. One man based at headquarters surveyed the stock situation and arranged for replenishments to be supplied. These arrangements were made by both telephone and written request. As already described, C.A.D. produced production requests consistent with the total Company stock situation and the load balancing function of domestic tube products.

Orders placed at sales offices were processed by the Sales Order Processing (S.O.P.) computer program suite in order to produce, among other things, invoices for the customer. As the information passed through S.O.P. it was picked up by the stock control program together with production receipts, so enabling a computer generated listing of stocks to be produced. No attempt was made to control stocks from this, for various reasons, as follow. At peak times, S.O.P. could be as much as three weeks behind in processing orders. Furthermore C.A.D. did not accept the relationship,

$$\text{stock produced} = \text{production receipts} - \text{despatches}$$

which they had at one time used. This practice had been discontinued when investigation had shown that apparent stock deficiencies were being created, because at any one time 60 - 100 tons of material were being moved between sites and thus were not available for issue. They had come to rely on a weekly "shelf count" carried out by warehouse staff to obtain a real measure of availablestock. The computer listing of stock mentioned above was therefore discounted by C.A.D.

Despite the fact that copper cost £500 - £700 per ton, occasionally rising considerably higher, the C.A.D. manager had no financial guidelines within which to operate. Minimum Stock levels were issued by the director in charge but these, as far as could be ascertained, were not based on such factors as variability of demand, production lead time and mean demand, as might be expected. In fact the Company did not accept the principle of offering less than 100% level of service to its customers despite the statistically variable nature of the production processes concerned. In consequence the C.A.D. manager and his staff appeared to spend quite a lot of their time trying to avoid stock-outs by diverting production material to deficient product lines where possible and making inter-warehouse transfers. Further mention of domestic tube stock holding procedures is made in 4.4.

3. INITIAL WORK

The outcome of initial work is described and it is shown how these led the original objectives to be modified. In particular, this involved a shift of emphasis from strategic to tactical decision making. Changes in individual status arising from Company reorganisation were instrumental in encouraging this action.

The value of a "what-if" simulation model was discussed in a series of meetings with the C.A.D. manager, who said that any aid to planning would be a great help in the situation existing at that time. However he was unable to indicate how he might use such a model. The apparent difficulty was that no model was available to form the basis of discussion and any serious consideration of potential applications soon halted for want of detailed information. Given the manager's interest and co-operation it seemed that this problem could be overcome by building a general but relatively simple simulation model of the area for which he was responsible. This would give hard information on which to base further discussion. It would also permit some simple demonstrations of how such models could be used in practice and this in turn was expected to enable the manager to suggest more realistic situations for evaluation.

Further discussions were held with the manager during which the information given in section (2.2) was obtained, together with more detailed information on production strategies and stock holding practices. A large amount of historical demand and production data was also examined and incorporated in the computer simulation model described in Appendix 1. The model was able to simulate on a weekly basis the whole of the domestic tube production activity and could have been used to evaluate the effect of various parameters on this activity. However it was never used for this purpose.

It had become increasingly obvious during the period of model development that the manager was ill at ease with the concept of modelling. It had also become apparent that the procedure used for planning production was far more ad hoc than had been supposed during initial discussions. Upon further reflection it seemed that a considerable amount of effort had been expended in producing an aid to management decision making that relied upon a concept that was alien to the intended user and utilised planning procedures that were unfamiliar to him. (In fairness to the work carried out, these procedures only made explicit what was already general practice, although this was not readily perceived by those concerned). It was realised at this stage that the prospect of successfully implementing a system of "what-if" modelling was bleak indeed.

Although this conclusion was very disappointing, even with the benefit of hindsight, it was difficult to see how the situation could have been avoided. A considerable amount of discussion had taken place with the manager before work began, but at that stage lack of hard detail had obscured the true reality which was that the manager did not comprehend his area of responsibility in terms of the research suggested. Only through insight gained by working closely with him did the real situation become evident, by which time a significant amount of effort had been expended. If the work described under Part II herein, could have been carried out before the project design had been formulated, this wasted effort might well have been avoided.

At this stage a number of changes took place within the Company that affected the course of the research. The C.A.D. manager was promoted to executive status and undertook wider responsibilities. He nominated his deputy to continue with direct participation in the research, as his own commitments would not permit continued involvement at a detailed level. While the deputy was much more familiar with modelling and computer-based planning procedures, he was quite unfamiliar with the particular planning procedures employed within the domestic tube product group. Lack of formalised decision rules made life difficult for him and was inconsistent with his personal approach to management. After considering the simulation model, he concluded that it would be difficult to have confidence in the results of simulation studies based upon it because the formal allocation procedures it embodied had not been used in practice. Although this was not strictly true - the rules used by the model were essentially those then being used - the conclusion was accepted because the old C.A.D. manager had never used the rules in more than a broadly systematic manner, while the model would apply its decision rules in a far more rigorous fashion. The deputy suggested instead that given the then present state of the planning function, it would be of greater benefit to him if an aid to weekly planning were produced. The procedures developed in this latter exercise could then be used with confidence in any subsequent simulation studies. Such an approach would have his strong support as it would be of considerable help in his day to day work. This was put to the old C.A.D. manager who still retained overall responsibility for production allocation. He too saw greater merit in the new suggestion, that a computer-based weekly planning procedure be developed. Thus the work described in subsequent chapters relates to a planning method for week by week monitoring of demand for products and adjustment of production quantities required. Ironically the man who suggested this approach moved within a few weeks to another position within the organisation. Fortunately, his replacement, who stayed in fact for the remaining duration of the research, was a man of similar outlook who helped to implement the system and use it on a weekly basis.

This chapter has referred to initial work (Appendix 1) carried out in the course of the research. Its product, a simulation model intended for application in evaluating new management policy was never used. The reasons for this have been described, but essentially the original objective had to be modified because the projected system could only be usefully used if supported by existing planning systems operating at a tactical decision making level. Although initial investigation had indicated that these systems did exist, experience later showed that this was not so. The then current planning procedures were remarkably unstructured and provided no basis for the approach originally envisaged. A suggested alternative was to improve these latter procedures.

Although a significant amount of effort had been expended on simulation work by the time it became clear that the tactical planning procedures would have to be improved, fortunately the work was by no means wasted. A close examination of the existing planning procedure had been made and its essential features had been established. Furthermore a great deal of historical demand and production data had been assembled to produce distributions upon which Monte Carlo simulation could be based. The range and variability of these data had given insight of the lack of control that existed in the current planning procedure.

Finally, perhaps the most important point this episode underscored was the difficulty of objectively assessing an existing system on the basis of simple interviews, no matter how helpful the subject.

4. FORECASTING DEMAND

An important factor in achieving a solution to the foregoing problem lies in obtaining good forecasts of demand. Previous work carried out by Marketing Dept., is summarised in 4.1. However this work was restricted to forecasting aggregate demand, while a major aspect of the allocation problem is deciding how aggregate demand will sub-divide among the various products, since this is a major consideration in planning the product mix. Arising out of the order entry system currently in use at Y.I.M. are difficulties in operationally defining "demand". These are described in 4.2. together with the definition ultimately adopted for this project. Section 4.3 presents the relevant theory and results obtained using three forecasting models with past demand defined as in 4.2. The reasons for choosing these models are given in 4.3., the results are discussed and the best of the three models considered is indicated. In the final section (4.4), the Company's stock holding policy is compared with results obtained from simple inventory theory, that is, that stock levels may be related to average demand, lead time and variability of demand.

4.1 PREVIOUS FORECASTING WORK

The theory of demand is fundamental to economic analysis and is well developed. In practice however as, for example, Baumol (1961) points out there are many pitfalls for those who seek to apply the theory. Economists define demand for a particular item at a chosen point in time, as the quantity of the item that would be purchased at that time given its price, the availability of substitutes at competitive prices, and taking into account any other factor that would affect individual preference for the item being considered. Although this is perfectly adequate at the theoretical level, there are certain obvious difficulties in operationally defining demand. In general it would be very difficult, if not impossible, to rigorously establish a comprehensive list of factors affecting individual preference, and even assuming this could be done, it would be at least equally difficult to establish how these factors affected demand as their influence varied with time. Nevertheless manufacturers have a great interest in the notion of demand curves for their products as these provide a rational starting point for decisions such as, the volume at which to pitch production, the product mix, the price to be charged for products and hence the profit available.

However, since the variables of the demand function for a particular product usually cannot be reliably deduced by theoretical means, in practice it is often necessary to attempt an empirical determination of demand relationships in order to provide a basis for decision making. Such approaches usually try to produce a demand function by analysis of historical data from which, after making certain assumptions, it is then possible to attempt prediction about future demand.

A limited amount of work had been done in this respect by the Marketing Department of the sponsoring Company. Numerous mathematical techniques (including moving averages, exponential smoothing, Box-Jenkins γ models and linear regression models) had been used in an attempt to forecast future total demand for domestic copper tubes. The method which gave least error in its predictions is described briefly below.

Thus it was found that in the case of domestic tubes, total national sales (I) in tons were given by the regression equation,

$$I = 1320.7 + 56.06\ I_H + 6.22I_{HP} - 963Q_1 - 1939Q_2 - 2134Q_3$$

(Pearson's product moment correlation coefficient = + 0.82)

where:-

I_H = Investment in housing in the private sector.

I_{HP} = Investment in housing in the public sector.

Q_1, Q_2, Q_3 are seasonal dummies taking the value unity in quarters 1-3 respectively and zeroelsewhere.

This equation could forecast demand reasonably reliably up to six quarters ahead, which is the limit imposed by the reliability of forecasts for I_H and I_{HP} published quarterly by the National Institute of Economic and Social Research.

In the future Marketing Department hoped to consider two further approaches,

(i) to introduce time trends into the regression equation, where this was appropriate

(ii) to sample a cross-section of users and attempt to make forecasts of their requirements. The total demand would be obtained by summing the individual forecasts and estimating demand from the resulting population.

Having forecast national demand, Company demand could be estimated with knowledge of the Company's market share. This varied in the range 48-52% and was fairly stable. This stability was greatly aided by the fact that a trade cartel existed where prices were fixed by the industry for stock items. In fact during the course of the study, one of the manufacturers in the cartel broke the price fixing agreement and the sponsoring Company's market share fell to 26% during the first quarter of the new situation. The final outcome of this development is not known. Clearly the value of the above equation would be diminished if market share could not be reliably predicted. (Market share could be established fairly accurately as all the major manufacturers sent returns of sales to the British Non-Ferrous Metals Research Association of which they were members). Nevertheless this method of forecasting future demand for domestic tubes appeared to be a great improvement over the usual practice which were based on intuitive guesswork.

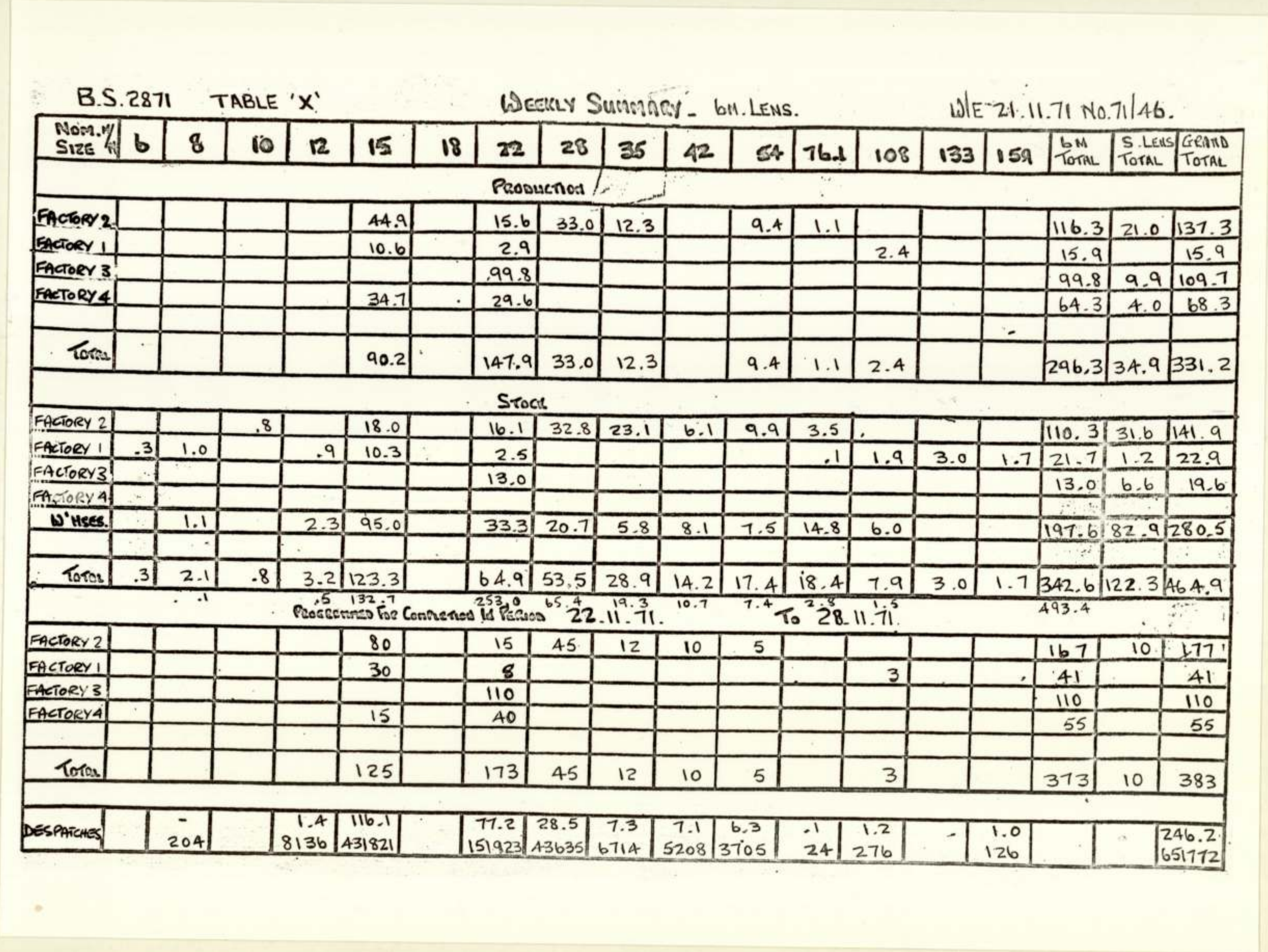
4.2 DEFINING DEMAND FOR INDIVIDUAL PRODUCTS

The method of forecasting devised by Marketing Department was concerned with a forecast of total demand for products in the domestic tubes product group. The reason for this was twofold. Firstly, Marketing Dept. required the information for other purposes, such as financial planning where a high level of aggregation was desirable and secondly, the information in this form corresponded with that already used by C.A.D.

However, C.A.D. used aggregate estimates of future demand by necessity rather than by choice. The problems of forecasting future demand are explained in detail in 4.3. Nevertheless, despite the fact that the domestic tubes product group contained 48 (later 45) individual products, plus a number of products of lesser importance, C.A.D. were able to control production in aggregate terms with some degree of success. This was possible largely because of the tolerance of the system to short term variations in demand for individual products, which derived from the regulatory or load balancing function of the product group as a whole. This latter function was mentioned in 2.2.2 and is described in detail in chapter 5.

However, during detailed production planning, provision had to be made for manufacture of individual products and this was done by the device of a production budget, also mentioned in 2.2.2. Unfortunately, during the period of the budget the vagaries both of demand and of production actually realized could cause embarrassing accumulations and depletions of stock to occur. Although stock was monitored on a weekly basis, the method of monitoring used by C.A.D. was inspection of a record sheet similar to that shown in Table (4.1) upon which the amount of stock of each of the three major product groups held was separately recorded.

The difficulty of estimating trends from such a series of tabular displays is readily apparent. This is particularly true when the values recorded contain a large random component, which was found to be the case in this instance. (See section 4.3 for detailed discussion). Thus, it was often the case that stock of a particular product would change quite



B.S. 2871 TABLE 'X' WEEKLY SUMMARY - 6M. LENS. W/E 21.11.71 NO. 71/46.

NOM. M/M SIZE	6	8	10	12	15	18	22	28	35	42	54	76.1	108	133	159	6M TOTAL	S. LENS TOTAL	GRAND TOTAL
PRODUCTION																		
FACTORY 2					44.9		15.6	33.0	12.3		9.4	1.1				116.3	21.0	137.3
FACTORY 1					10.6		2.9						2.4			15.9		15.9
FACTORY 3							99.8									99.8	9.9	109.7
FACTORY 4					34.7		29.6									64.3	4.0	68.3
TOTAL					90.2		147.9	33.0	12.3		9.4	1.1	2.4			296.3	34.9	331.2
STOCK																		
FACTORY 2			.8		18.0		16.1	32.8	23.1	6.1	9.9	3.5				110.3	31.6	141.9
FACTORY 1	.3	1.0		.9	10.3		2.5					.1	1.9	3.0	1.7	21.7	1.2	22.9
FACTORY 3							13.0									13.0	6.6	19.6
FACTORY 4																		
W'HSES.		1.1		2.3	95.0		33.3	20.7	5.8	8.1	7.5	14.8	6.0			197.6	82.9	280.5
TOTAL	.3	2.1	.8	3.2	123.3		64.9	53.5	28.9	14.2	17.4	18.4	7.9	3.0	1.7	342.6	122.3	464.9
		.1		.5	132.7		253.0	65.4	19.3	10.7	7.4	2.8	1.5			493.4		
PROGRAMMED FOR COMPLETION IN PERIOD 22.11.71. TO 28.11.71.																		
FACTORY 2					80		15	45	12	10	5					167	10	177
FACTORY 1					30		8						3			41		41
FACTORY 3							110									110		110
FACTORY 4					15		40									55		55
TOTAL					125		173	45	12	10	5		3			373	10	383
DESPATCHES		- 204		1.4 8136	116.1 431821		77.2 151923	28.5 43635	7.3 6714	7.1 5208	6.3 3705	.1 24	1.2 276	-	1.0 126			246.2 651772

TABLE 4.1

drastically before this was noticed and action taken. If a trend continued during the re-order lead time it was possible for a stock-out, or alternatively a considerable excess of stock, to be obtained simply because stock level rather than trend was being monitored.

The C.A.D. manager had perceived this problem and asked whether the new system being devised could incorporate some mechanism to detect the type of situation described above. Obviously some method of monitoring demand for individual products was called for and ideally a method of forecasting demand should also be devised in order that changes could be anticipated. Before either of these problems could be considered it was necessary to define how demand was to be measured in this particular case.

For some sizes of domestic tube it might have been possible to use a method analogous to that employed by Marketing Dept. for forecasting aggregate demand. Thus, by using linear regression techniques an equation based on investment in private and public sector housing could have been obtained. However, despite the nomenclature, certain sizes of domestic copper tube are not used greatly in housing construction and the rational basis for using these variables does not apply. A more serious disqualifier for this approach was that forecasts of investment in public and private housing (and hence any related forecast) were only available quarterly, while it was desired to monitor stocks, and if necessary, modify production on a weekly basis.

An apparently direct measure of demand was available by inspecting orders on hand. However, a well recognised practice in the trade was for customers to place inflated orders in the hope of obtaining better deliveries, and indeed to place the same order with several companies for similar reasons. This measure of demand was quite unacceptable therefore, particularly so if the problems of obtaining the information from the Sales Order Processing system (2.3.3 iii) were taken into account.

It was concluded that the only practical method of operationally defining demand was to measure warehouse despatches. These were recorded on a weekly basis and were a direct measure of the amount of each product sold. In addition 2 - 3 years of historical data were available enabling the use of more discriminating analytical techniques for determining the best predictors of demand. Despite these attractive practical considerations, warehouse despatches could become a very biased indicator of "true" demand under certain circumstances. Thus, in a sellers' market when demand exceeded supply for example, the latter would be governed by the amount of production possible. In such circumstances, demand measured by warehouse despatches would, in fact, be a function of production capability rather than a function of the market.

4.3 FORECASTING PROCEDURES

4.3.1 Historical Demand Data

Past records of demand data (as defined in 4.2) had been collected for the simulation exercise described in Appendix 1. These were now used to produce computer generated plots of demand for all individual products in the B.S. 659 Table 'A' and B.S. 3931 product groups for the years 1967 to 1969 inclusive. Records before 1967 were not considered relevant for analysis purposes, as both product groups had only been introduced in the two years preceding 1967. These plots were inspected to obtain a visual indication of the level and variability of demand for each product. In fact a third product group of less importance was that covered by B.S. 1386. However, of these three groups, B.S. 659 Table 'A' - subsequently referred to as Table 'A' - accounted for approximately 90% of sales by weight in the construction category. Furthermore, about 5 individual products accounted for most of the sales in the Table 'A' product group, as can be seen in Table (4.2). Since there were 48

Table 4.2 THE RELATIVE DEMAND FOR INDIVIDUAL PRODUCTS - BASED ON DATA PRESENTED IN FIG. 5.19

Product Size (mm)	Table X (%)	Table Y (%)	Table (%)
6	0.0	0.0	0.0
8	0.1	0.0	0.0
10	0.0	0.0	0.0
12	0.2	0.0	0.0
15	37.4	0.5	1.9
18	0.0	0.0	0.0
22	35.7	0.4	2.0
28	11.0	0.1	1.1
35	3.6	0.1	0.3
42	2.2	0.1	0.3
54	1.5	0.1	0.2
76	0.3	0.0	0.1
108	0.4	0.0	0.1
133	0.1	0.0	0.0
159	0.4	0.0	0.0

individual products within the 3 product groups, the existence of 5 products which in total accounted for most of the sales was an obvious focus for forecasting effort.

A typical plot of demand for one of these high volume products is shown in figure (4.1). Some increase in demand is evident in the final quarter of each year. In general, demand was found to be low in the first two quarters of the year and high in the last two quarters, the variation reputedly reflecting plumbing activity in the construction industry with respect to new housing.

Broadly speaking, demand patterns for individual products could be split into two categories. In the first of these there was some demand for each product each week, although the amount of demand varied considerably. Fig. (4.1) is typical of this category. In the second, demand was generally low and often several weeks could pass without any demand being registered. Fig. (4.2) shows a typical demand plot for products in the second category. In order to use currently available methods of time series analysis it would be necessary to accumulate such results over longer periods of time, say perhaps on a monthly basis, and in this way ensure that "missing values" in the series were eliminated. Unfortunately, this would have had the effect of extending the response time to changes. Additionally, it would have reduced the number of data points available.

After due consideration it was decided that products having regular weekly demand offered a more potentially profitable approach. It would be possible to forecast on a weekly basis and thus obtain the shortest response time of practical value given the production constraints. Furthermore, inspection of the records suggested that if a predictive model of demand for one product in this category were established, then the demand patterns were sufficiently similar for the same model to be used in predicting demand for the other products in

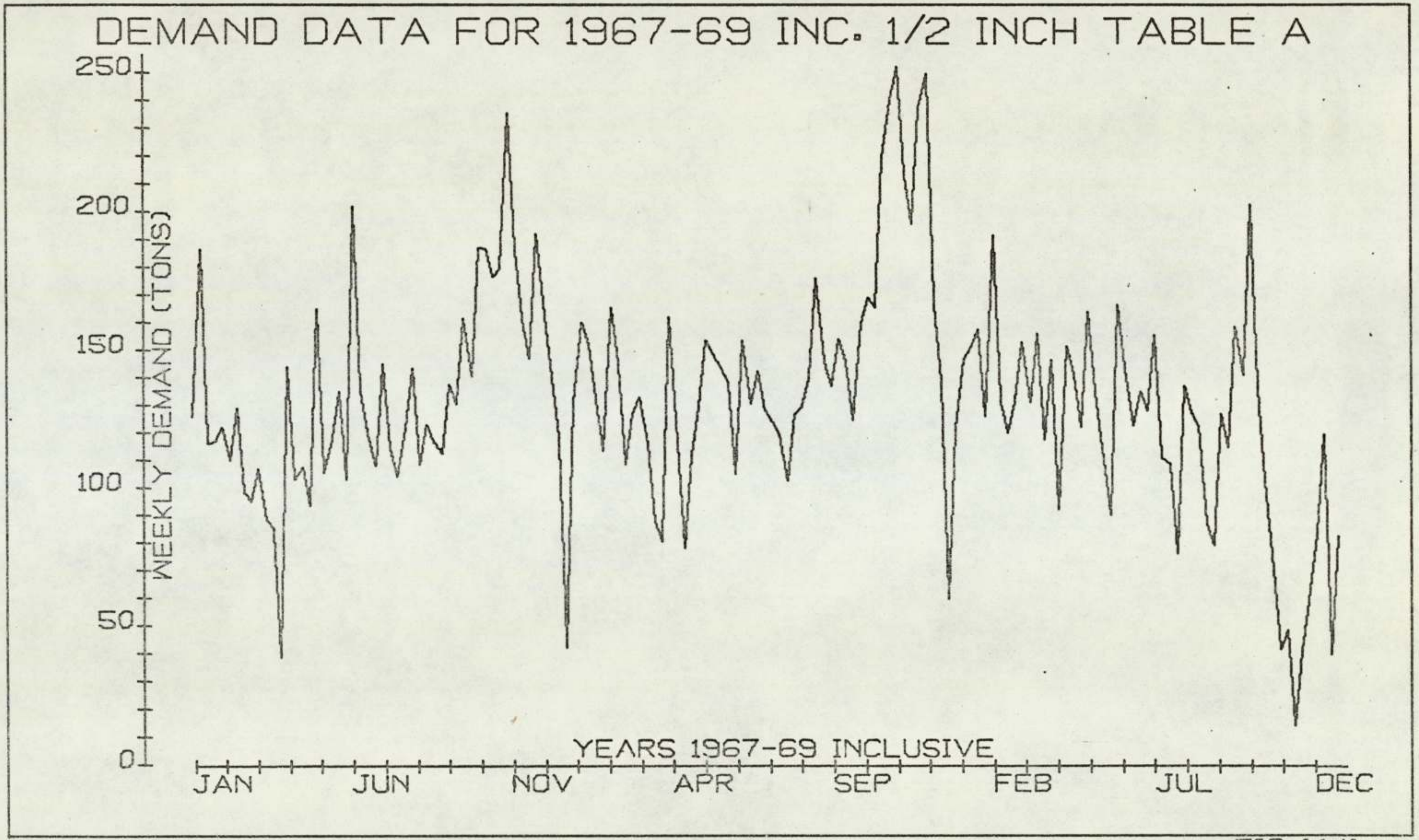

DEMAND DATA FOR 1967-69 INC. 1/2 INCH TABLE A
250
200
150
100
50
0
WEEKLY DEMAND (TONS)
YEARS 1967-69 INCLUSIVE
JAN
JUN
NOV
APR
SEP
FEB
JUL
DEC

FIG (4.1)

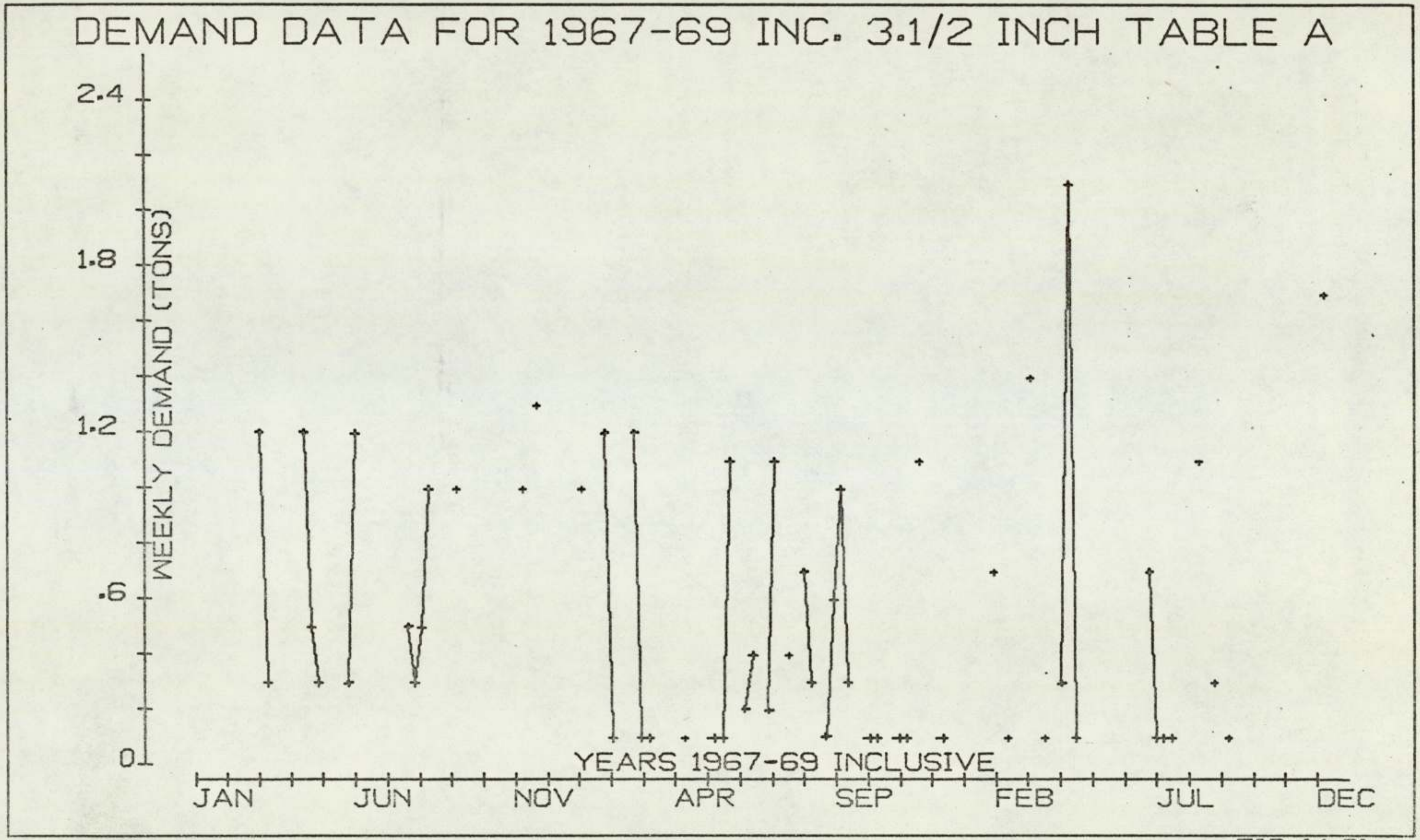

DEMAND DATA FOR 1967-69 INC. 3.1/2 INCH TABLE A
WEEKLY DEMAND (TONS)
2.4
1.8
1.2
.6
0
YEARS 1967-69 INCLUSIVE
JAN
JUN
NOV
APR
SEP
FEB
JUL
DEC

FIG (4.2)

the category, albeit with different weighting coefficients. If this were successful, then a relatively small amount of work involving forecasting demand for about 5 - 10 products would enable the greater part of the total demand to be forecast.

It was decided that ¾" Table 'A' was a typical representative of the first demand category described above, and attempts to predict demand concentrated on this product. Demand for ¾" Table 'A' in the years 1967 - 1969 inclusive is plotted in fig. (4.3). It may be compared for similarity with fig (4.1).

4.3.2 A Review of the Theory

This section briefly reviews the principles of analysis and prediction of time series. It is presented as an introduction to the account of work carried out in forecasting demand during the present research, which is described in 4.3.3 and 4.3.4.

The starting point of any attempt to forecast by mathematical modelling is an analysis of past data. Such data takes the form of a series of numbers representing the value of some variable of interest at regular intervals stretching from the present back through time. (In the present case the variable of interest was weekly demand). In the analysis of such a series it is usual to attempt to build a mathematical model which is capable of reproducing the observed series within acceptable limits of accuracy. The observed series may then be regarded as the product of a mathematical process. Most of the series observed in real life require models that take account of random variation in reproducing the observed series. Such models are known as stochastic processes. When a stochastic model is used for forecasting purposes, it is only possible to state the probability of the forecast falling within a given range. Because it takes account of random variation, a stochastic process will generate an infinite number of series known as realisations of the process. The observed series is regarded as one such realisation.

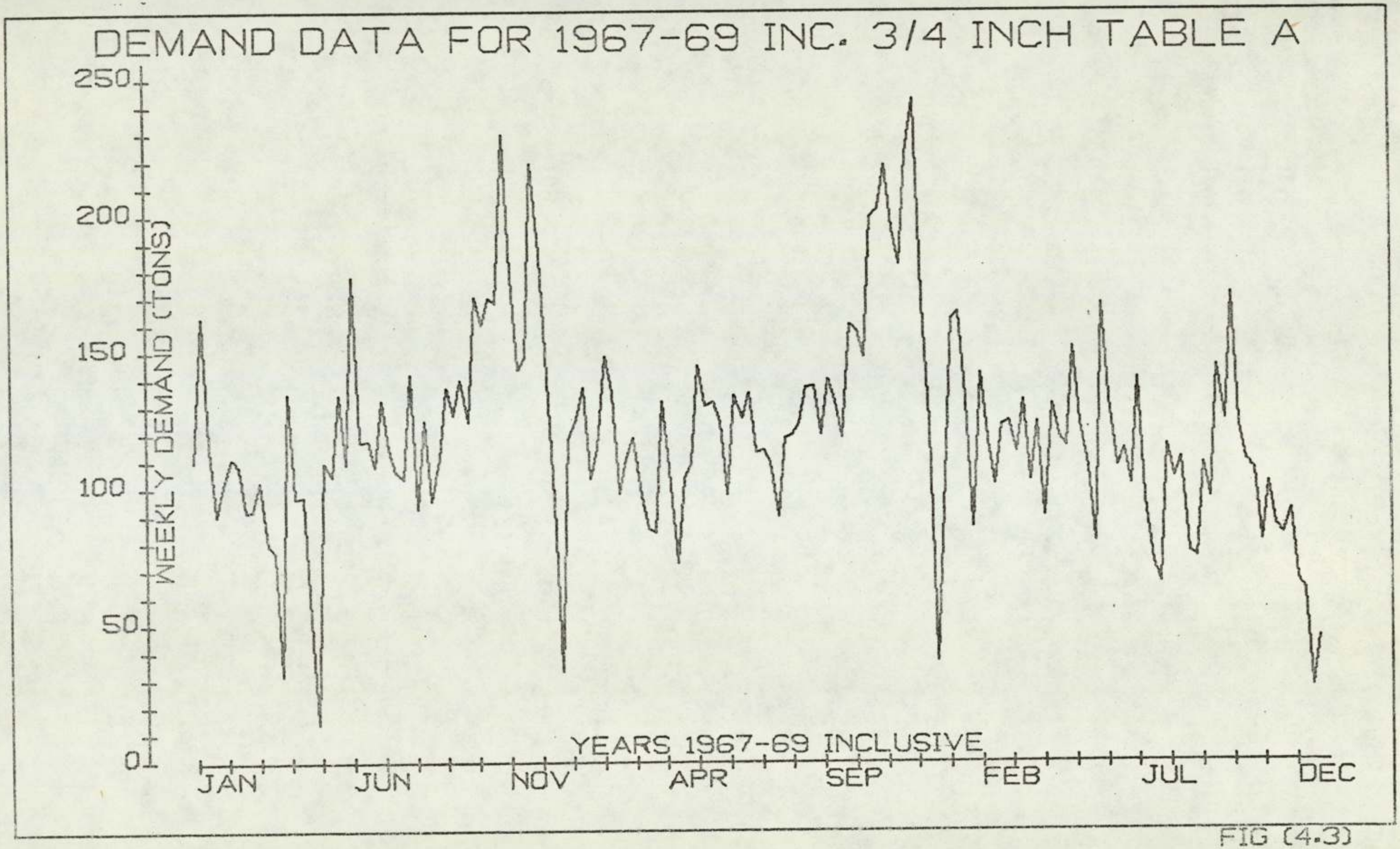

DEMAND DATA FOR 1967-69 INC. 3/4 INCH TABLE A
250
200
150
100
50
0
WEEKLY DEMAND (TONS)
YEARS 1967-69 INCLUSIVE
JAN
JUN
NOV
APR
SEP
FEB
JUL
DEC

FIG (4.3)

During the course of the analysis it is usual to produce statistics summarising information averaged over the period of the data. For example, the simple average may be computed for the period considered. This would be called a "time-averaged" statistic. Clearly the value of such statistics for prediction purposes is limited unless the series from which they are derived is typical, in a statistical sense, to any other realisation of the process producing the series. If other realisations of the process do differ in statistically significant respects then the summarising statistics of relevance must be derived at a given time (t) from the "ensemble" of several, or perhaps all possible, alternative realisations of the process, and not with respect to time. Thus it is possible to derive "ensemble - averaged" statistics. This is illustrated in fig. (4.4)

The above conclusion has great practical significance as it is unusual in real life situations to have available more than one realisation of a process, which means that only time averaged statistics may be derived. It also explains the theoretical interest shown in stationary processes in general and in particular, in ergodic processes which are a particular sub-class of the former category. A stationary process is invariant under translation in time, and thus realisations of the same process over consecutive periods are statistically equivalent. Furthermore the ergodic theorem states that if a process is ergodic, then time averaged properties and their corresponding ensemble averaged properties are equal.

In practice, many time series of interest are non-stationary, which precludes the use of time averaged statistics in lieu of their "ensemble" equivalents, given that only one realisation is available. To overcome this problem a series that is patently non-stationary is often regarded as being composed of two components. One of these is usually a deterministic component which may be evaluated to predict trend, and the

TIME AVERAGED AND ENSEMBLE AVERAGED STATISTICS

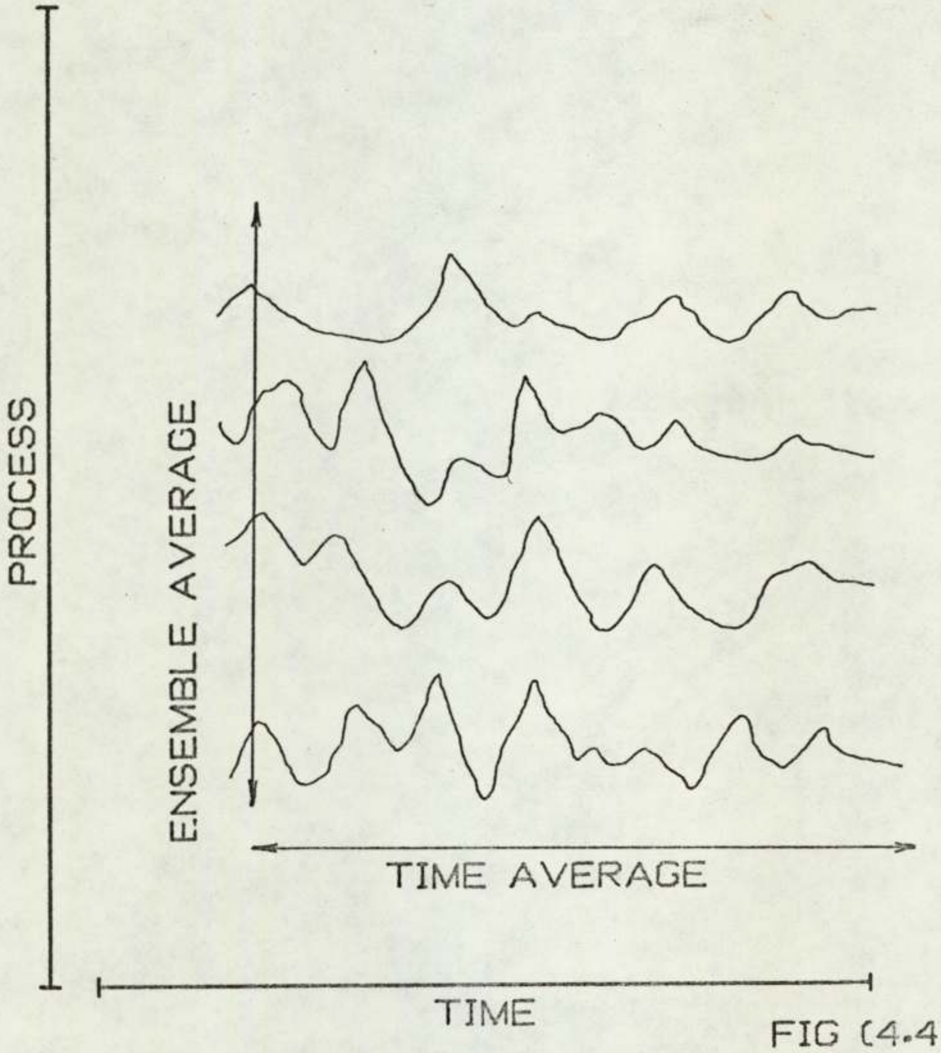


FIG (4.4)

second is a series of uncorrelated random residuals which may be stationary if the deterministic component is suitably chosen. The original process $x(t)$ may thus be represented as

$$x(t) = \xi(t) + \varepsilon(t) \qquad \text{Equ (4.3.1)}$$

where $\xi(t)$ = deterministic component

$\varepsilon(t)$ = uncorrelated, random residuals having zero mean

Although the deterministic component may take a variety of forms, interest has classically centred on polynomials of the form,

$$x(t) = a_o + a_1\, t + a_2 t^2 + \text{-----}\ a_p t^p \text{ ----- Equ (4.3.2)}$$

This is a particularly useful class of components to study since any function having finite derivatives may be expressed in the form of a polynomial by use of a Taylor expansion. It is possible of course to fit a given number of points perfectly by using a polynomial of sufficiently high degree. For practical purposes however it is usual to decide the degree of the polynomial in advance, after considering the origins of the series and the purpose to which the analysis is to be put.

Once the order (p) of the polynomial has been chosen, in order to obtain an optimal choice of coefficients $(a_o, a_1, a_2 \text{ ----- } a_p)$ in terms of least square error, it is necessary to solve (p+1) equations of the type,

$$\partial/\partial a_j \sum_{t=1}^{n} (x_t - a_o - a_1\, t - a_2 t^2 \text{ ---- } - a_p t^p)^2 = 0 \text{ ----- Equ (4.3.3)}$$

for j = 0, 1, 2 --- p

where n = p+1 is the number of points to be fitted.

The solution to the above for a_o is of the form,

$$\hat{a}_o = \sum_{t=}^{n} w_t . x_t$$ where the w_t are a series of weights

Thus at t=o the trend is optimally given by a_o which is found to be a weighted average of the x's included in the extent 'n' of the moving average so defined. The actual values of the w_t depend on the values of p and n used.

Often a local origin is defined based upon the mid-point of the n points. Thus if n is odd, let n = 2m + 1, when the local origin is found at m+1. The expression for a_o then becomes,

$$\hat{a}_o = \sum_{\tau=-m}^{m} w_\tau . x_\tau \qquad \text{Equ } (4.3.4)$$

For the simple case of p=o, $w_i = \frac{1}{n}$ for i = - m --- m. Thus the optimal estimator of trend in this case is a simple average of x_τ centred on the local origin x_o. A series of trend values can thus be calculated by successively advancing the local origin by one relative to the original series and re-computing the trend.

The above summarises what may be described as the classical approach to time series analysis in the time domain (ie excluding the methods of spectral analysis) as described, for example, by Kendall (1966a). There are of course a number of refinements. The variate-difference method of Tintner (1963) is able to give an indication of the order of the $\xi(t)$ component present, while the method of multiple simple moving averages enables the fitting of high order polynomials to be approximated.

However during the last decade there have been a number of important developments in time series analysis. One of the greatest disadvantages of time series analysis by the fitting of polynomials or by moving average of even modest extent was the computational effort required. With the advent of high speed electronic computers this problem was considerably reduced and it became possible for forecasting techniques to be widely utilised. However until recently direct access computer storage has been very expensive and its cost is still not inconsiderable, while the methods discussed above require that a relatively large number of items of past data (and various coefficients) be stored in order that forecasts can be computed. Although one of the largest applications for short term forecasting has been in industrial stock control where demand for many thousands of separate items is now regularly monitored by computer, this was not possible with the above methods of forecasting. Thus there has been great interest in exponentially weighted forecasting methods of the kind proposed by Brown (1962), Holt (1957) and Winters (1960).

Typically these forecasts take the form

$$\hat{z}_{t+1,t} = \alpha (z_t - \hat{z}_{t,t-1}) + \hat{z}_{t,t-1} \qquad \text{Equ (4.3.5)}$$

or alternatively,

$$\hat{z}_{t+1,t} = \alpha . z_t + (1 - \alpha)\hat{z}_{t,t-1} \qquad \text{Equ (4.3.6)}$$

where

$\hat{z}_{t,t-1}$ = forecast of z made at period (t-1) for period (t)

z_t = actual value of z at period (t)

α = coefficient, usually in the range 0.05 - 0.2.

Thus in the above scheme to make a forecast it is only necessary to know the forecast computed for the last period, the value of 'α' to be used and the actual value of z_t realised for the current period (t). When such a procedure is carried out by computer, both the storage and the amount of computation required are minimal and it is schemes like these that are utilised in stock control systems.

Although simple exponentially weighted forecasts are easy to use and give most importance to recent events, which is usually a very desirable characteristic in practical forecasting schemes, the method does evade the general problem of non-stationary processes. Therefore, while Brown (1962) has shown how trends can be accommodated by multiple smoothing, when additional operations are required the basic simplicity of the method is lost and the approach also loses some of its attractiveness.

In an attempt to encompass non-stationary processes, Box-Jenkins (1962) developed a class of models often known collectively as the Box-Jenkins γ- model. The general model may be written as follows,

$$\hat{z}_{t+1,t} = (\gamma_{-\ell}\nabla^{\ell} + ---- \gamma_{-2}\nabla^{2} + \gamma_{-1}\nabla^{1} + \gamma_{0} + \gamma_{1}S^{1} + \gamma_{2}S^{2} + --- + \gamma_{m}S^{m})\, e_t + \hat{z}_{t,t-1} \quad \text{---- Equ (4.3.7)}$$

where $\hat{z}_{t+1,t}$, $\hat{z}_{t,t-1}$ are as defined above,

γ_i, i=1, ..., o, ... m is a system of weights

∇ = backward difference operator

(defined as $\nabla x_t = x_t - x_{t-1}$)

operating on e_t.

S = summation operator, defined as

$$S x_t = \sum_{j=o}^{\infty} x_{t-j} \,;$$

$$S^2 x_t = \sum_{j=o}^{\infty} \sum_{k=o}^{\infty} x_{t-j-k}$$

and in general $S^p x_t$ is the p^{th} multiple sum of the past x's

e_t = error in forecast = $z_t - \hat{z}_{t,t-1}$

The significance of the γ-model is that it enables a predictor of a process to be found without recourse to defining a deterministic and a random component.

It is possible of course to investigate how the model would perform if the process were composed of various types of deterministic components combined with a random component. In fact the authors show that a predictor based on the above model would be optimal for any stochastic variable 'z' - which thus may be non-stationary - whose (m+1) difference may be represented by a moving average of order (l+m+1). In practical forecasting it is usual to use only the terms $\gamma_{-1}\nabla^{-1}e_t$, $\gamma_o e_t$ & $\gamma_1 S^1 e_t$ known as the first-difference, proportional and cumulative terms, by analogy with control theory terminology. (The authors exemplify much of their work in process control applications).

More recently however Box & Jenkins (1968) proposed a class of models frequently referred to as (p,d,q) models after the parameters which define them. These are more formally described as Autoregressive Integrated Moving Average Models in a subsequent work by Box and Jenkins (1970). The essence of the method is that the original series is differenced until stationarity is obtained and the resulting series is then fitted by an autoregressive model, a moving average model or a combined autoregressive moving average model.

Using the notation of the Box-Jenkins (1968) paper, an autoregressive model might be as follows,

$$\dot{w}_t = \not{0}_1 \dot{w}_{t-1} + \not{0}_2 \dot{w}_{t-2} + a_t \text{ (order 2) --- Equ (4.3.8)}$$

where $\not{0}_1$, $\not{0}_2$ are coefficients,

$$\dot{w}_t = w_t - \mu$$

and w_t, w_{t-1}, w_{t-2} etc are the values of a stationary series having mean 'μ', at intervals t, t-1, t-2 etc. The series a_t , a_{t-1}, a_{t-2} etc are uncorrelated normally distributed deviates.

Alternatively, a moving average model to fit the series w_t might be as follows,

$$\dot{w}_t = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \theta_3 a_{t-3} \text{ (order 3) ---- Equ (4.3.9)}$$

where θ_1, θ_2 & θ_3 are coefficients.

A mixed autoregressive-moving average model of order (p,q) may be written,

$$\dot{w}_t - \not{0}_1 \dot{w}_{t-1} - \dots - \not{0}_p \dot{w}_{t-p} = a_t - \theta_1 a_{t-1} \dots \theta_q a_{t-q} \dots \text{ Equ (4.3.1}$$

The above equation may be condensed by defining a backward shift operator B such that,

$$B\, w_t = w_{t-1}$$

Using the backward shift operator equation (4.3.10) may be written,

$$\Phi_p (B) \dot{\omega}_t = \Theta_q (B)\, a_t$$

where $\Phi_p (B) = 1 - \not{0}_1 B - \not{0}_2 B^2 - \dots \not{0}_p B^p$

and $\Theta_q (B) = 1 - \theta_1 B - \theta_2 B^2 - \dots \theta_q B^q$

Thus Φ_p (B) is called the autoregressive operator and Θ_q (B) the moving average operator. Box-Jenkins describe the conditions that limit the range of values ϕ_i & θ_j (i=1, ... p, j=1, ... q) may take in their paper (1968). If the model is written in terms of the w_t rather than the deviations from the mean μ we have,

$$\phi_p (B) w_t = \theta_o + \Theta_q (B) a_t \qquad \text{Equ } (4.3.11)$$

$$\text{where } \theta_o = (1 - \phi_1 - \phi_2 - \ldots . \phi_p). \mu$$

Equation (4.3.11) may be expanded to accommodate a non-stationary series z_t that is stationary in its d^th^ difference. Since

$$\nabla z_t = (1-B) z_t,$$

$$\text{if } \nabla^d z_t = (1-B)^d z_t = w_t,$$

the model for z_t becomes,

$$\Phi p (B) (1-B)^d z_t = \theta_o + \Theta_q (B) a_t \qquad \text{Equ } (4.3.12)$$

However equation (4.3.12) supplies a very large number of models and in practical applications it is necessary to apply some criterion whereby the potential variety can be reduced. This the authors do by using the "principle of parsimony" ennunciated by Tukey. That is, the simplest model capable of representing the chosen series is used. The use of a model that is more complicated than necessary to adequately represent a series is known as "overfitting" the series. It is claimed that many of the series encountered in real life may be represented by models where p,d & q each do not exceed the value 2.

The method described by Box-Jenkins in their (1968) paper involves calculating the correlogram (autocorrelation structure) of the series to be fitted. The authors present tables describing the behaviour of the theoretical autocorrelation function of the d^{th} difference of series for selected simple (p,d,q) models. The derivation of the data described therein is given in Box-Jenkins (1970).

Once a model has been selected by inspection of the sample autocorrelation function of the series, particular weights

($\phi_1, \phi_2 \ldots \phi_p$ and $\theta_1, \theta_2 \ldots . \theta_q$) are chosen to minimise the sum of squared errors over the period to which the model is being fitted.

Where seasonal variation exists, a multiplicative model may be used. Thus if seasonality exists for items separated by 's' intervals, differencing across this interval ($\nabla_s Z_t = Z_t - Z_{t-s}$) produces a new series which may be fitted using the model described in equation (4.3.12). After fitting an appropriate model in this way, a series of residuals (e_t) would be generated which could then be fitted by a second model. Thus operating on the original series would produce "seasonal free" residuals (e_t) and operating subsequently on the (e_t) would produce an uncorrelated series of error residuals (a_t). The parameters obtained at each stage could then be combined to give a multiplicative model for use as a predictor with untreated members of the original series.

The autocorrelation function and the power spectrum obtained by the methods of spectral analysis are mathematically equivalent (one is a Fourier transform of the other). Thus use of the autocorrelation function to identify the essential features of a discrete series may be regarded as analogous to use of the power spectrum to identify the essential frequencies in a continuous series. It is the potential ability of the Box-Jenkins (p,d,q) model approach to identify the autocovariance structure of the series under examination and then to fit a mixture of autoregressive/moving average models having a similar autocovariance structure that makes it so attractive.

4.3.3 Use of Box-Jenkins (p,d,q) models

It was decided to use this method of attempting to predict weekly demand because of its ability to extract any structure inherent in the past data, since such structure could be used for prediction purposes.

The originators of the method recommend that models be fitted on the basis of "a minimum of 50 and preferably more than 100 data values". At the time of the exercise in mid-1970, records of weekly demand were available from 1st January, 1967, onwards. It was decided that the demand records for 1967-69 inclusive would be used for fitting a model, and that demand in 1970 would be used to test the model selected. For the reasons given in 4.3.1 attention was confined to the product $\frac{3}{4}$" Table 'A'.

In the original data a problem arises because of spuriously low demand at each year end. This occurs because demand is measured as warehouse despatches (4.2) and customers generally do not want deliveries over the Xmas/New Year period. To minimise the effect of this, the demand data were homogenised by recording demand in the last two weeks of the year as the average demand for those weeks.

Following the method of Box-Jenkins outlined above, a correlogram for the homogenised data series was calculated using the program AUTOCOR (Appendix P.4). This is shown in fig. (4.5). Ignoring for the moment autocorrelations at lags 4 & 5, the correlogram corresponds fairly reasonably to a (1,0,0) model with $\phi_1 = 0.6$. This may be deduced from the fact that the theoretical autocorrelation function for a (1,0,0) model is given by the relationship,

$\rho_k = \phi_1^k$ where ρ_k = theoretical autocorrelation function for lags 1,2 --- k

ϕ_1 = weight in the range $- 1 < \phi_1 < 1$

Since the sample autocorrelation ρ_k for k=1 is 0.6, equating ρ_k & r_k leads to the conclusion $\phi_1 = 0.6$. This conclusion is supported by fitting the homogenised series with various values of ϕ_1 in the (1,0,0) model. A plot of sum of squared errors versus ϕ_1 shows a minimum of $\phi_1 = 0.6$, as can be seen from fig. (4.6).

If it is assumed that the errors are normally distributed, values of the weights ϕ_i & θ_j (i=1, 2, ... p, j=1, 2 .... q) will be approximately equal to the maximum likelihood estimators of ϕ_i & θ_j.

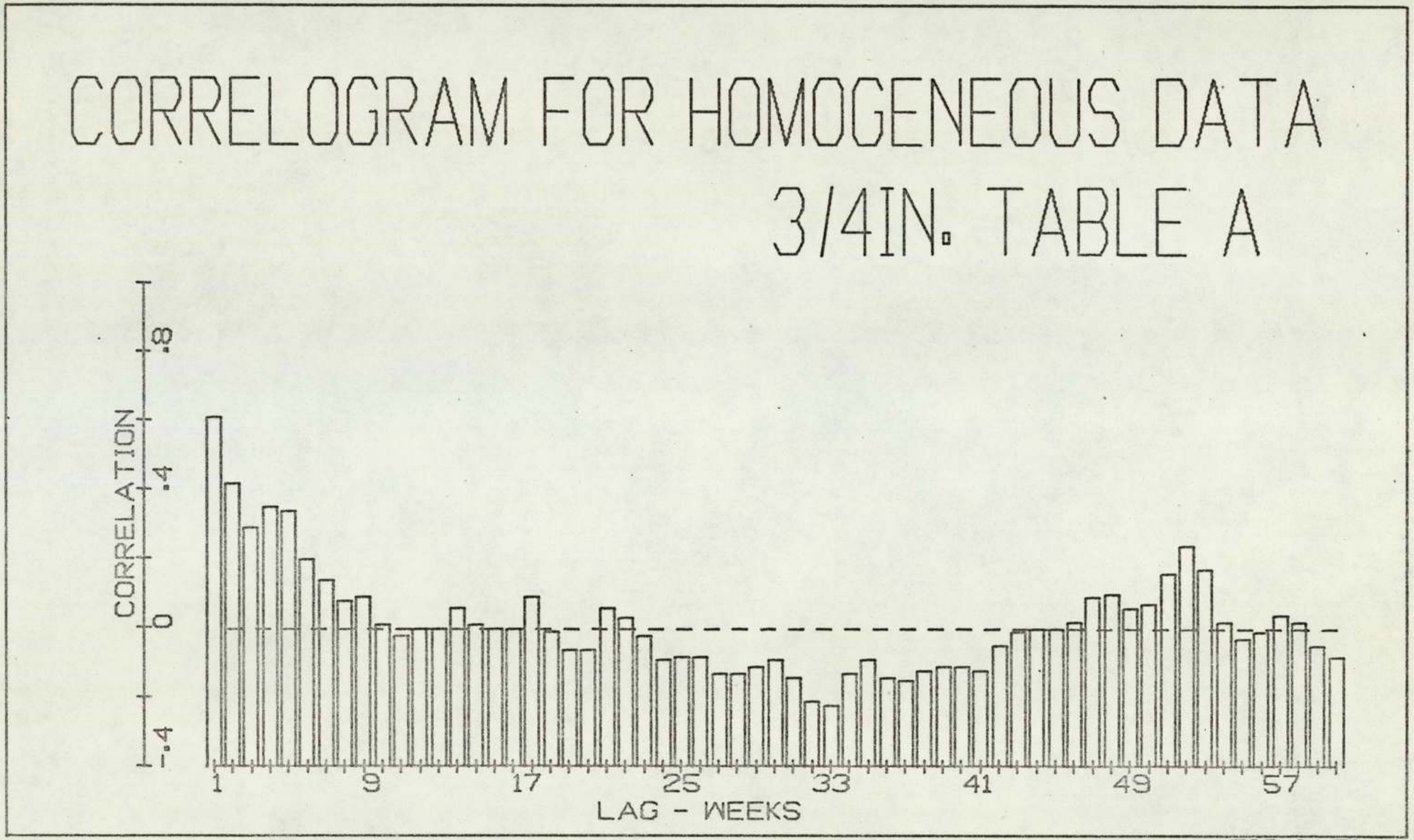

CORRELOGRAM FOR HOMOGENEOUS DATA 3/4IN. TABLE A

FIG (4.5)

SUM OF SQUARED ERRORS AFTER FITTING (1,0,0) MODEL TO HOMOGENISED DATA - 3/4 IN. TABLE A.

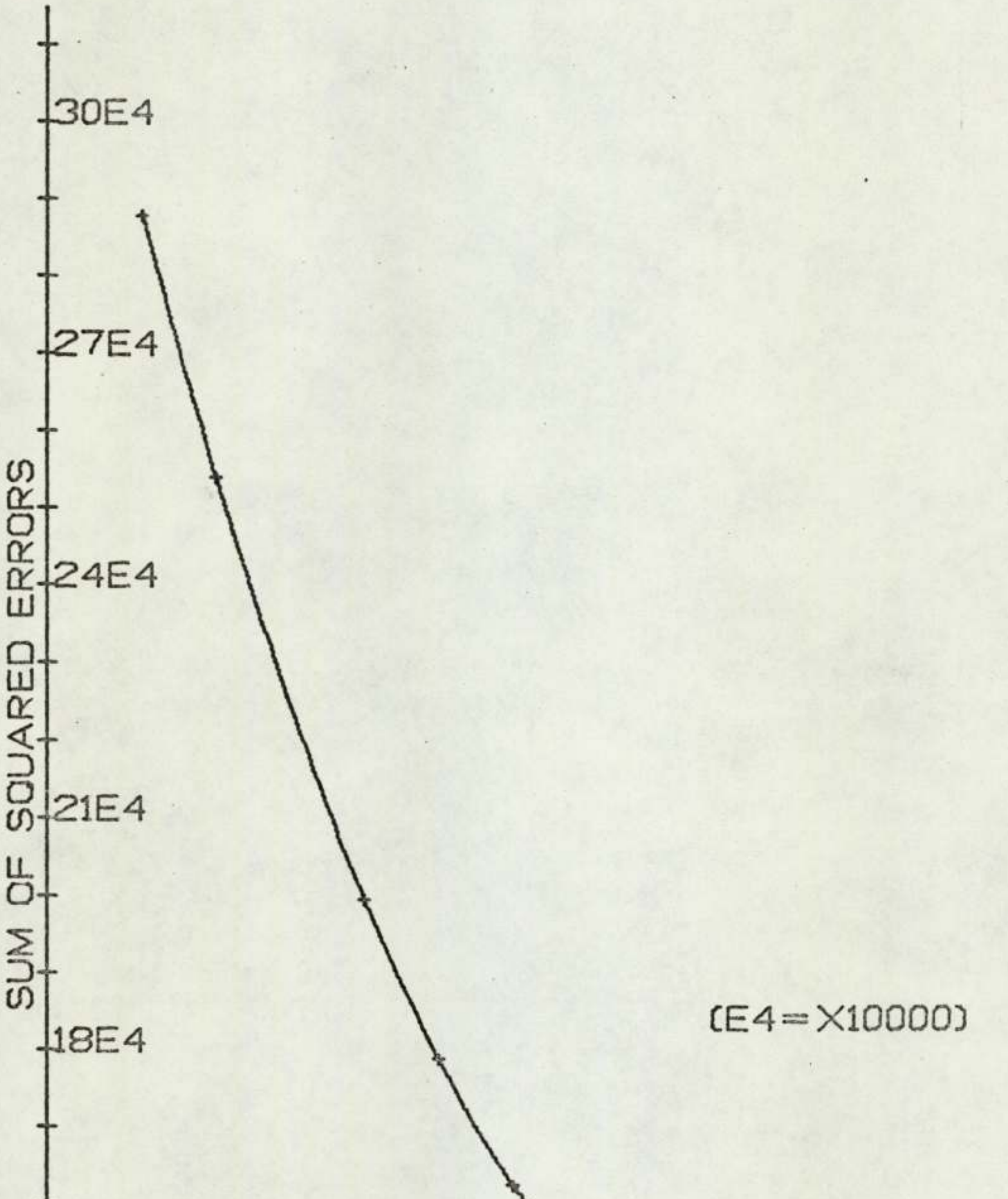
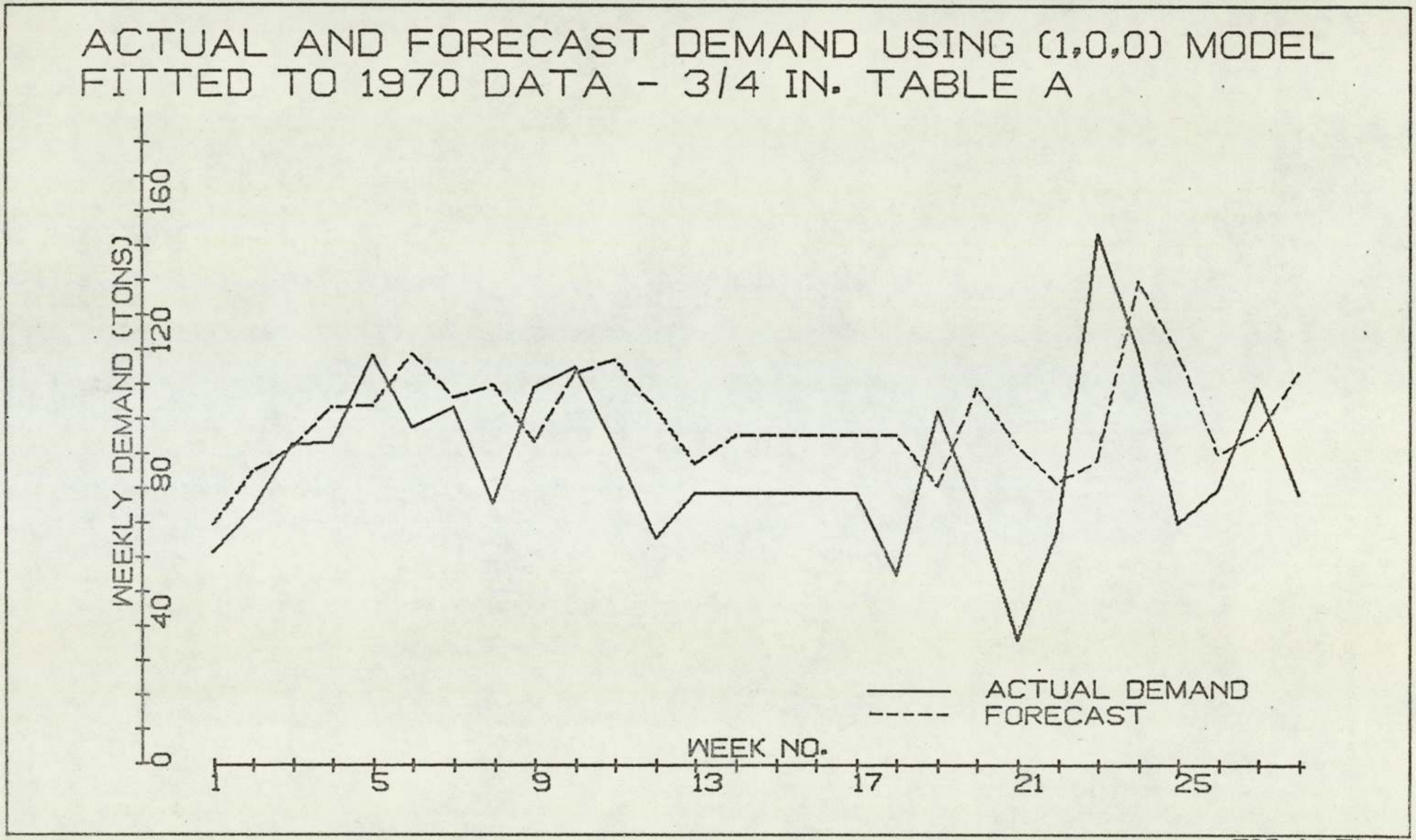


FIG (4.6)



ACTUAL AND FORECAST DEMAND USING (1,0,0) MODEL FITTED TO 1970 DATA - 3/4 IN. TABLE A

FIG (4.7)

Box-Jenkins (1970) develop an expression giving a confidence contour for the maximum likelihood estimators of $\emptyset$ & θ which is as follows,

$$S_{1-\varepsilon}(\emptyset, \theta) = S(\hat{\emptyset}, \hat{\theta}) \left[1 + \frac{\chi^2_{\varepsilon} \cdot (p+q)}{n-p-q} \right] \qquad \text{Equ (4.3.13)}$$

where χ^2_{ε} (p+q) is the significance point exceeded by a proportion ε of the χ^2 distribution having (p+q) degrees of freedom;

and S $(\hat{\emptyset}, \hat{\theta})$ is the sum of squared errors corresponding to the best estimators of $\emptyset$ & θ .

Thus in the case of the (1,0,0) model above, $\emptyset_1 = 0.6$ and $S(\hat{\emptyset}, \hat{\theta}) =$ 139,367. Taking a 95% confidence level ($\varepsilon = 0.05$) with n = 157, p=1 & q=o, equation (4.3.13) gives,

$$S_{1-\varepsilon}(\emptyset, \theta) = 139,367. \left\{ 1 + \frac{3.84}{156} \right\}$$

$$= 142,799$$

This result defines a 95% confidence range of 0.75 - 0.50 for $\emptyset_1$.

However the model obtained was not regarded as being particularly satisfactory in that the minimum sum of squared errors taken over the three year period had a root mean square value of approximately 30 tons per week. Assuming a similar performance over future weeks when the model was used as a predictor, the forecast would be in error on the average by about 25% of the mean demand (120 tons per week).

In fitting the (1,0,0) model described above, no account was taken of the autocorrelations occurring at lags 4 & 5, which deviated markedly from the theoretical autocorrelation function for this model. Since the sampling distributions of autocorrelations can have large variances and can themselves be highly correlated (Kendall, 1945), this seemed to be justified. However in view of the disappointing performance of the (1,0,0) model, these deviations from the theoretical were investigated further. Although no actual cause was known for the relatively high autocorrelations at lags 4 & 5, it seemed quite reasonable to hypothesise that they might correspond to a real monthly relationship. Such cycles are very common in commerce. It was decided therefore to attempt to fit a seasonal model of the type already described, based on a period of 4 & 5 weeks respectively. In the event, neither of these models approached the performance of the autoregressive model identified initially, and it was decided that the results did not support the hypothesis that the autocorrelations were evidence of a real monthly relationship.

Because of the relative ease of fitting models to data using an electronic computer a number of other models were examined using program PDQMODEL (Appendix P4) and the results are presented in Table (4.3). However the (1,0,0) model with $\phi_1 = 0.6$ remained the best choice on a least squares basis. (Fig 4.7)

4.3.4 Use of Adaptive Smoothing

The relatively large residuals left after fitting an optimal (1,0,0) model suggested that an alternative model should be fitted to the data in the hope of obtaining a better fit. It may easily be shown that a (0,1,1) model is equivalent to a simple exponentially weighted moving average model. Thus the (0,1,1) model is written,

$\nabla z_t = \theta_o + (1 - \theta_1 B)\, a_t$ (using previous notation) Equ (4.3.14)

If the process is stationary θ_o may be omitted yielding at time (t+1),

$z_{t+1} = z_{t+1-1} + a_{t+l} - \theta_1 a_{t+1-1}$ Equ (4.3.15)

Taking expectations at time (t) and assuming the model is optimal,

$E(\hat{z}_{t+1}) = z_{t+1} \quad 1 \leq 0$

$E(a_{t+1}) = 0 \quad 1 \geqslant 1$

$\hat{z}_{t+1} = \hat{z}_{t+1-1} \quad 1 \geqslant 2$ Equ (4.3.16)

$\hat{z}_{t+1} = \hat{z}_t - \theta_1 a_t \quad 1 = 1$ Equ (4.3.17)

But $z_t = \hat{z}_{t,t-1} + a_t$, Equ (4.3.18) therefore substituting for a_t in equation (4.3.17)

$$\hat{z}_{t+1,t} = z_t - \theta_1 (z_t - \hat{z}_{t,t-1})$$

$$= (1-\theta_1) z_t + \theta_1 \hat{z}_{t,t-1} \qquad \text{Equ (4.3.19)}$$

substituting $\alpha = 1-\theta_1$ in equation (4.3.19) gives

$$\hat{z}_{t+1,t} = \alpha . z_t + (1 - \alpha) \hat{z}_{t,t-1} \qquad \text{Equ (4.3.20)}$$

It will be seen that equation (4.3.20) is identical with equation (4.3.6), the expression given for simple exponential smoothing.

It can be seen by inspection of Table (4.3) that a (0,1,1) model having weight $\theta_1 = 0.5$ gives a minimum sum of squared errors greater than that obtained with the first order regressive model described in section 4.3.3. Thus a simple exponentially smoothed model is inferior to the (1,0,0) model previously fitted.

However the original data series is not orderly, as is shown by its mean value of 120 tons per week and standard deviation of 40 tons. It may be deduced that no significant linear trends exist in the series by the fact that the data are best fitted by a zero difference (d=o) Box-Jenkins (p,d,q) model, and thus the relatively high standard deviation about the overall mean may be assumed to be due to random variation. However it is unlikely that the overall mean accurately reflects the value of the process mean over the full period considered. Instead the process mean may have drifted during this period. If this hypothesis were correct, then a simple exponentially weighted moving average would not be able to respond readily to such changes in the mean.

A better model would be a system proposed by Trigg & Leach (1967) and later modified by Shone (1967). The basis of Trigg's method was an index of control called a tracking signal. The tracking signal is defined as follows (Trigg, 1964).

$$-1 \leq \left(\frac{\text{smoothed forecast error}}{\text{mean absolute deviation}} \right) \leq 1$$

Table (4.3) SUMMARY OF RESULTS OBTAINED FROM FITTING VARIOUS (p,d,q) MODELS TO ¾" TABLE "A" DEMAND DATA. (1967-69)

(p,d,q)	Period of seasonality (wks) ∇_{seas}	Optimum weights $(\phi_1 \ldots, \theta_1 \ldots)$	Minimum sum of squared errors (residuals)	Mean of residuals	Variance of residuals
		SIMPLE MODELS			
Raw data	0			120.6(1)	1,455(1)
Homogenised data	0			120.6(1)	1,431(1)
(0,1,0) homogenised data	0			- 0.5	1,086
Homogenised data	5			- 1.9	1,781
(1,0,0)	0	0.6	139,367	- 0.1	888
(1,0,0)	5	0.5	205,117	- 0.6	1,349
(1,0,0)	52	0.6	127,489	- 2.5	1,208
(0,0,1)	0	- 0.5	165,676	0.3	1,055
(1,1,0)	0	- 0.3	157,728	**	1,011(2)
(0,1,1)	0	0.5	148,096	**	949(2)
		MULTIPLICATIVE MODELS			
(1,0,0) (0,0,1)	52 13	0.3	171,941	**	1,869(2)
(1,0,0) (0,0,1)	52 4	0.3	192,400	**	2,091(2)

Notes:-

(1) Refers to original series, not residuals

(2) Estimates of variance based on assumption of zero mean residuals

** - not computed.

N.B. All data used in fitting models have been homogenised - see section 4.3.3.

In calculating the tracking signal it is usual to exponentially smooth both the actual and absolute deviations from the observed value using a smoothing coefficient in the range 0.1 - 0.2. When the forecasts are accurate, positive and negative deviations tend to cancel and the tracking signal varies about zero. If a systematic error in the forecast occurs, the tracking signal approaches ± 1 depending on the sign of the error. Brown (1962) has shown a relationship between the mean absolute deviation and the standard deviation of forecast errors which enables control limits to be constructed for the tracking signal in a manner analogous to the construction of shewhart process control charts.

However the following facts may be noted,

a. the absolute value of Trigg's tracking signal varies between zero and unity

b. when large systematic deviations occur between forecast and observed values, the tracking signal approaches its maximum value and vice versa.

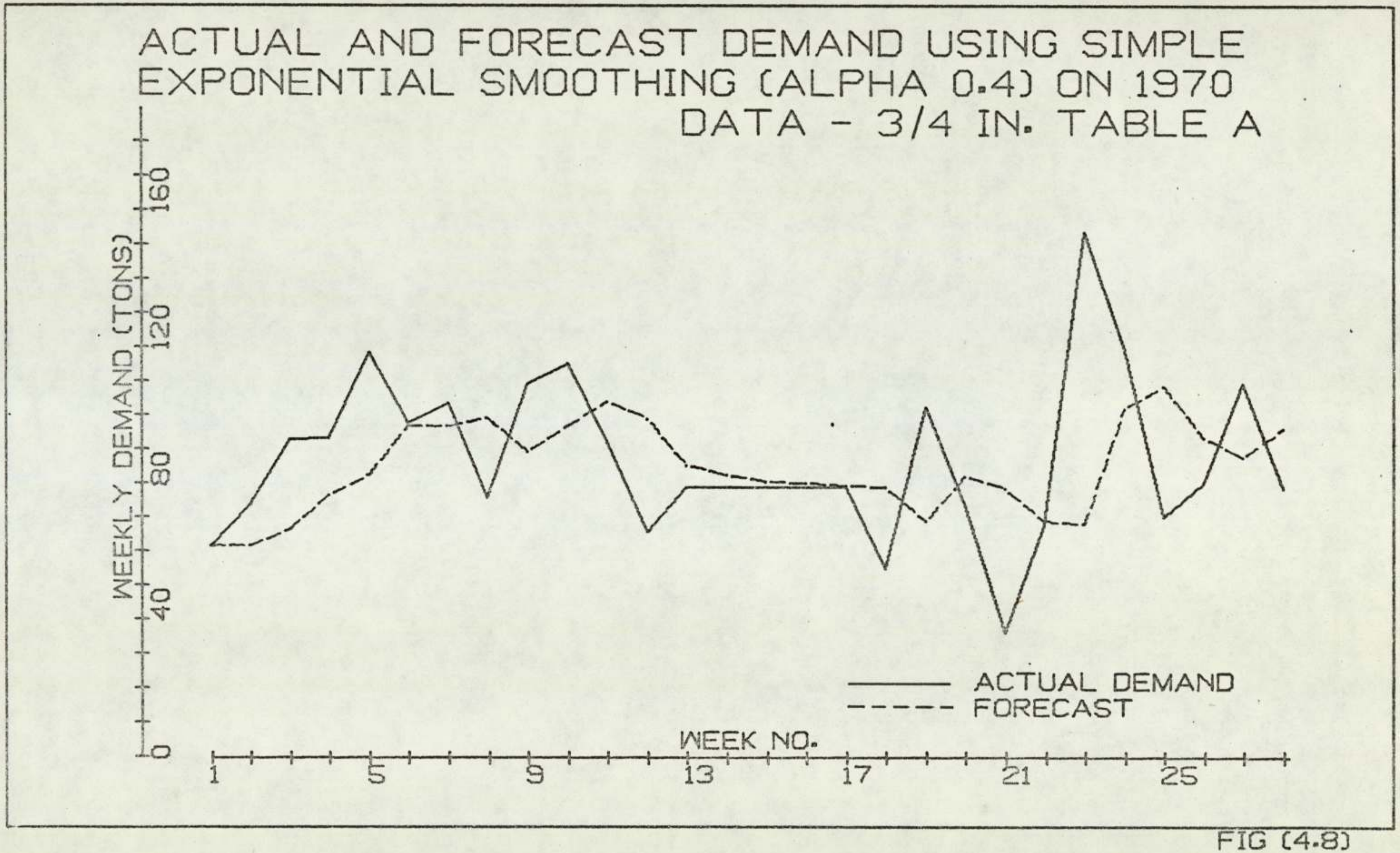
On the other hand, when simple exponentially smoothing with a low value of smoothing coefficient is applied to a process having a stable mean, low errors are observed and the forecasting system is relatively unaffected by occasional large deviations from the mean (spikes). Conversely, when the process mean is unstable and shifts from time to time, the exponentially smoothed forecast can most rapidly adjust to the new level if the smoothing coefficient has a large value. Unfortunately, a large smoothing coefficient makes the forecast very sensitive to occasional 'spikes'.

Ideally, therefore, an exponential smoothing system used for forecasting should have a value of smoothing coefficient which is low when conditions are stable but which would automatically increase in response to persistent changes in process mean. It should subsequently subside when conditions stabilise about the new mean. Obviously this situation can be achieved by using the absolute value of Trigg's tracking signal as the smoothing coefficient (α). The method is an example of adaptive smoothing.

At first sight it would seem reasonable to use the latest estimate of the tracking signal when forecasting for periods t+l (l >0). However Schone (1967) has demonstrated that better results may be obtained in certain circumstances if the tracking signal calculated at (t-1) is used in adaptive smoothing. In particular, the latter system is less affected by spikes. Ordinarily a spike occuring at time (t) would increase the value of the tracking signal calculated at (t) which would then be used to produce the spike affected forecast. Not only would the forecast be affected by the magnitude of the spike, but the increased value of the smoothing coefficient would give greater weight to this effect. Using Schone's approach, although the effect of the magnitude of the spike could not be discounted, this would not be magnified by the smoothing coefficient which would not have been affected at period (t-1). It would of course be affected when calculated for period (t), but if the spike was not distributed over more than one period, the transitory increase of the value of the smoothing coefficient would have relatively little effect on subsequent forecasts.

Adaptive smoothing using the modification described above was applied to the data series using program TRIG (Appendix P.5) which gives details of the initialisation procedures used. It was found that the sum of squared errors obtained was 188,803.

The results of fitting an exponential smoothing model (α=0.4) to the 1970 data available are shown in fig. (4.8) and in fig. (4.9) the adaptive smoothing model described above is fitted to the same data. Although the results shown in Table (4.3) indicate that a simple exponentially smoothed moving average model with coefficient (α=0.5)



ACTUAL AND FORECAST DEMAND USING SIMPLE EXPONENTIAL SMOOTHING (ALPHA 0.4) ON 1970 DATA - 3/4 IN. TABLE A

FIG (4.8)

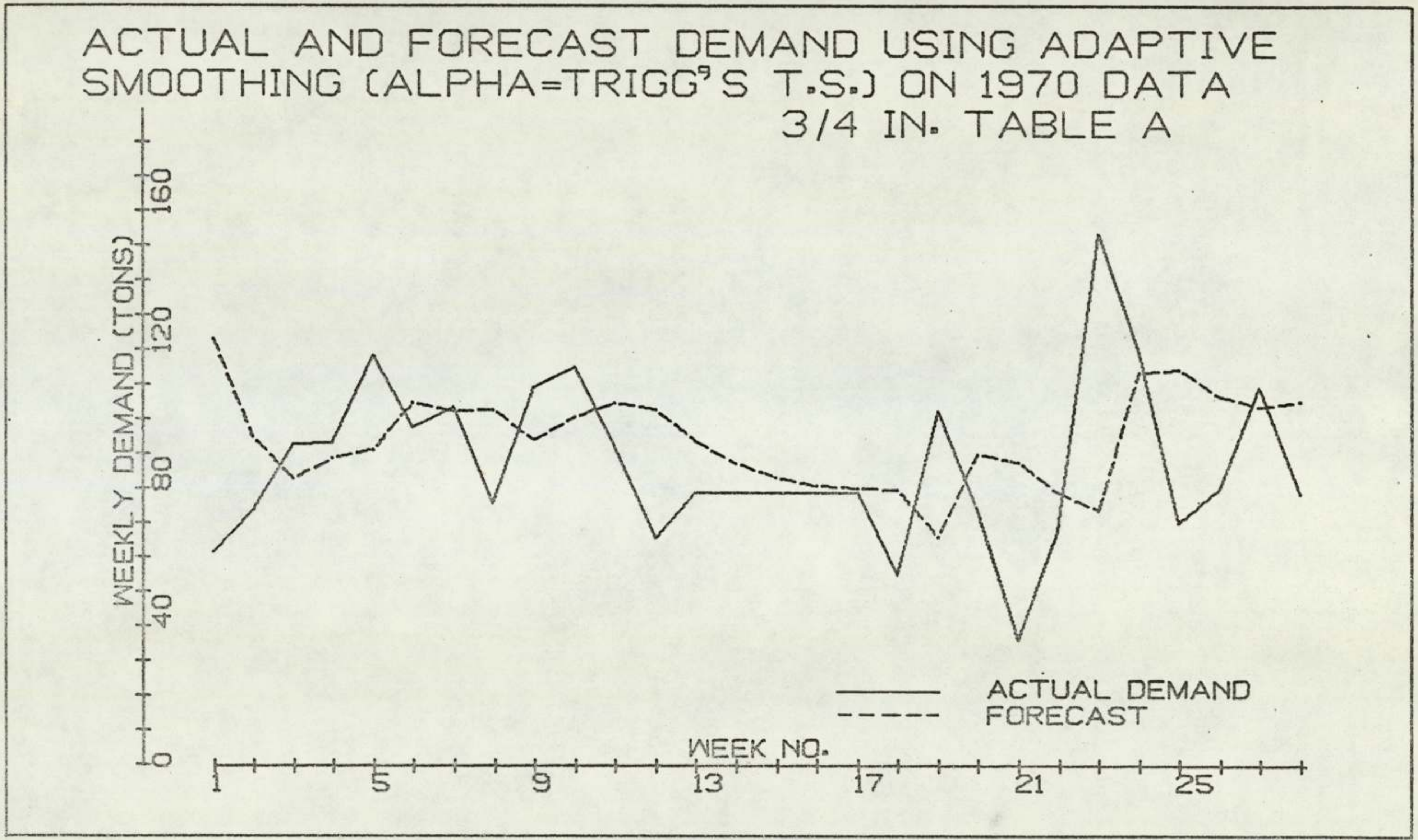

ACTUAL AND FORECAST DEMAND USING ADAPTIVE SMOOTHING (ALPHA=TRIGG'S T.S.) ON 1970 DATA
3/4 IN. TABLE A
WEEKLY DEMAND (TONS)
0
40
80
120
160
WEEK NO.
1
5
9
13
17
21
25
ACTUAL DEMAND
FORECAST

FIG (4.9)

would be optimal, fitting a model of the form given in equation (4.3.6) gave a least squares result at (α =0.4). Although these particular differences arise purely out of the methods of computation, fig. (4.10) shows the 95% confidence limits include the equivalent of α=0.4 (ie θ_1 =0.6) for the (0,1,1) model. In practice there is little to choose between the two values of smoothing coefficient.

4.3.5 Discussion

It is felt that the necessity of defining demand as warehouse despatches introduced its own significant contribution to the observable noise in the demand series. This was certainly true in the last quarter of 1969 where fig (4.3) shows that the high seasonal demand recorded in the previous two years is absent. This was due to a labour dispute occuring at the factory where most of the starting stock for Table 'X' production was made, during November and December of that year. On occasion the records of warehouse despatch almost certainly were not an accurate record of weekly demand. Thus the data shown, for example, in fig. (4.7) shows a constant demand for weeks 13-17 inclusive. The source documents suggest that is the result of weekly despatches not being individually recorded and subsequently the total demand recorded in that period being shared evenly over the preceding weeks. If the study reported here were to be extended then an investigation of the system of recording warehouse despatches would be advocated to establish (and subsequently eliminate) the various ways the recording system could introduce noise.

In fitting forecasting models to time series it is usual not to use all the data available for fitting the model. The data not used in fitting the model are subsequently used to test it. This latter procedure gives an indication of the robustness of the model to new data. The results of fitting the three models considered in 4.3.4 to data representing demand in the first 28 weeks in 1970 are plotted in

SUM OF SQUARED ERRORS AFTER FITTING (0,1,1) MODEL TO HOMOGENISED DATA - 3/4 IN. TABLE A.

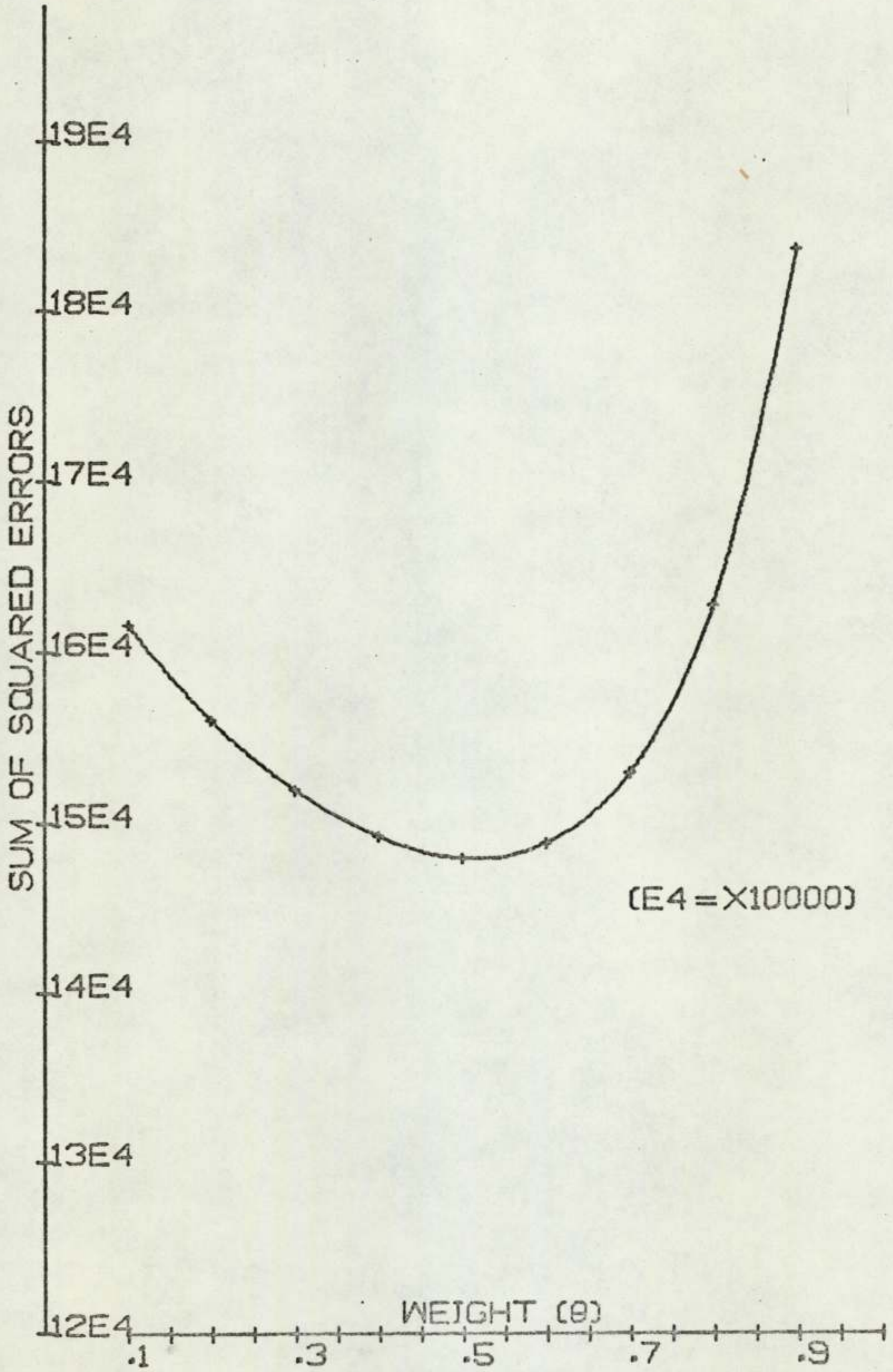


FIG (4.10)

figs. (4.7) - (4.9) inclusive and the observed errors are presented in Table (4.4).

The Box-Jenkins (p,d,q) models were selected for their potential in revealing structure in the demand data, and this they appeared to do. Higher than anticipated autocorrelations at lags of 4-5 weeks and at 13 weeks were detected and seem reasonable. One of the immediately apparent features of fig. (4.3) is the yearly seasonal effect and thus is evident in the correlogram of the series shown in fig. (4.5). The negative autocorrelation approximately centred about lag 33 is diffuse and is difficult to explain causally. It might be evidence of a weak seasonal effect based on a three yearly cycle, but on the other hand it might be completely spurious as the large variance associated with the sampling distribution of correlograms has already been remarked upon (section 4.3.3).

Despite the apparent reasonableness of the above picture of demand, it proved impossible to incorporate the various aspects into an efficient seasonal model. As Table (4.3) shows, the simple (1,0,0) model gave best results in terms of least squares fit, although this model does not explicitly take into account any of the seasonal effects - including the obvious yearly effect. In attempting to fit a multiplicative seasonal model to the data it was found impossible to fit an adequate subsequent model to the residuals obtained after extracting the seasonal component. The effect of fitting the second model was to increase the sum of squares of residuals to an amount greater than that obtained after fitting the first model.

Although the (p,d,q) models ultimately gave the best fit to the data series this was only achieved at the expenses of considerable computation and the plotting of a large number of correlograms and residuals. In terms of the root mean square error of the best model fitted it is debatable whether the much simpler approach described in section 4.3.4

Table (4.4) A COMPARISON OF FORECASTS OBTAINED FROM THREE MODELS APPLIED TO 1970 DATA. ($\frac{3}{4}$" TABLE "A" DEMAND)

	Exponential (A) Smoothing (α=0.4)		Box-Jenkins (B) (1,0,0) ϕ_1 = 0.6		Adaptive (C) Smoothing-Trigg's T.S.	
	Error	$(\text{Error})^2$	Error	$(\text{Error})^2$	(Error)	$(\text{Error})^2$
1	0	0	- 8.1	65.61	- 67.85	4603.62
2	12.16	147.87	- 11.3	127.69	21.31	454.12
3	26.32	692.74	0.44	0.19	9.73	94.67
4	16.92	286.29	- 10.36	107.33	4.52	20.43
5	35.78	1280.21	14.38	206.78	27.20	739.84
6	0.74	0.55	- 21.48	461.39	- 7.34	53.88
7	6.86	47.06	- 2.80	7.84	1.18	1.39
8	- 23.72	562.64	- 34.62	1198.54	- 27.32	746.38
9	19.92	396.81	15.84	250.91	15.39	236.85
10	18.38	337.82	1.58	2.50	14.48	209.67
11	- 11.68	136.42	- 24.86	618.02	- 12.78	163.33
12	- 33.52	1123.60	- 37.92	1437.93	- 37.36	1395.77
13	- 5.9	34.81	- 8.24	67.90	- 14.40	207.36
14	- 3.24	10.50	- 16.40	268.96	- 8.20	67.24
15	- 1.44	2.07	- 16.40	268.96	- 4.21	17.73
16	- 0.84	0.71	- 16.40	268.96	- 2.03	4.12
17	- 0.24	0.06	- 16.40	268.96	- 0.94	0.88
18	- 23.74	563.59	- 40.50	1640.25	- 24.53	601.72
19	33.40	1115.56	21.46	460.53	36.46	1329.33
20	- 8.20	67.24	- 35.64	1270.21	- 16.27	264.71
21	- 22.46	504.45	- 36.38	1323.50	- 31.59	997.93
22	- 1.40	1.96	- 14.14	199.94	- 11.45	131.10
23	85.60	7327.36	65.16	4245.83	79.87	6379.22
24	17.32	299.98	- 20.96	439.32	5.68	32.26
25	- 38.42	1476.10	- 49.42	2442.34	- 44.26	1958.95
26	- 13.18	173.71	- 10.16	103.23	- 26.60	707.56
27	22.00	484.00	13.46	181.17	5.76	33.18
28	- 17.80	316.84	- 35.24	1241.86	- 26.61	708.09
Totals	89.62	17,390.95	- 335.40	19,176.65	- 142.16	22,161.32

Note: (i) There is no significant difference between Mean Square Error of (A) & (B) - ($t = 0.4$, $p > 0.2$) or of (B) & (C) - ($t = 0.53$, $p > 0.2$)

(ii) Initialisation procedures: (A) forecast set to actual value (C) initial forecast taken from an earlier run is not to be preferred in terms of economy of effort and ultimate performance.

The simple exponential smoothing and adaptive smoothing models were used when the results of fitting the (p,d,q) models proved disappointing. Contrary to initial expection the simple exponential smoothing model was able to fit the data better than the adaptive smoothing model. Subsequent investigation showed that in the latter case 'α' the coefficient of smoothing had ranged between 0.84 and 0.05 with an average value of 0.36. During the seasonal increase in demand observed during the last quarters 1967 and 1968 'α' had achieved its maximum values, the model interpreting the increase as a shift in process mean. A secondary effect of this would be to cause the forecast to be more affected by weekly fluctuations and this undoubtedly contributed to the increased sum of squared errors observed. In the simple exponential smoothing model, 'α' was fixed and would not be affected in this way. Obviously both models would have given better results if the seasonal effect had been removed. The removal of seasonal effects is discussed later.

In general it is not possible to fit a "best" simple exponential smoothing constant to a data series using the minimum squared error (M.S.E.) criterion used above. This is because exponential smoothing is an averaging process in which the number of terms effectively included in the average is governed by α. By progressively reducing the value of alpha, the number of previous data points incorporated in the average is increased until, when $\alpha = 0$, the forecast is completely fixed by previous observations and is unaffected by new results. In the case of a constant exponential smoothing model of the form used above it may be shown (Brown, 1962, p. 106 et. seq.) that,

$$\text{M.S.E.} = \frac{2 \cdot \sigma_x^2}{2 - \alpha}$$

and thus $\alpha = 0$ minimises the error providing (a) that the process mean is accurately known (b) that the inputs are purely random and (c) that the process mean does not change. However Brown (ibid) also shows that when inputs are correlated as in the present case it is possible to obtain a minimum squared error with $\alpha \neq 0$. He indicates that with correlated results, a smaller value of α should suffice than with uncorrelated inputs. This is in contrast with the above findings that the "best" value is given with $\alpha = 0.4$. This may be explained by the presence of the yearly seasonal effect. Brown suggests that when it appears that a value of α greater than 0.3 is desirable then the possibility of a seasonal variation being present should be investigated.

Seasonal factors derived from only three cycles are necessarily limited in their accuracy, although the computation could be carried out relatively simply using the ratio-to-moving-average method. In the present study the calculation of seasonal factors for application to the simple exponential smoothing and adaptive smoothing models was not attempted however. By the time the work relating to these models was carried out it had become clear that few benefits would accrue to the wider study by extending the models.

The overall picture that emerged from the intensive study of one representative of the high production volume group of products discussed in 4.2 was of an extremely noisy, relatively constant demand function on to which a yearly seasonal variation was super-imposed. Buried in the noise were possibly two further components associated with 4 and 13 week period respectively. However the latter components were not sufficiently pronounced to be useful in forming the basis of forecasting models. The best model developed explained only approximately 40% of the original variance. It seemed likely that the much simpler exponential smoothing or adaptive smoothing models could have given improved if not similar results if attempts had been made to allow for seasonal variation.

At the present time therefore it must be concluded that the series are unsuitable for forecasting even one week ahead, while with increasing lead time the forecast error variance would increase further. A more orderly time series of demand could be obtained by grouping weekly demand over 4 week intervals to give mean weekly demand. This would reduce the variance of the original series. If the individual members of the series were independently distributed with identical variance the above grouping procedure would produce a new series of mean weekly demand having exactly half the variance of the original series. However the positive autocorrelation exhibited by members of the series studied above would make the reduction in variance observed in the present case rather less than a factor of two. Unfortunately, grouping data in the above way would also reduce the number of data points by a factor of four, thus leaving very few points to be used for fitting forecasting models to.

4.4 COMPANY STOCK HOLDING POLICY

Even when demand for a product is sufficiently high to justify continuous production it is still necessary to keep stocks because actual demand usually fluctuates about its average or forecast level. In the present study minimum stock levels had been issued by the production director although the basis upon which they were compiled was never satisfactorily established. Instead they were accepted as being the operating rules of the Company at that time and were used in the production allocation system described in the next chapter. (However provision was made for the specific values of the maximum and minimum stock levels to be easily changed). This section tentatively examines the customer service levels provided by the minimum stock levels according to simple stock control theory.

In theory minimum stock levels are determined by the requirement that sufficient stock is held to meet demand through the period of the lead time, where the lead time is defined as the time taken from placing an order to having the ordered supply in stock and ready for distribution. The minimum stock level, therefore, must be at least equal to the expected level of demand in the lead time. Thus if average weekly demand is $\bar{D}$ and the lead time L weeks, the stock required to meet average demand during the period of the lead time is given by $L.\bar{D}$ and is known as the period stock. However, as already indicated, actual demand deviates from its average value and to ensure that stocks are adequate a further amount of stock, called the buffer stock, must be kept to protect against unpredictable variations in demand during the lead time. If it is assumed that the variation of demand about its average value is distributed according to a normal distribution having variance σ_D^2 then assuming independence between individual levels of actual demand, the variance of demand over the lead time is given by $L.\sigma_D^2$ and has a standard distribution of $\sigma_D\sqrt{L}$. Since the probabilities associated with a particular standard normal deviate z have been tabulated (see, for example, Tables for Statisticians and Biometricians, Pearson, 1954) it is possible to arbitrarily select a value for z giving a known probability of the buffer stock being exhausted before the renewal stock is received. Thus $z = 1.65$ gives a 5% probability of depletion and $z = 2.33$ gives a 1% probality and hence define a customer service levels of 95% and 99% respectively. The associated buffer stocks are given by $1.65\sigma_D\sqrt{L}$ and $2.33\sigma_D\sqrt{L}$. Therefore minimum stock level M is given by

$$M = \bar{D}.L + z.\sigma_D\sqrt{L} \qquad \text{Equ. (4.4.1)}$$

In the present application L could take two values in some cases. This situation arose because certain products in the domestic tubes product group were made on a continuous basis rather than by batch, and accordingly it was possible to divert partially processed material from one production line to make a related product on another. When products were made in this way the lead time was two weeks (L = 2). However, if an order had to be made from raw materials thus requiring casting and extrusion to be carried out, the lead time extended to six weeks or longer depending on the amount of spare capacity existing in the casting and extrusion departments. In practice it was attempted at all costs to avoid disrupting the casting and extrusion programmes and thus it was unusual to fall back to a six week lead time. A detailed discussion of the production planning procedure is given in chapter 5.

The relationship given by equation (4.4.1) was used to calculate the customer service levels associated with the existing minimum stock levels and to calculate new stock levels based on customer service levels of 95% and 99%. This was done assuming lead times of 2 and 6 weeks. The results are presented in Tables 4.5 and 4.6. Only Table 'X' (the metric equivalent of Table 'A') was used as this product type accounted for approximately 90% of domestic product group demand.

Before these results are discussed, simplifying assumptions made in arriving at equation (4.4.1) will be examined. Firstly it was assumed that an order would be placed the moment the minimum stock level was broken. In practice this would be unlikely, as the receipt of a large order when stock was just above the re-order point would cause the new stock level to fall well below the minimum at the start of the lead period, and thus the period stock calculated as $L.\bar{D}$ would be inadequate. A practical scheme would have to make a correction for this effect. Also, in reality, the lead time would not necessarily be constant and a further correction would have to be made for this. Finally, the results presented in 4.3 show that the individual values of demand are not independent but are autocorrelated.

However the purpose of the exercise reported in Tables 4.5 and 4.6 was not to provide a definitive statement about optimum stock levels but to provide an initial look in a consistent manner at the situation as it

Table (4.5) CUSTOMER SERVICE LEVELS (2 WEEK LEAD TIME)

Table 'X' Size (mm)	Mean Weekly Demand ($\bar{D}$) (tons)	Standard Deviation σ_D (tons)	Standard Normal Deviate Z^*	Company Minimum Stock Level (tons)	Existing Customer Service Level (%)	Stock Level for 95% Service (tons)	Stock Level for 99% Service (tons)
8	0.3	0.8	3.89	5	>99.99	2.46	3.24
12	0.6	0.3	9.05	5	>99.99	1.90	2.19
15	117.1	39.8	1.17	300	87.90	326.93	365.58
22	112.0	23.7	2.27	300	98.84	279.22	302.21
28	34.4	16.8	3.42	150	99.97	107.94	124.24
35	11.2	4.1	13.38	100	>99.99	31.95	35.93
42	6.8	3.7	12.70	80	>99.99	22.22	25.81
54	4.8	2.5	14.24	60	>99.99	15.43	17.85
76	1.0	0.7	28.28	30	>99.99	3.63	4.31
108	1.1	0.8	24.60	30	>99.99	4.06	4.84
133	0.2	0.3	10.95	5	>99.99	1.10	1.39
159	1.1	1.0	1.99	5	97.67	4.53	5.50
Total				1070		801.37	893.09

*Standard normal deviate of demand in the lead time, computed from Equ (4.4.1).

Table (4.6) CUSTOMER SERVICE LEVELS (6 WEEK LEAD TIMES)

Table 'X' Size (mm)	Mean Weekly Demand ($\bar{D}$) (tons)	Standard Deviation σ_D (tons)	Standard Normal Deviate $\underline{z}$*	Company Minimum Stock Level (tons)	Existing Customer Service Level (%)	Stock Level for 95% Service (tons)	Stock Level for 99% Service (tons)
8	0.3	0.8	1.63	5	94.84	5.03	6.37
12	0.6	0.3	1.89	5	97.06	4.81	5.31
15	117.1	39.8	- 4.13	300	< .01	863.39	929.86
22	112.0	23.7	- 6.41	300	< .01	767.75	807.33
28	34.4	16.8	- 1.37	150	8.53	274.27	302.33
35	11.2	4.1	3.26	100	>99.94	83.85	90.61
42	6.8	3.7	4.32	80	>99.99	55.75	61.93
54	4.8	2.5	5.09	60	>99.99	38.90	43.08
76	1.0	0.7	13.95	30	>99.99	8.83	10.00
108	1.1	0.8	11.94	30	>99.99	9.83	11.17
133	0.2	0.3	5.14	5	>99.99	2.41	2.91
159	1.1	1.0	- 0.65	5	25.78	10.64	12.31
Total				1070		2125.46	2283.21

*Standard normal deviate of demand in the lead time, computed from Equ (4.4.1) existed at that time. In this way it was hoped to discover whether potential existed for the reduction of stock levels. It is concluded that such potential does indeed exist, even after an allowance is made for the approximate nature of the calculations.

Thus it can be seen that the levels of customer service the various minimum stock levelsyield are extremely high, corresponding to standard normal deviates of up to $z = 28$. Strangely, the highest volume products (15 mm., 22 mm. and 28 mm. sizes) have the lowest customer service levels, although with the exception of the 15 mm. size, these too are very high. The reason for the very high stock levels appears to be largely historical and dates from when it was a Company policy to explicitly offer 100% customer service on stock items. Of course the variability of supply and demand meant that on occasion stock-outs would occur, but this was regarded very seriously by senior management and reportedly the re-occurence of such a situation was combatted by disciplinary action and presumably attempts to further increase stock. However in later years the raw material, copper, began to become a scarce resource and the benefits of low stock levels became more apparent. The tables indicate that opportunity exists for these benefits to be realised to a greater degree.

Considering Table (4.5), since a lead time of 2 weeks is more relevant to the real life situation, it can be seen that to obtain a 99% customer service level a total minimum stock holding of 899 tons is required, while the existing minimum stock holding involved 1070 tons. Assuming a cost of £700 ton for copper this would represent a saving of rather more than £120,000 in raw material alone. Since the stocks are of processed product, the capital saving would be considerably more.

As the above findings are based on values of $\overline{D}$ and σ_D produced by the PSALM system as described in Appendix M1, they are calculated on the demand recorded for only one quarter and it might be thought that these values would not be typical. In fact they compare remarkably closely with mean weekly demands and standard deviations calculated for high volume products in 1969 as may be seen from Table 4.7. The latter statistics are based on 52 weeks' demand. This correspondence is to be expected from the model of demand suggested in 4.3 which was a constant model incorporating a seasonal trend.

Another reason why the above potential savings are conservative arises because equation (4.4.1) does take account of the fact that some of the products are made on a continuous basis. Assuming an average production rate $\overline{P}$ during the lead time, equation (4.4.2) may be written as follows

$$M = \overline{D}.L - \overline{P}.L + Z\sqrt{L(\sigma_D^2 + \sigma_P^2)}$$

$$= L\ (\overline{D} - \overline{P}) + Z\sqrt{L(\sigma_D^2 + \sigma_P^2)} \qquad (4.4.2)$$

where σ_p = standard deviation of achieved production about its mean $\overline{P}$

It may be assumed by a consideration of the relative magnitudes of $L.\overline{D}$ and $Z \cdot \sigma_D \sqrt{L}$ that although the buffer stock would be increased by the introduction of the σ_p^2 component, this would be small compared with the reduction in M caused by the difference term $(\overline{D} - \overline{P})$. Thus it can be seen that continuous production effectively reduces demand in the period of the lead time.

A final point that must be appreciated is that the suggested savings could only be achieved if demand were to be satisfied from a single supply point. In reality this was effectively the case as material in short supply would be transferred between warehouses. If, however, the 15 warehouses were regarded as independent supply points the total amount of buffer stock required throughout the Company would increase.

Table 4.7 DEMAND STATISTICS

Product	1969 (based 52 week's demand)		1972 (based on 1st quarter-13 weeks)	
	D	σ_D	D	σ_D
$\frac{1}{2}$" Table 'A'/15 mm Table 'X'	118.98	39.50	117.1	29.8
$\frac{3}{4}$" Table A/22 mm Table 'X'	109.07	30.39	112.0	23.7
1" Table A/28 mm Table 'X'	38.18	13.51	34.4	16.9
$1\frac{1}{4}$" Table A/35 mm Table 'X'	25.18	13.96	11.2	4.1
$1\frac{1}{2}$" Table A/42 mm Table 'X'	13.46	5.33	6.8	3.7

D = mean weekly demand

σ_D = standard deviation

5. PRODUCTION SMOOTHING AND ALLOCATION

This chapter is concerned with the problem of production smoothing and allocation. It is suggested that if the problem is approached by means of an appropriate set of consistent rules and procedures, then in the long run such a system would give a better performance than the then current practice which relied heavily on the experience and intuitive skills of the allocator. After the present method of allocation has been described in 5.1, section 5.2 presents a set of procedures which form a basis for testing the above hypothesis. Section 5.3 describes salient features of the actual model devised (PSALM), further details being given in Appendices M1 and M2. Test results from data relating to the final quarter 1971 are presented and discussed in section 5.4.

5.1 THE PRESENT ALLOCATION SYSTEM

5.1.1 The Factors involved in Production Allocation

The basis of allocation was an attempt to achieve a constant labour capacity. (The term "constant labour capacity" is used here to mean that work was organised to avoid laying men off in some weeks of low demand and working overtime in subsequent weeks of high demand). This is considered in more detail in the next section.

Other important facts were (i) the quarterly demand forecast (ii) production efficiency and (iii) product interchangeability.

(i) Quarterly Demand Forecast

Table (5.1) shows the product types and expected demand for a typical quarter.

(ii) Production Efficiency

In considering how expected demand should be met in the coming quarter, an estimate of the current and expected efficiency of production had to be made. Thus potential capacity might not be available because of plant overhaul, high levels of illness absence etc.

Table (5.1) FORECAST OF QUARTERLY DEMAND

Product Type No.	Product Type	Demand (tons)
1	B.S. 659 Table 'A'	4,475
2	Minibore	40
3	B.S. 3931	270
4	B.S. 1386	470
5	N.Z. Domestic	90
6	Metric Domestic	600
7	Miscellaneous	1,360
8	Skin hard/Level wound	200
9	Commercial coils	210
10	Refrigerator coils/straights	240
11	Restrictor/Capillary	45
12	Redraw	360
13	Tube for fittings manufacture	1,000

A forecast similar to the one above was available quarterly and was made by Marketing Department. Additional shorter term forecasts could be available more frequently for certain product types.

(iii) Product Inter-changeability

In practice certain products were made on the same plant. However, because of differences between such products, largely relating to their dimensions, particular items of plant had different capacities depending on the product being made at the time. This gave rise to the notion of capacity inter-change factors whereby 'x' tons of product 'A' were equivalent to 'y' tons of product B. The interdependence of certain products on some plant had to be taken into account when production allocations were made.

5.1.2 Combination of Production Allocation Factors

On the basis of (5.1.1 (i) - (iii)) above, an amended production forecast was arrived at for all product types except numbers 1, 3 and 4 in Table (5.1), and the remaining production capacity was established. (As orders were received for the former product types they would be treated as bespoke work).

A similar calculation was then performed for product types 1,3 and 4 and the capacity required to meet expected demand was compared with that available. Since demand for these product types was met via Company stock it was sufficient in general that stock on hand plus capacity available during the coming quarter was equal to, or was in excess of expected demand. (It will be seen in section (5.2.3) that this was not always a sufficient condition). If this was not so, capacity had to be made available to produce more of types 1, 3 and 4 during the quarter. This was done at the expense of bespoke work on which increased delivery dates were quoted.

If there was excess capacity even when expected demand for all product types had been allowed for, some or all of it could be used to keep labour capacity filled. That is, more of product types 1, 3 and 4 could be made for stock. Thus the function of these product types was that of "load balancing".

In addition, production could be allowed to exceed demand in order to build up notional stocks such as "holiday stocks" used to meet demand during factory shut-downs. This is termed "policy smoothing".

5.2 A METHOD OF PRODUCTION SMOOTHING

Broadly speaking the rules and procedures described below are a formalisation of what is currently practiced by the allocator, so that although some improvement may possibly result from the increased consistency achieved by formalisation, the real benefits accrue by virtue of closer monitoring of stock movements, both actual and projected, and reduced demands on the allocator's time.

Two further benefits may also be obtained. Firstly the allocation function may be performed by a man of less experience and secondly the basic allocation model may be used to evaluate the effect of changes to the existing system. Thus changes in production loadings together with their associated stock levels may be projected for various production efficiencies, demand forecasts etc.

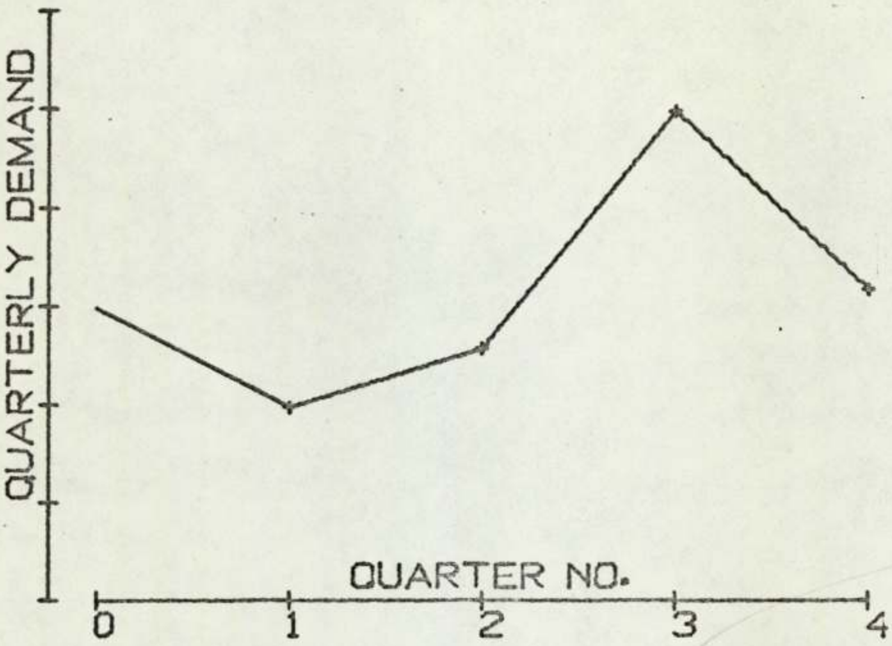
5.2.1 Schematic Outline

This section describes in principle one way in which production smoothing may be achieved.

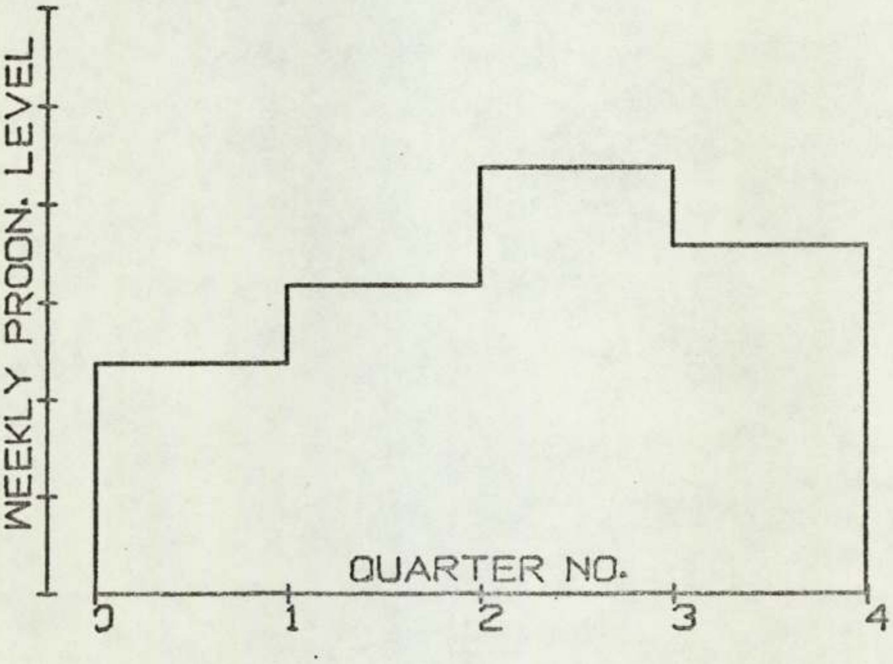
Since product types other than numbers 1, 3 and 4 in Table (5.1) are made to order, they are not available for smoothing production except in the sense that if total capacity is filled, bespoke order delivery dates may be extended until spare capacity appears. Only product types 1, 3 and 4 are considered below.

Experience shows that there is a strong seasonal demand for these products, based on the period of one year, and this permits "seasonal smoothing". Thus fig. (5.1) shows the pattern of demand in a typical year.

(FIG (5.1)


QUARTERLY DEMAND
QUARTER NO.
0
1
2
3
4

(FIG (5.2)


WEEKLY PRODN. LEVEL
QUARTER NO.
0
1
2
3
4

If production were to meet the quarterly forecast demand, this would imply the production levels of fig (5.2).

However if the presence of seasonal variation is anticipated, the production level can be smoothed to meet yearly forecast demand so giving seasonal smoothing as shown in fig. (5.3).

The scheme shown in fig. (5.3) would give rise to the stock levels shown in fig. (5.4), while that shown in fig. (5.2) would give stock levels which did not differ significantly from the base stock level.

Thus seasonal smoothing permits constant production levels at the expense of an increased stockholding.

5.2.2 Response to Trend

Although the scheme described in 5.2.1. will smooth out seasonal variation, it will respond to an overall trend. Such a trend is shown in fig. (5.5) which would be seasonally smoothed as follows:-

(a) a smoothed production level would be calculated based on quarters 1-4 as described in 5.2.1.

(b) at the end of quarter 1, a revised forecast for quarters 2-4 would be available together with a first forecast for quarter number 5.

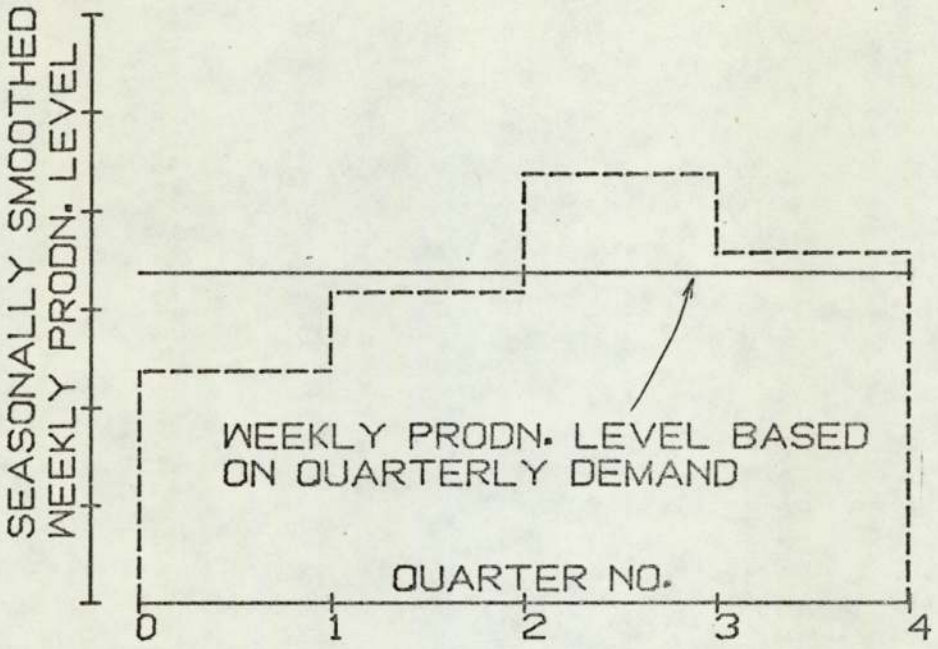
(c) A new smoothed production level would be calculated for the period, quarter numbers 2-5, bearing in mind the stock available at the start of quarter number 2. This is shown in fig. (5.6).

It will be seen that the smoothed production level is a moving average of forecast demand for the coming year and thus it will reflect an overall change in demand as shown in figs. (5.5) & (5.6), while remaining relatively insensitive to seasonal variations which occur over time periods less than that of the extent of the moving average.

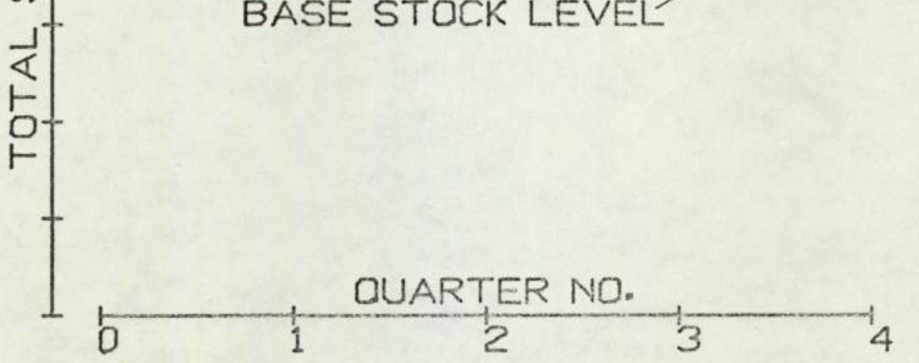
5.2.3 Short Term Adjustment of Production Level

Although the scheme described in 5.2.2. will accommodate long term trends in both directions, under certain circumstances it can give rise to

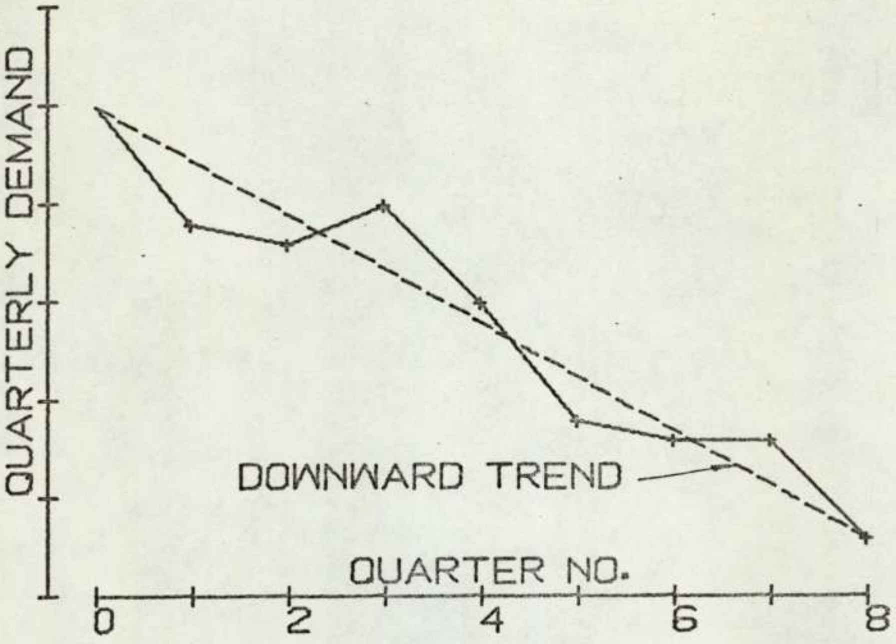
(FIG (5.3)


SEASONALLY SMOOTHED
WEEKLY PRODN. LEVEL
WEEKLY PRODN. LEVEL BASED
ON QUARTERLY DEMAND
QUARTER NO.
0
1
2
3
4

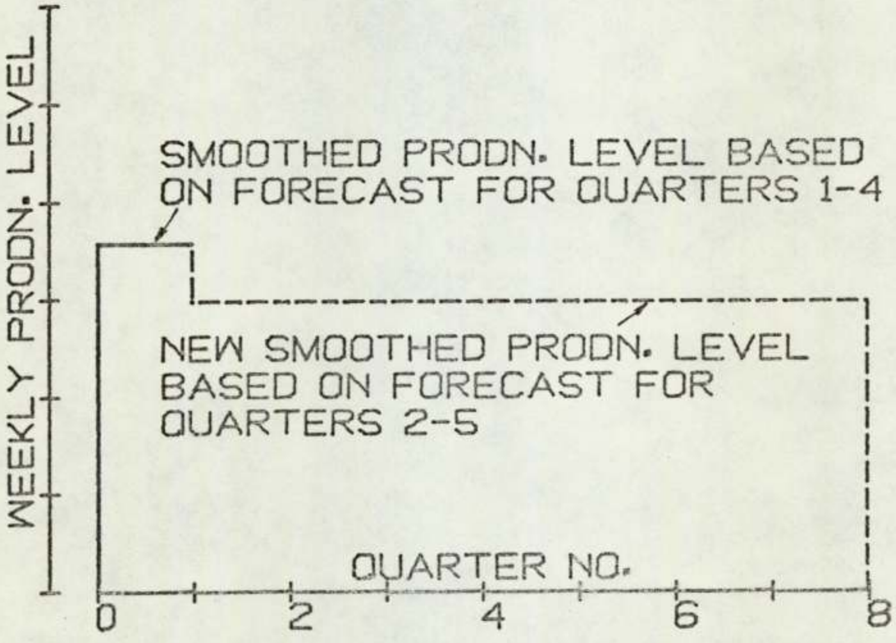
(FIG (5.4)


TOTAL STOCKS
ACTUAL STOCK
BASE STOCK LEVEL
QUARTER NO.
0
1
2
3
4

(FIG (5.5)


QUARTERLY DEMAND
DOWNWARD TREND
QUARTER NO.
0
2
4
6
8

(FIG (5.6)


WEEKLY PRODN. LEVEL
SMOOTHED PRODN. LEVEL BASED
ON FORECAST FOR QUARTERS 1-4
NEW SMOOTHED PRODN. LEVEL
BASED ON FORECAST FOR
QUARTERS 2-5
QUARTER NO.
0
2
4
6
8

short term stock deficiencies. Circumstances which would give rise to this are depicted in fig. (5.7) where seasonal variation has been intensified in the second year without a change in the overall trend.

Fig (5.8) shows the same situation for quarters 1-4 after production smoothing for seasonal variation. The quarterly forecasts (F) of demand are shown and it can be seen that the excess $(E_1 + E_2)$ of smoothed production over demand in the first and second quarters is equal to the deficiency $(D_3 + D_4)$ of smoothed production with respect to the demand in the third and fourth quarters. This, of course, is simply the result of the smoothing operation.

Assuming that the system started at period 0 with only the base stock S_o, which is to be preserved at all times to meet contingencies, the weekly smoothed production level for the first quarter P, is given by,

$$P_1 = \frac{F_1 + F_2 + F_3 + F_4}{52} \qquad \text{Equ (5.2.1)}$$

However at the end of the first quarter, a new forecast is made, which for the first time includes quarter number 5 for which demand is very low. (For simplicity it is assumed that the new forecast demands for quarters 2-4 inclusive remain unchanged). At this time the new stock level $(S_1 = S_o + E_1)$ has increased by E_1. On production smoothing for periods 2-5, the weekly production level P_2 would be calculated so that,

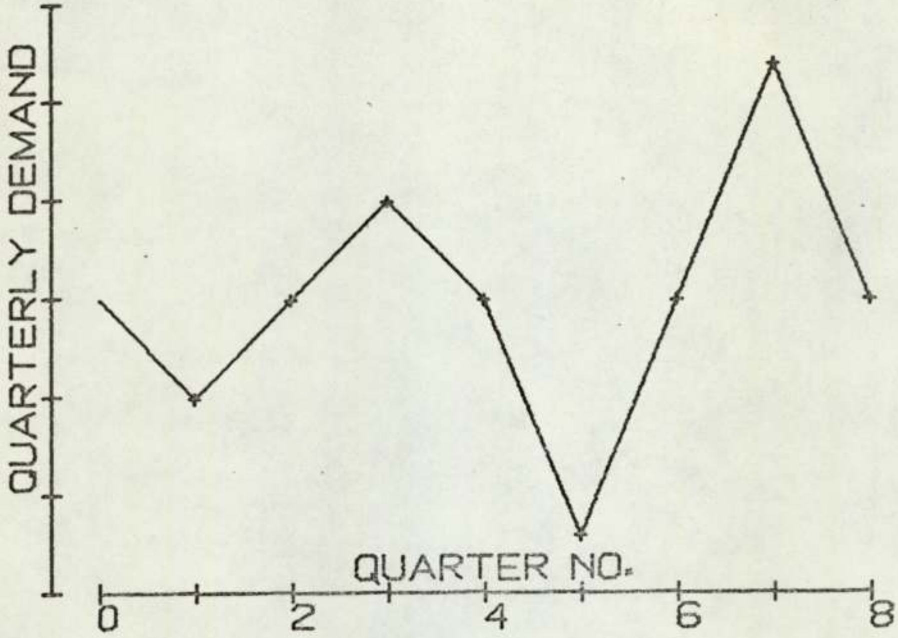
$$52\,P_2 + E_1 = F_2 + F_3 + F_4 + F_5$$

$$\text{ie } P_2 = \frac{F_2 + F_3 + F_4 + F_5 - E_1}{52} \qquad \text{Equ (5.2.2)}$$

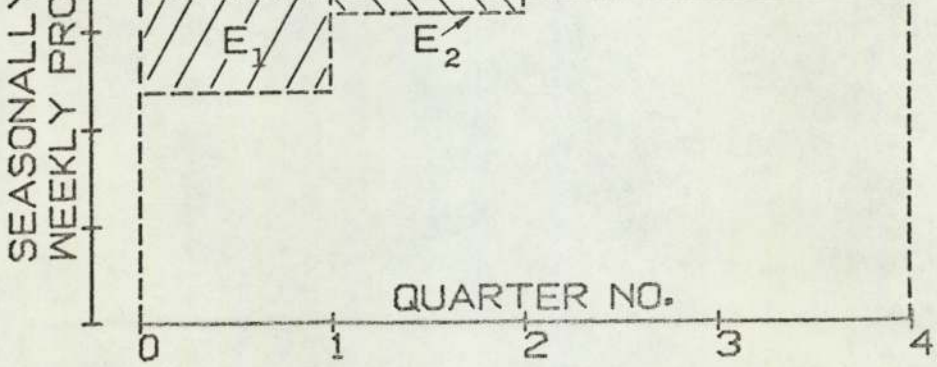
Since $F_5 < F_1$ from fig. (5.7), and the right hand side of equation (2) is reduced by E_1, it can be seen by comparison of equations (1) and (2) that $P_2 < P_1$.

Inspection of fig. (5.8) shows the effect of this is to reduce E_2 (possibly to make $E_2 \rightarrow D_2$) and to increase D_3 and D_4 so breaking the

(FIG (5.7)


QUARTERLY DEMAND
QUARTER NO.
0
2
4
6
8

(FIG (5.8)


SEASONALLY SMOOTHED WEEKLY PRODN. LEVEL
E_1
E_2
D_3
D_4
QUARTER NO.
0
1
2
3
4

equality $E_1 + E_2 = D_3 + D_4$; the new situation being represented by

$$E_1 + E_2 < D_3 + D_4 \quad \text{(or possible } E_1 < D_2 + D_3 + D_4\text{)}$$

Thus the expectation of fulfilling demand in quarters 3 & 4 from stock accumulated in quarters 1 & 2 is diminished, because subsequent smoothing at the end of period one has reduced, possibly completely removed, the contribution to stock in quarter 2(E_2). Unless the forecast for quarter number 6 drastically changes the smoothed production level for the third quarter, there will be insufficient stock to supplement actual production in order to meet demand in that quarter, even though in the long term (ie the coming year) demand and production are perfectly matched.

If the possibility of a short term stock shortage were detected, the production level would have to be increased to the minimum level necessary to meet demand in the quarter in question, and the position reviewed again at the end of the quarter in the light of the new smoothed production level. If such an occurence took place at a time of full capacity working, bespoke work would have to be promised on increased delivery.

5.2.4 Updating the Smoothing System

Sections 5.2.1 to 5.2.3 describe an ideal system utilizing perfect forecasts. In reality, for the purposes of monitoring allocation on a weekly basis it is assumed that, because no weekly forecasts exist, weekly demand is in linear proportion to the quarterly forecast, although this is not necessarily the case. In addition the quarterly forecasts themselves are always in error to some extent, and finally the production requests made on the factories are not reliably met. All these sources of error mean that if forecast and requested quantities are not replaced by the actual values as they become available, the smoothing system will rapidly deviate from reality and its predictions will become meaningless.

In practice such actual values become available weekly and thus updating could be carried out with a maximum frequency of one week, or possibly at longer intervals, to obtain the latest stock projections. Similarly a new projection could be obtained when new short term forecasts became available to replace the weekly forecasts obtained by linear proportion from the quarterly ones.

5.2.5 Policy Smoothing

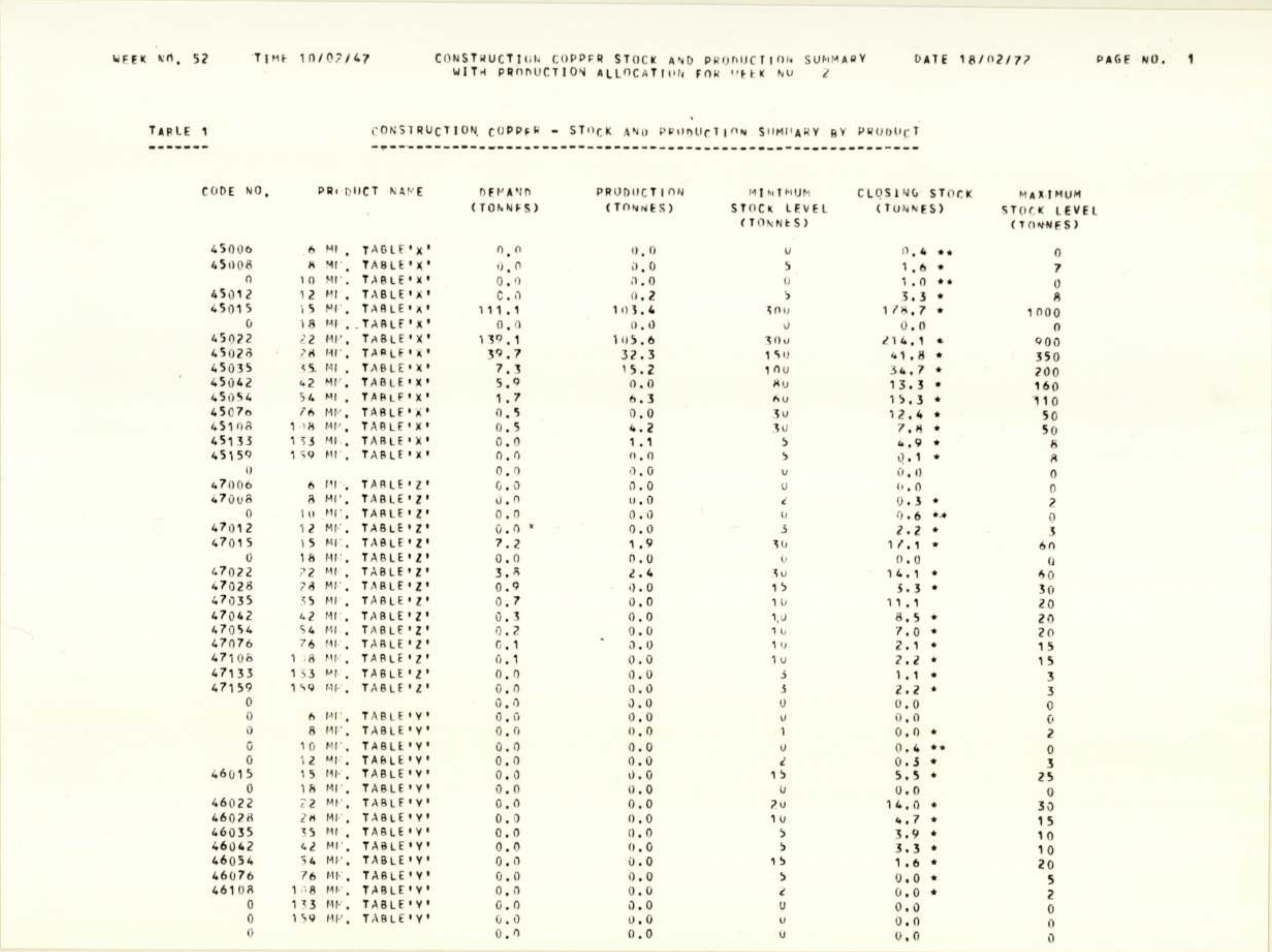
In addition to the seasonal smoothing already described, it may be decided to make stock in excess of long term forecast demand with the intention of filling labour capacity. Clearly such stock will accumulate until either demand increases and/or the production level is reduced.

5.3 PSALM: PRODUCTION SMOOTHING AND ALLOCATION MODEL

The PSALM system embodies the principles described in 5.2 and additionally provides a means of monitoring various relevant variables. A detailed description of the functions carried out by PSALM is given in appendix M1 and will not be repeated here. Chapter 3 of that Appendix refers to the print-out obtained from PSALM which is reproduced in figs. (5.9) - (5.19) inclusive. It should be noted that Figs. 5.11 and 5.15 only show part of Tables 4 and 6 respectively in the print-out. In each case the layout is repeated a further three times to present information for the remaining factories not shown. This is also the case in figs. (5.17) and (5.18) Fig. (5.16) shows Table 7 of the print-out which would normally contain the results of the cusum analysis. This table may contain as many entries as there are forecasts of demand/production budgets to be monitored. For reasons given in 5.4 the cusum facility finally was not used in the trial quarter - it will be seen that a forecast of zero is contained in the output.

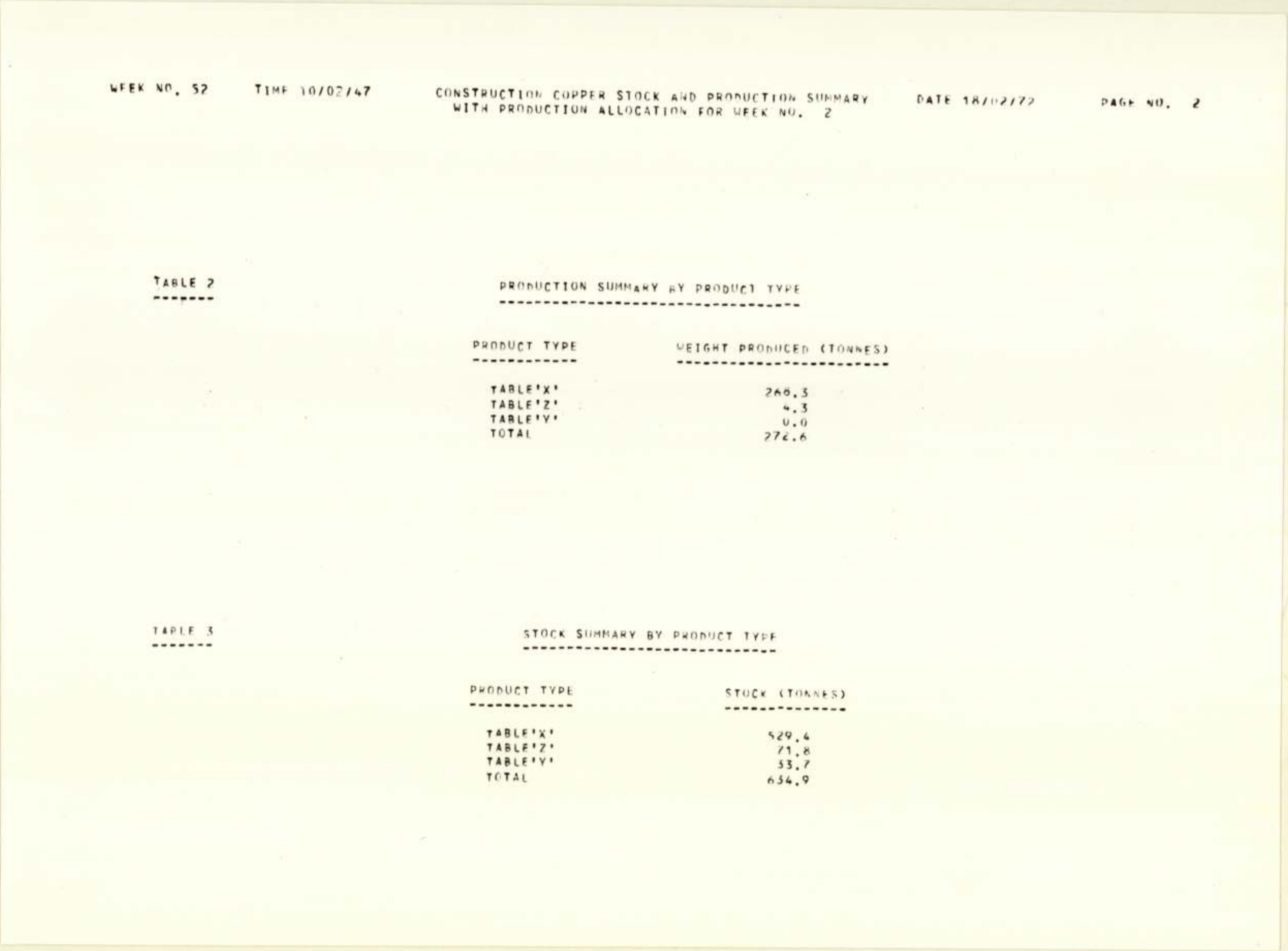
In designing the layout of the output the findings of Hitt (1961)* were taken into consideration. However since the only output device available was a line printer this introduced severe restraints upon what was possible. In particular no graph plotting facilities

* See also Slutz(1961a&b)



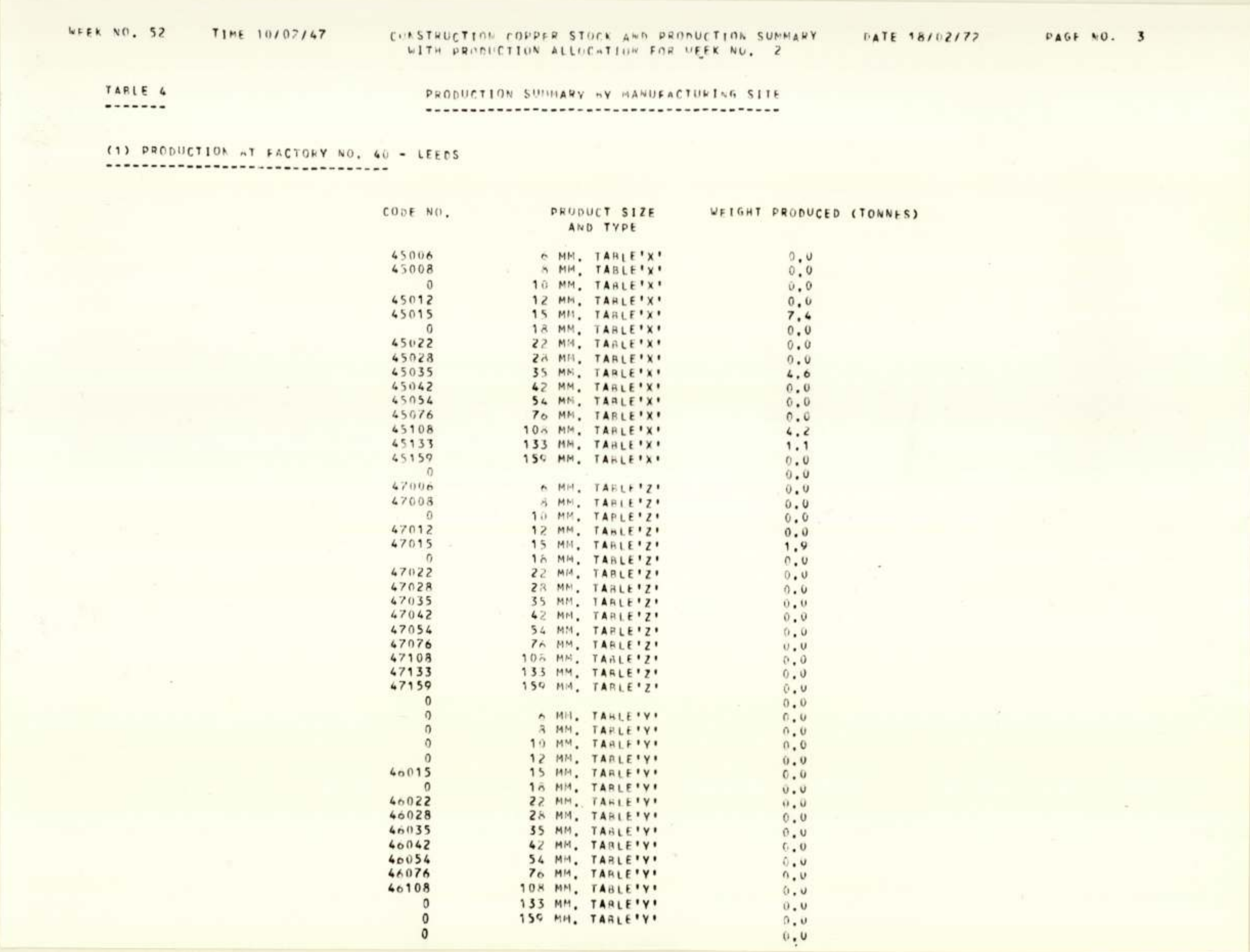
```
WEEK NO. 52      TIME 10/02/47        CONSTRUCTION COPPER STOCK AND PRODUCTION SUMMARY      DATE 18/02/72          PAGE NO.  1
                                        WITH PRODUCTION ALLOCATION FOR WEEK NO    2


     TABLE 1                   CONSTRUCTION COPPER - STOCK AND PRODUCTION SUMMARY BY PRODUCT
     -------                   --------------------------------------------------------------

          CODE NO.       PRODUCT NAME        DEMAND          PRODUCTION         MINIMUM       CLOSING STOCK       MAXIMUM
                                            (TONNES)          (TONNES)        STOCK LEVEL       (TONNES)        STOCK LEVEL
                                                                                (TONNES)                          (TONNES)

           45006       6 MM. TABLE'X'         0.0               0.0               0             0.4 **              0
           45008       8 MM. TABLE'X'         0.0               0.0               5             1.6 *               7
               0      10 MM. TABLE'X'         0.0               0.0               0             1.0 **              0
           45012      12 MM. TABLE'X'         0.0               0.2               5             3.3 *               8
           45015      15 MM. TABLE'X'       111.1             103.4             300           178.7 *            1000
               0      18 MM. TABLE'X'         0.0               0.0               0             0.0                 0
           45022      22 MM. TABLE'X'       139.1             105.6             300           214.1 *             900
           45028      28 MM. TABLE'X'        39.7              32.3             150            41.8 *             350
           45035      35 MM. TABLE'X'         7.3              15.2             100            34.7 *             200
           45042      42 MM. TABLE'X'         5.9               0.0              80            13.3 *             160
           45054      54 MM. TABLE'X'         1.7               6.3              60            15.3 *             110
           45076      76 MM. TABLE'X'         0.5               0.0              30            12.4 *              50
           45108     108 MM. TABLE'X'         0.5               4.2              30             7.8 *              50
           45133     133 MM. TABLE'X'         0.0               1.1               5             4.9 *               8
           45159     159 MM. TABLE'X'         0.0               0.0               5             0.1 *               8
               0                              0.0               0.0               0             0.0                 0
           47006       6 MM. TABLE'Z'         0.0               0.0               0             0.0                 0
           47008       8 MM. TABLE'Z'         0.0               0.0               2             0.3 *               2
               0      10 MM. TABLE'Z'         0.0               0.0               0             0.6 **              0
           47012      12 MM. TABLE'Z'         0.0               0.0               3             2.2 *               3
           47015      15 MM. TABLE'Z'         7.2               1.9              30            17.1 *              60
               0      18 MM. TABLE'Z'         0.0               0.0               0             0.0                 0
           47022      22 MM. TABLE'Z'         3.8               2.4              30            14.1 *              60
           47028      28 MM. TABLE'Z'         0.9               0.0              15             5.3 *              30
           47035      35 MM. TABLE'Z'         0.7               0.0              10            11.1                 20
           47042      42 MM. TABLE'Z'         0.3               0.0              10             8.5 *              20
           47054      54 MM. TABLE'Z'         0.2               0.0              10             7.0 *              20
           47076      76 MM. TABLE'Z'         0.1               0.0              10             2.1 *              15
           47108     108 MM. TABLE'Z'         0.1               0.0              10             2.2 *              15
           47133     133 MM. TABLE'Z'         0.0               0.0               3             1.1 *               3
           47159     159 MM. TABLE'Z'         0.0               0.0               3             2.2 *               3
               0                              0.0               0.0               0             0.0                 0
               0       6 MM. TABLE'Y'         0.0               0.0               0             0.0                 0
               0       8 MM. TABLE'Y'         0.0               0.0               1             0.0 *               2
               0      10 MM. TABLE'Y'         0.0               0.0               0             0.4 **              0
               0      12 MM. TABLE'Y'         0.0               0.0               2             0.3 *               3
           46015      15 MM. TABLE'Y'         0.0               0.0              15             5.5 *              25
               0      18 MM. TABLE'Y'         0.0               0.0               0             0.0                 0
           46022      22 MM. TABLE'Y'         0.0               0.0              20            14.0 *              30
           46028      28 MM. TABLE'Y'         0.0               0.0              10             4.7 *              15
           46035      35 MM. TABLE'Y'         0.0               0.0               5             3.9 *              10
           46042      42 MM. TABLE'Y'         0.0               0.0               5             3.3 *              10
           46054      54 MM. TABLE'Y'         0.0               0.0              15             1.6 *              20
           46076      76 MM. TABLE'Y'         0.0               0.0               5             0.0 *               5
           46108     108 MM. TABLE'Y'         0.0               0.0               2             0.0 *               2
               0     133 MM. TABLE'Y'         0.0               0.0               0             0.0                 0
               0     159 MM. TABLE'Y'         0.0               0.0               0             0.0                 0
               0                              0.0               0.0               0             0.0                 0
```

FIG 5.9



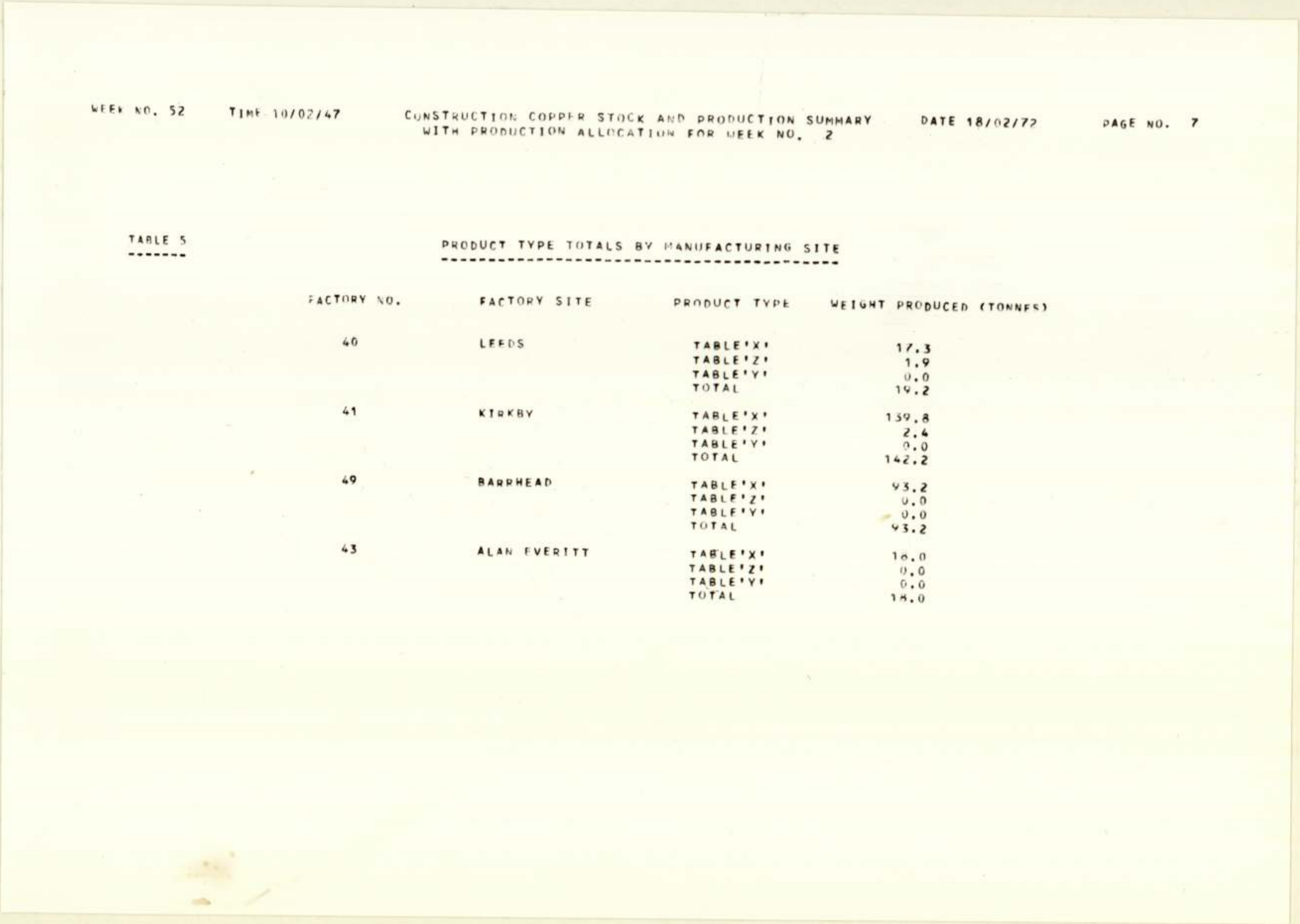
```
WEEK NO. 52     TIME 10/02/47       CONSTRUCTION COPPER STOCK AND PRODUCTION SUMMARY      DATE 18/02/72        PAGE NO.  2
                                      WITH PRODUCTION ALLOCATION FOR WEEK NO.  2




    TABLE 2                              PRODUCTION SUMMARY BY PRODUCT TYPE
    -------                              ----------------------------------

                                    PRODUCT TYPE              WEIGHT PRODUCED (TONNES)
                                    ------------              ------------------------

                                      TABLE'X'                          268.3
                                      TABLE'Z'                            4.3
                                      TABLE'Y'                            0.0
                                      TOTAL                             272.6




    TABLE 3                                 STOCK SUMMARY BY PRODUCT TYPE
    -------                                 -----------------------------

                                    PRODUCT TYPE                   STOCK (TONNES)
                                    ------------                   --------------

                                      TABLE'X'                          529.4
                                      TABLE'Z'                           71.8
                                      TABLE'Y'                           33.7
                                      TOTAL                             634.9
```

FIG 5.10



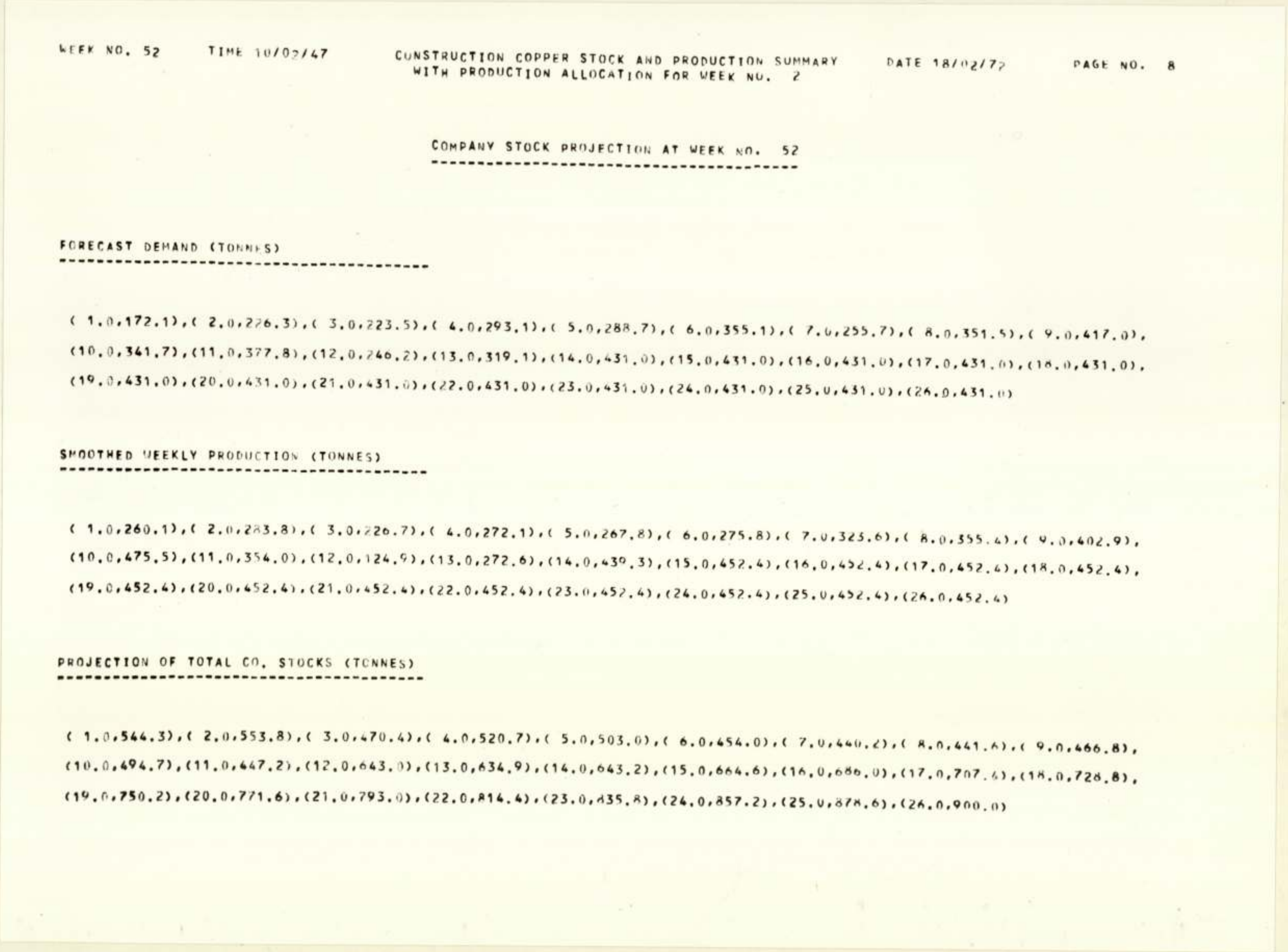
```
WEEK NO. 52     TIME 10/07/47     CONSTRUCTION COPPER STOCK AND PRODUCTION SUMMARY     DATE 18/02/72     PAGE NO.  3
                                   WITH PRODUCTION ALLOCATION FOR WEEK NO.  2

    TABLE 4                              PRODUCTION SUMMARY BY MANUFACTURING SITE
    -------                              ----------------------------------------

    (1) PRODUCTION AT FACTORY NO. 40 - LEEDS
    --------------------------------

                          CODE NO.          PRODUCT SIZE      WEIGHT PRODUCED (TONNES)
                                              AND TYPE

                           45006          6 MM. TABLE'X'            0.0
                           45008          8 MM. TABLE'X'            0.0
                               0         10 MM. TABLE'X'            0.0
                           45012         12 MM. TABLE'X'            0.0
                           45015         15 MM. TABLE'X'            7.4
                               0         18 MM. TABLE'X'            0.0
                           45022         22 MM. TABLE'X'            0.0
                           45028         28 MM. TABLE'X'            0.0
                           45035         35 MM. TABLE'X'            4.6
                           45042         42 MM. TABLE'X'            0.0
                           45054         54 MM. TABLE'X'            0.0
                           45076         76 MM. TABLE'X'            0.0
                           45108        108 MM. TABLE'X'            4.2
                           45133        133 MM. TABLE'X'            1.1
                           45159        159 MM. TABLE'X'            0.0
                               0                                    0.0
                           47006          6 MM. TABLE'Z'            0.0
                           47008          8 MM. TABLE'Z'            0.0
                               0         10 MM. TABLE'Z'            0.0
                           47012         12 MM. TABLE'Z'            0.0
                           47015         15 MM. TABLE'Z'            1.9
                               0         18 MM. TABLE'Z'            0.0
                           47022         22 MM. TABLE'Z'            0.0
                           47028         28 MM. TABLE'Z'            0.0
                           47035         35 MM. TABLE'Z'            0.0
                           47042         42 MM. TABLE'Z'            0.0
                           47054         54 MM. TABLE'Z'            0.0
                           47076         76 MM. TABLE'Z'            0.0
                           47108        108 MM. TABLE'Z'            0.0
                           47133        133 MM. TABLE'Z'            0.0
                           47159        159 MM. TABLE'Z'            0.0
                               0                                    0.0
                               0          6 MM. TABLE'Y'            0.0
                               0          8 MM. TABLE'Y'            0.0
                               0         10 MM. TABLE'Y'            0.0
                               0         12 MM. TABLE'Y'            0.0
                           46015         15 MM. TABLE'Y'            0.0
                               0         18 MM. TABLE'Y'            0.0
                           46022         22 MM. TABLE'Y'            0.0
                           46028         28 MM. TABLE'Y'            0.0
                           46035         35 MM. TABLE'Y'            0.0
                           46042         42 MM. TABLE'Y'            0.0
                           46054         54 MM. TABLE'Y'            0.0
                           46076         76 MM. TABLE'Y'            0.0
                           46108        108 MM. TABLE'Y'            0.0
                               0        133 MM. TABLE'Y'            0.0
                               0        159 MM. TABLE'Y'            0.0
                               0                                    0.0
```

FIG 5.11



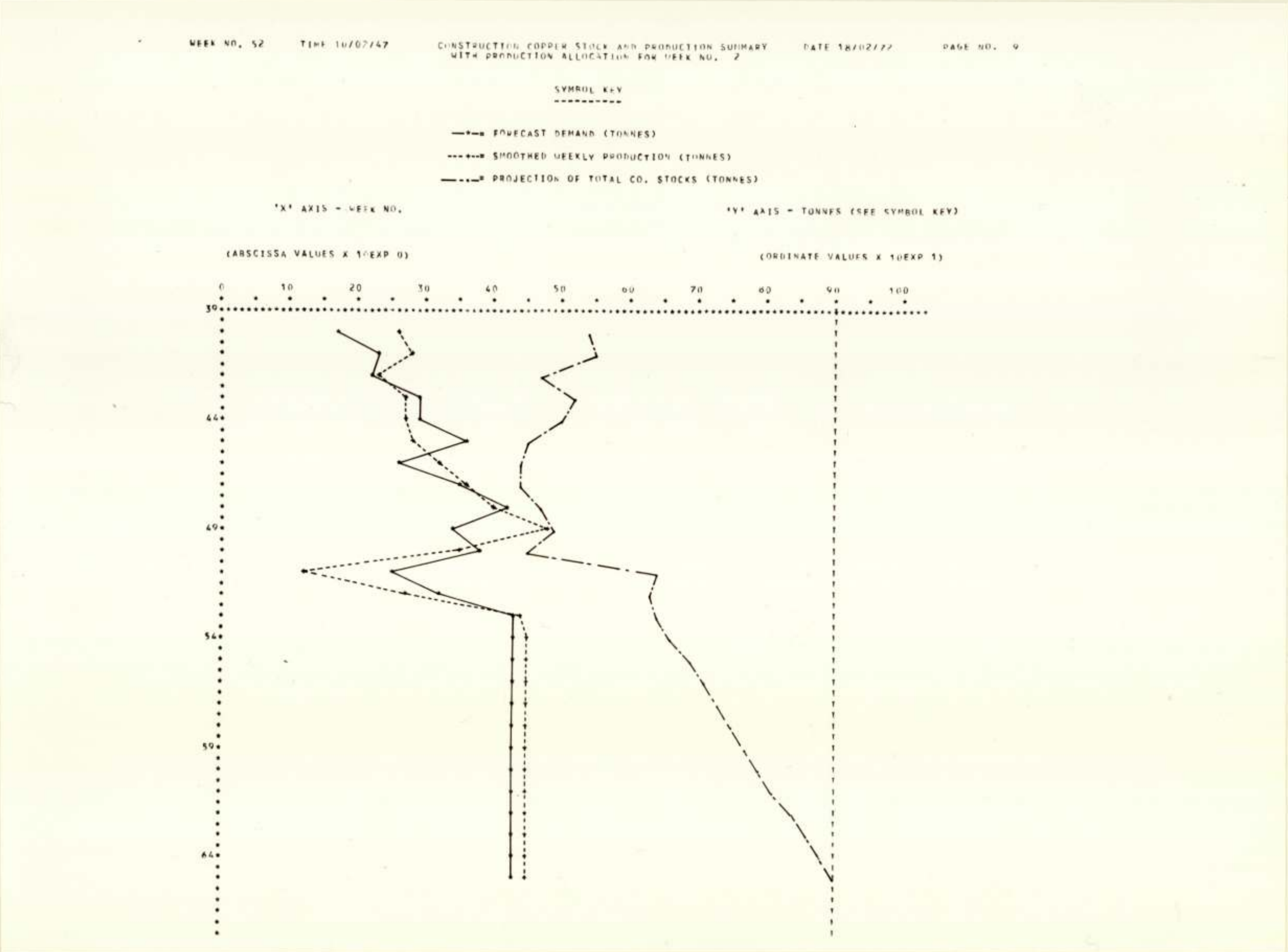
```
WEEK NO. 52     TIME 10/02/47      CONSTRUCTION COPPER STOCK AND PRODUCTION SUMMARY     DATE 18/02/72       PAGE NO.  7
                                    WITH PRODUCTION ALLOCATION FOR WEEK NO.  2



  TABLE 5                                PRODUCT TYPE TOTALS BY MANUFACTURING SITE
  -------                                -----------------------------------------

                FACTORY NO.        FACTORY SITE          PRODUCT TYPE    WEIGHT PRODUCED (TONNES)

                    40             LEEDS                   TABLE'X'              17.3
                                                           TABLE'Z'               1.9
                                                           TABLE'Y'               0.0
                                                           TOTAL                 19.2
                    41             KIRKBY                  TABLE'X'             139.8
                                                           TABLE'Z'               2.4
                                                           TABLE'Y'               0.0
                                                           TOTAL                142.2
                    49             BARRHEAD                TABLE'X'              93.2
                                                           TABLE'Z'               0.0
                                                           TABLE'Y'               0.0
                                                           TOTAL                 93.2
                    43             ALAN EVERITT            TABLE'X'              18.0
                                                           TABLE'Z'               0.0
                                                           TABLE'Y'               0.0
                                                           TOTAL                 18.0
```

FIG 5.12



```
WEEK NO. 52      TIME 10/02/47       CONSTRUCTION COPPER STOCK AND PRODUCTION SUMMARY      DATE 18/02/72        PAGE NO.  8
                                      WITH PRODUCTION ALLOCATION FOR WEEK NO.  2

                                          COMPANY STOCK PROJECTION AT WEEK NO.  52
                                          ----------------------------------------


FORECAST DEMAND (TONNES)
----------------------------------------

 ( 1.0,172.1),( 2.0,226.3),( 3.0,223.5),( 4.0,293.1),( 5.0,288.7),( 6.0,355.1),( 7.0,255.7),( 8.0,351.5),( 9.0,417.0),
 (10.0,341.7),(11.0,377.8),(12.0,246.2),(13.0,319.1),(14.0,431.0),(15.0,431.0),(16.0,431.0),(17.0,431.0),(18.0,431.0),
 (19.0,431.0),(20.0,431.0),(21.0,431.0),(22.0,431.0),(23.0,431.0),(24.0,431.0),(25.0,431.0),(26.0,431.0)


SMOOTHED WEEKLY PRODUCTION (TONNES)
----------------------------------------

 ( 1.0,260.1),( 2.0,283.8),( 3.0,226.7),( 4.0,272.1),( 5.0,267.8),( 6.0,275.8),( 7.0,325.6),( 8.0,355.4),( 9.0,402.9),
 (10.0,475.5),(11.0,354.0),(12.0,124.9),(13.0,272.6),(14.0,439.3),(15.0,452.4),(16.0,452.4),(17.0,452.4),(18.0,452.4),
 (19.0,452.4),(20.0,452.4),(21.0,452.4),(22.0,452.4),(23.0,452.4),(24.0,452.4),(25.0,452.4),(26.0,452.4)


PROJECTION OF TOTAL CO. STOCKS (TONNES)
----------------------------------------

 ( 1.0,544.3),( 2.0,553.8),( 3.0,470.4),( 4.0,520.7),( 5.0,503.0),( 6.0,454.0),( 7.0,440.2),( 8.0,441.6),( 9.0,466.8),
 (10.0,494.7),(11.0,447.2),(12.0,643.0),(13.0,634.9),(14.0,643.2),(15.0,664.6),(16.0,686.0),(17.0,707.4),(18.0,728.8),
 (19.0,750.2),(20.0,771.6),(21.0,793.0),(22.0,814.4),(23.0,835.8),(24.0,857.2),(25.0,878.6),(26.0,900.0)
```

FIG 5.13



WEEK NO. 52   TIME 10/07/47   CONSTRUCTION COPPER STOCK AND PRODUCTION SUMMARY   DATE 18/02/22   PAGE NO. 9
WITH PRODUCTION ALLOCATION FOR WEEK NO. 2

SYMBOL KEY

—*—* FORECAST DEMAND (TONNES)

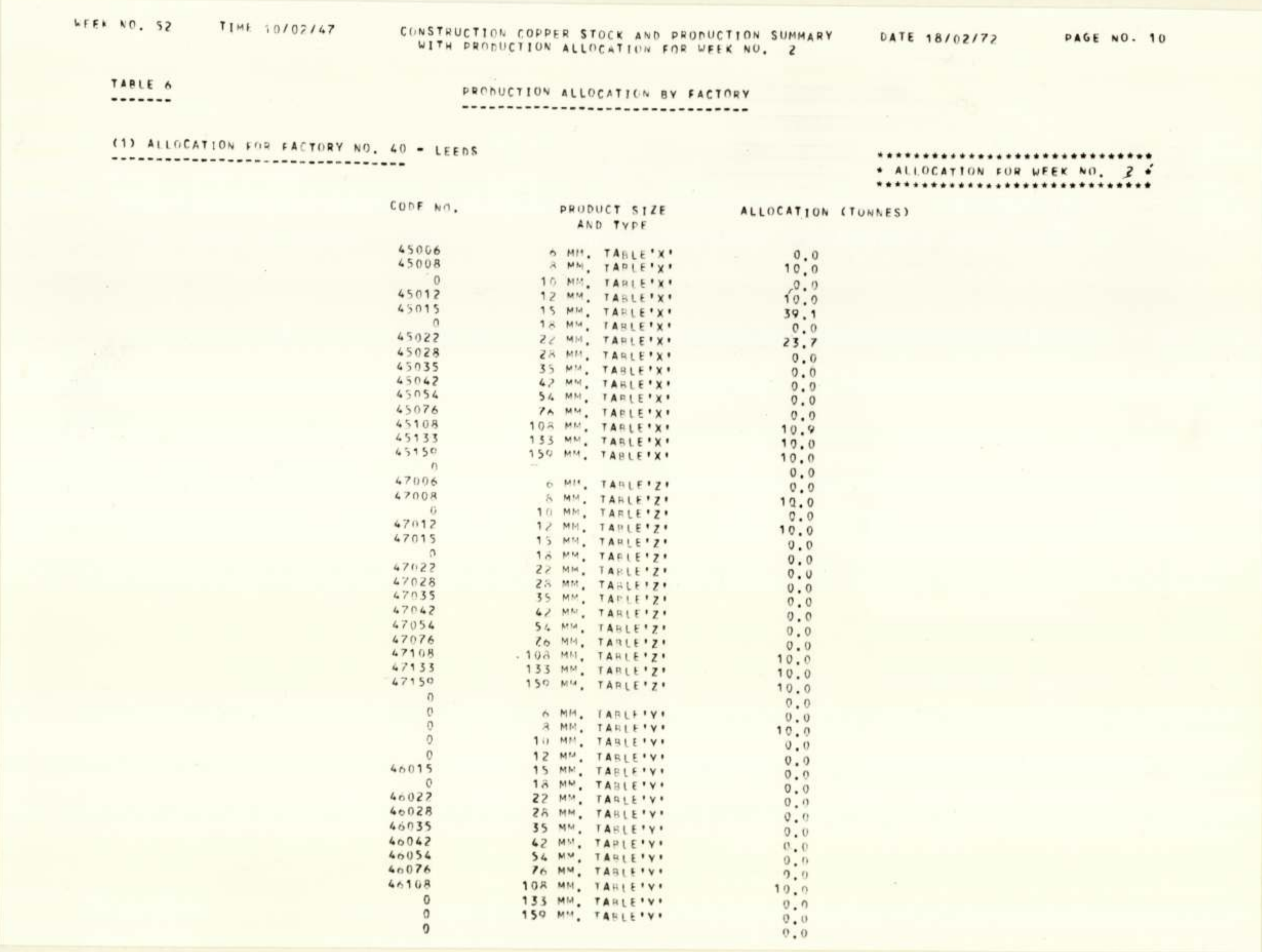
---*--* SMOOTHED WEEKLY PRODUCTION (TONNES)

—-·—* PROJECTION OF TOTAL CO. STOCKS (TONNES)

'X' AXIS - WEEK NO.   'Y' AXIS - TONNES (SEE SYMBOL KEY)

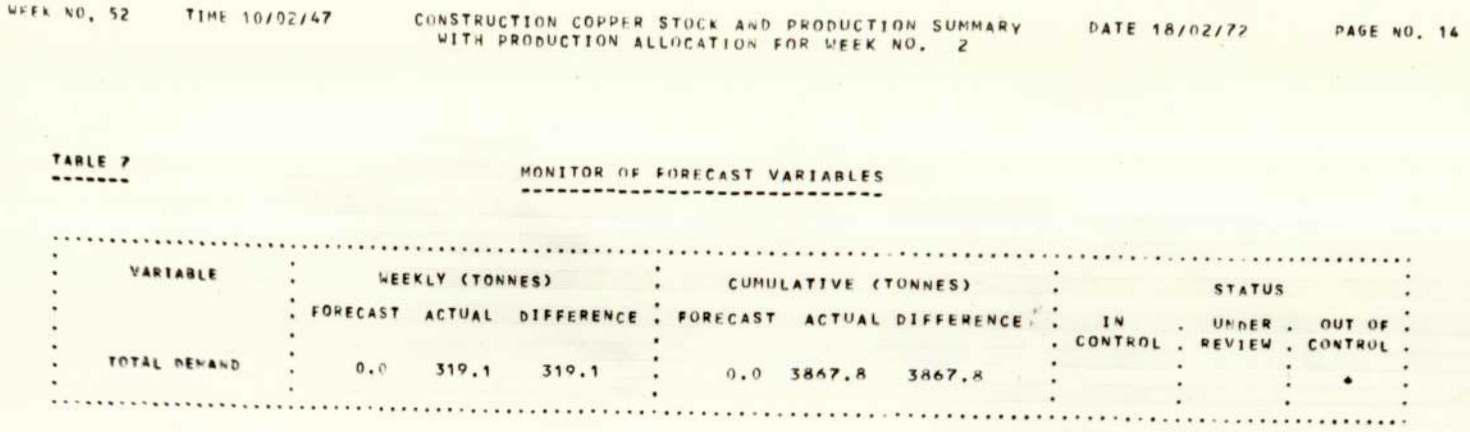
(ABSCISSA VALUES X 10EXP 0)   (ORDINATE VALUES X 10EXP 1)

FIG 5.14



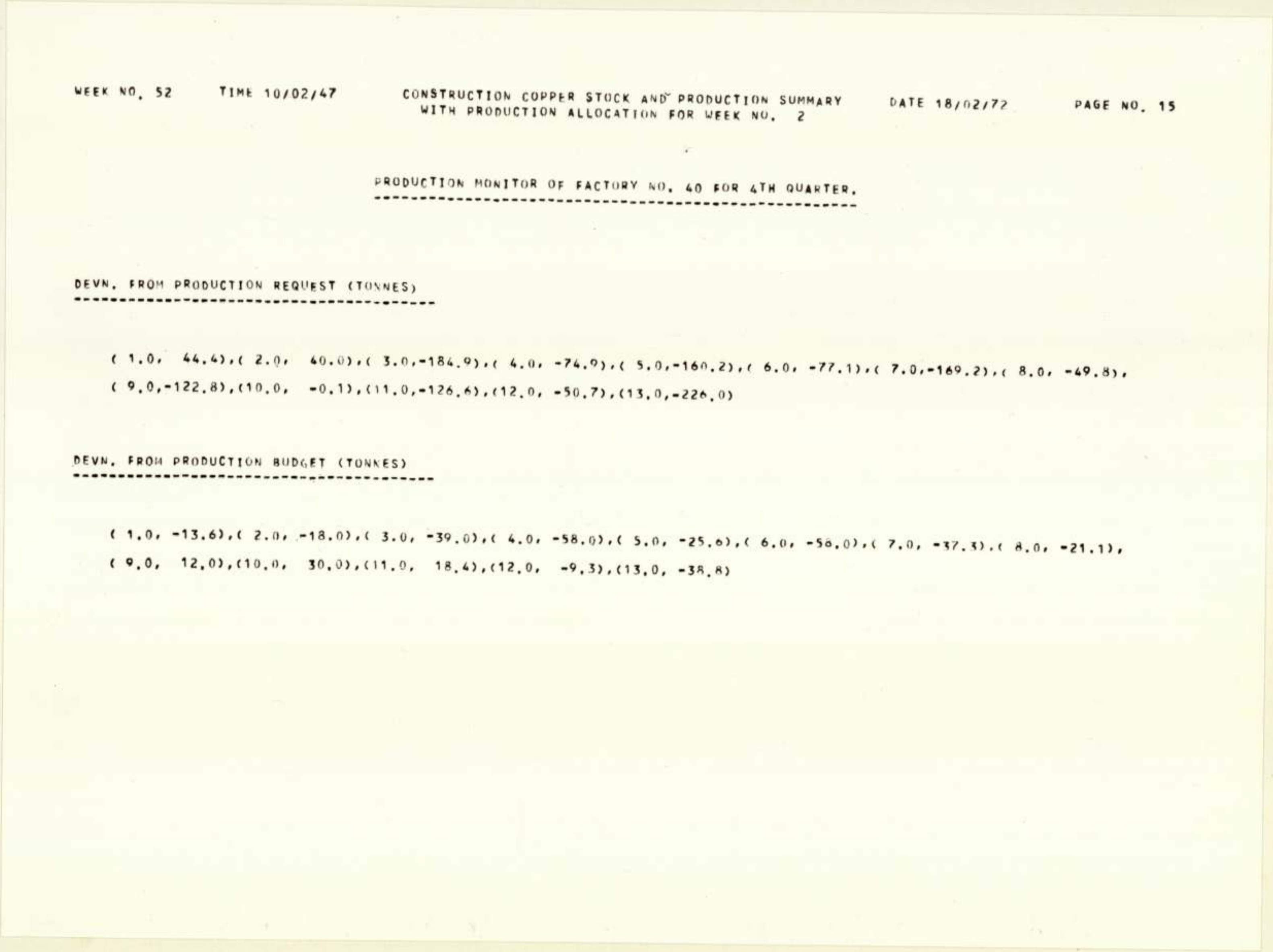
```
WEEK NO. 52     TIME 10/02/47     CONSTRUCTION COPPER STOCK AND PRODUCTION SUMMARY     DATE 18/02/72      PAGE NO. 10
                                   WITH PRODUCTION ALLOCATION FOR WEEK NO.  2

     TABLE 6                              PRODUCTION ALLOCATION BY FACTORY
     -------                              --------------------------------

     (1) ALLOCATION FOR FACTORY NO. 40 - LEEDS
     ---------------------------------                                         ******************************
                                                                               * ALLOCATION FOR WEEK NO.  2 *
                                                                               ******************************
                          CODE NO.           PRODUCT SIZE          ALLOCATION (TONNES)
                                               AND TYPE

                          45006            6 MM. TABLE'X'             0.0
                          45008            8 MM. TABLE'X'            10.0
                              0           10 MM. TABLE'X'             0.0
                          45012           12 MM. TABLE'X'            10.0
                          45015           15 MM. TABLE'X'            39.1
                              0           18 MM. TABLE'X'             0.0
                          45022           22 MM. TABLE'X'            23.7
                          45028           28 MM. TABLE'X'             0.0
                          45035           35 MM. TABLE'X'             0.0
                          45042           42 MM. TABLE'X'             0.0
                          45054           54 MM. TABLE'X'             0.0
                          45076           76 MM. TABLE'X'             0.0
                          45108          108 MM. TABLE'X'            10.9
                          45133          133 MM. TABLE'X'            10.0
                          45159          159 MM. TABLE'X'            10.0
                              0                                       0.0
                          47006            6 MM. TABLE'Z'             0.0
                          47008            8 MM. TABLE'Z'            10.0
                              0           10 MM. TABLE'Z'             0.0
                          47012           12 MM. TABLE'Z'            10.0
                          47015           15 MM. TABLE'Z'             0.0
                              0           18 MM. TABLE'Z'             0.0
                          47022           22 MM. TABLE'Z'             0.0
                          47028           28 MM. TABLE'Z'             0.0
                          47035           35 MM. TABLE'Z'             0.0
                          47042           42 MM. TABLE'Z'             0.0
                          47054           54 MM. TABLE'Z'             0.0
                          47076           76 MM. TABLE'Z'             0.0
                          47108          108 MM. TABLE'Z'            10.0
                          47133          133 MM. TABLE'Z'            10.0
                          47159          159 MM. TABLE'Z'            10.0
                              0                                       0.0
                              0            6 MM. TABLE'Y'             0.0
                              0            8 MM. TABLE'Y'            10.0
                              0           10 MM. TABLE'Y'             0.0
                              0           12 MM. TABLE'Y'             0.0
                          46015           15 MM. TABLE'Y'             0.0
                              0           18 MM. TABLE'Y'             0.0
                          46022           22 MM. TABLE'Y'             0.0
                          46028           28 MM. TABLE'Y'             0.0
                          46035           35 MM. TABLE'Y'             0.0
                          46042           42 MM. TABLE'Y'             0.0
                          46054           54 MM. TABLE'Y'             0.0
                          46076           76 MM. TABLE'Y'             0.0
                          46108          108 MM. TABLE'Y'            10.0
                              0          133 MM. TABLE'Y'             0.0
                              0          159 MM. TABLE'Y'             0.0
                              0                                       0.0
```

FIG 5.15



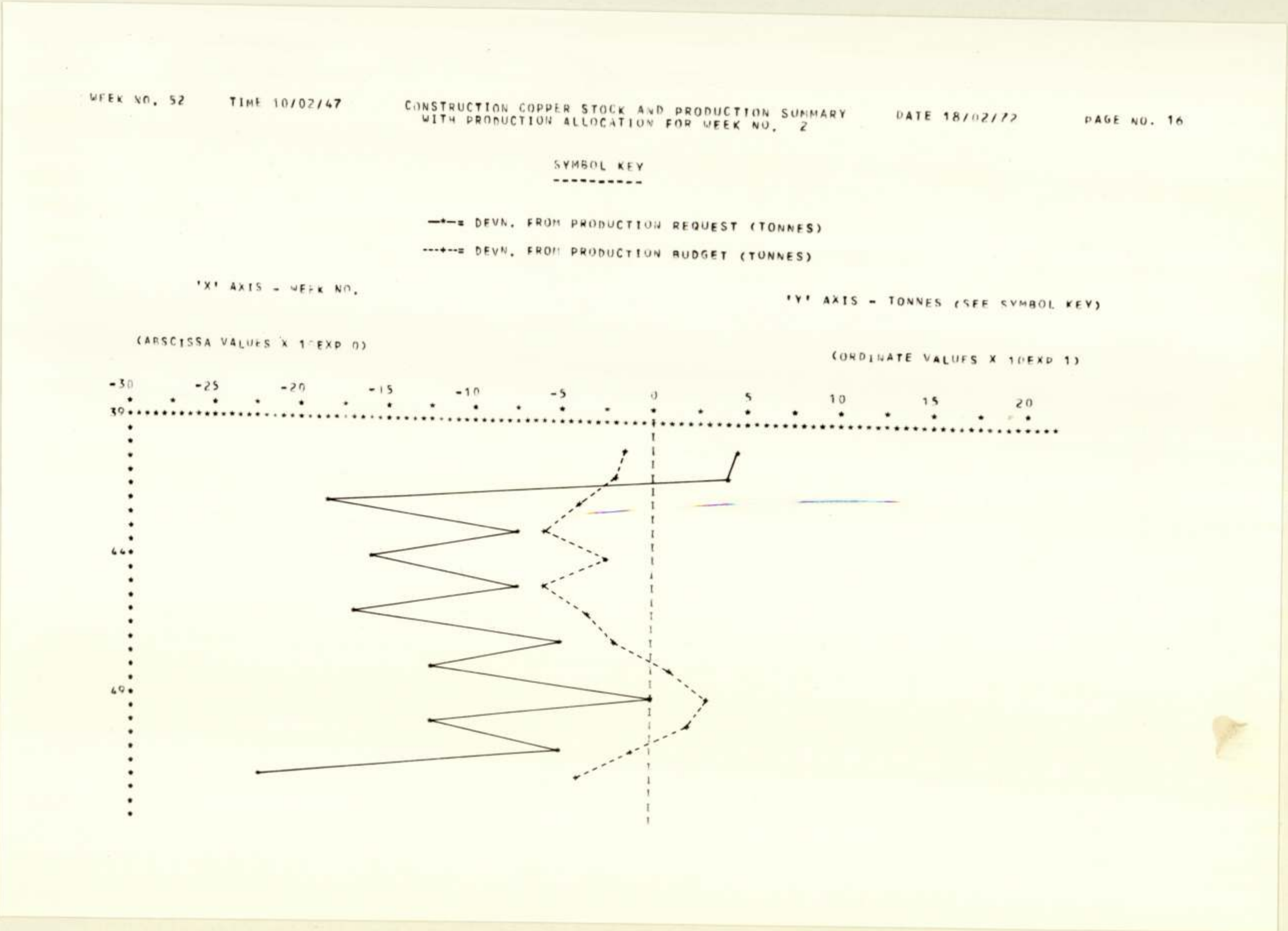
```
WEEK NO. 52     TIME 10/02/47       CONSTRUCTION COPPER STOCK AND PRODUCTION SUMMARY      DATE 18/02/72        PAGE NO. 14
                                     WITH PRODUCTION ALLOCATION FOR WEEK NO.  2


    TABLE 7
    -------                                     MONITOR OF FORECAST VARIABLES
                                                -----------------------------

    ...........................................................................................................................
    .      VARIABLE      .        WEEKLY (TONNES)          .        CUMULATIVE (TONNES)          .              STATUS           .
    .                    . FORECAST  ACTUAL  DIFFERENCE    . FORECAST   ACTUAL  DIFFERENCE       .    IN     .  UNDER  .  OUT OF .
    .                    .                                 .                                     .  CONTROL  .  REVIEW .  CONTROL.
    .    TOTAL DEMAND    .    0.0     319.1    319.1       .     0.0   3867.8    3867.8          .           .         .    *    .
    .                    .                                 .                                     .           .         .         .
    ...........................................................................................................................
```

FIG 5.16



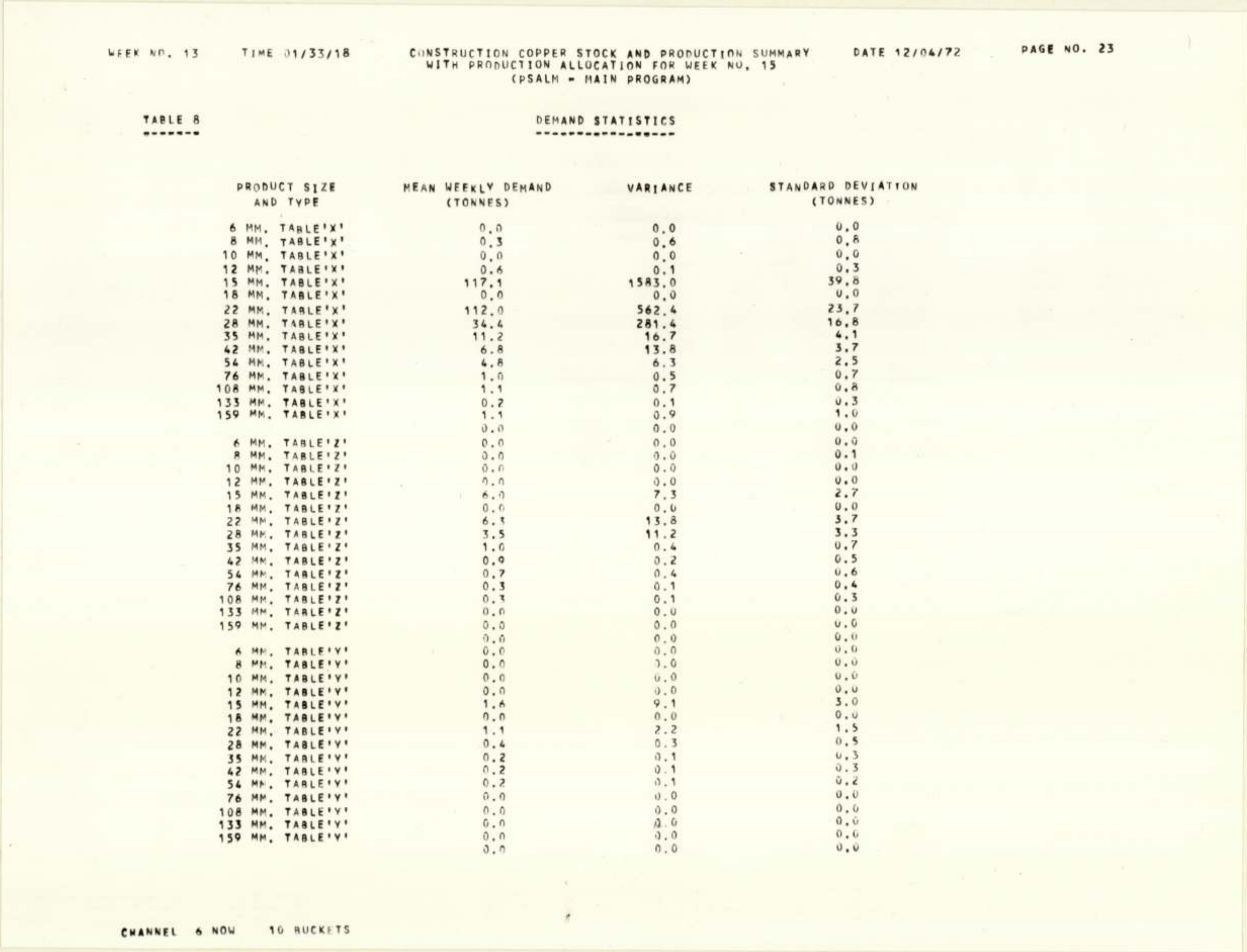
```
WEEK NO. 52      TIME 10/02/47       CONSTRUCTION COPPER STOCK AND PRODUCTION SUMMARY      DATE 18/02/72      PAGE NO. 15
                                        WITH PRODUCTION ALLOCATION FOR WEEK NO.  2


                               PRODUCTION MONITOR OF FACTORY NO. 40 FOR 4TH QUARTER.
                               -----------------------------------------------------


DEVN. FROM PRODUCTION REQUEST (TONNES)
--------------------------------------

    ( 1.0,  44.4),( 2.0,  40.0),( 3.0,-184.9),( 4.0, -74.9),( 5.0,-160.2),( 6.0, -77.1),( 7.0,-169.2),( 8.0, -49.8),
    ( 9.0,-122.8),(10.0,  -0.1),(11.0,-126.6),(12.0, -50.7),(13.0,-226.0)


DEVN. FROM PRODUCTION BUDGET (TONNES)
-------------------------------------

    ( 1.0, -13.6),( 2.0, -18.0),( 3.0, -39.0),( 4.0, -58.0),( 5.0, -25.6),( 6.0, -56.0),( 7.0, -37.3),( 8.0, -21.1),
    ( 9.0,  12.0),(10.0,  30.0),(11.0,  18.4),(12.0,  -9.3),(13.0, -38.8)
```

FIG 5.17



```
WEEK NO. 52     TIME 10/02/47     CONSTRUCTION COPPER STOCK AND PRODUCTION SUMMARY     DATE 18/02/72     PAGE NO. 16
                                   WITH PRODUCTION ALLOCATION FOR WEEK NO.  2

                                              SYMBOL KEY
                                              ----------

                              —*—= DEVN. FROM PRODUCTION REQUEST (TONNES)
                              ---+--= DEVN. FROM PRODUCTION BUDGET (TONNES)

        'X' AXIS - WEEK NO.
                                                                  'Y' AXIS - TONNES (SEE SYMBOL KEY)

    (ABSCISSA VALUES X 10EXP 0)
                                                                  (ORDINATE VALUES X 10EXP 1)
```

FIG 5.18



```
WEEK NO. 13    TIME 01/33/18    CONSTRUCTION COPPER STOCK AND PRODUCTION SUMMARY    DATE 12/04/72    PAGE NO. 23
                                  WITH PRODUCTION ALLOCATION FOR WEEK NO. 15
                                          (PSALM - MAIN PROGRAM)

   TABLE 8                                       DEMAND STATISTICS
   -------                                       -----------------


          PRODUCT SIZE        MEAN WEEKLY DEMAND        VARIANCE        STANDARD DEVIATION
            AND TYPE               (TONNES)                                  (TONNES)

         6 MM. TABLE'X'              0.0                   0.0                 0.0
         8 MM. TABLE'X'              0.3                   0.6                 0.8
        10 MM. TABLE'X'              0.0                   0.0                 0.0
        12 MM. TABLE'X'              0.6                   0.1                 0.3
        15 MM. TABLE'X'            117.1                1583.0                39.8
        18 MM. TABLE'X'              0.0                   0.0                 0.0
        22 MM. TABLE'X'            112.0                 562.4                23.7
        28 MM. TABLE'X'             34.4                 281.4                16.8
        35 MM. TABLE'X'             11.2                  16.7                 4.1
        42 MM. TABLE'X'              6.8                  13.8                 3.7
        54 MM. TABLE'X'              4.8                   6.3                 2.5
        76 MM. TABLE'X'              1.0                   0.5                 0.7
       108 MM. TABLE'X'              1.1                   0.7                 0.8
       133 MM. TABLE'X'              0.2                   0.1                 0.3
       159 MM. TABLE'X'              1.1                   0.9                 1.0
                                     0.0                   0.0                 0.0
         6 MM. TABLE'Z'              0.0                   0.0                 0.0
         8 MM. TABLE'Z'              0.0                   0.0                 0.1
        10 MM. TABLE'Z'              0.0                   0.0                 0.0
        12 MM. TABLE'Z'              0.0                   0.0                 0.0
        15 MM. TABLE'Z'              6.0                   7.3                 2.7
        18 MM. TABLE'Z'              0.0                   0.0                 0.0
        22 MM. TABLE'Z'              6.3                  13.8                 3.7
        28 MM. TABLE'Z'              3.5                  11.2                 3.3
        35 MM. TABLE'Z'              1.0                   0.4                 0.7
        42 MM. TABLE'Z'              0.9                   0.2                 0.5
        54 MM. TABLE'Z'              0.7                   0.4                 0.6
        76 MM. TABLE'Z'              0.3                   0.1                 0.4
       108 MM. TABLE'Z'              0.3                   0.1                 0.3
       133 MM. TABLE'Z'              0.0                   0.0                 0.0
       159 MM. TABLE'Z'              0.0                   0.0                 0.0
                                     0.0                   0.0                 0.0
         6 MM. TABLE'Y'              0.0                   0.0                 0.0
         8 MM. TABLE'Y'              0.0                   0.0                 0.0
        10 MM. TABLE'Y'              0.0                   0.0                 0.0
        12 MM. TABLE'Y'              0.0                   0.0                 0.0
        15 MM. TABLE'Y'              1.6                   9.1                 3.0
        18 MM. TABLE'Y'              0.0                   0.0                 0.0
        22 MM. TABLE'Y'              1.1                   2.2                 1.5
        28 MM. TABLE'Y'              0.4                   0.3                 0.5
        35 MM. TABLE'Y'              0.2                   0.1                 0.3
        42 MM. TABLE'Y'              0.2                   0.1                 0.3
        54 MM. TABLE'Y'              0.2                   0.1                 0.2
        76 MM. TABLE'Y'              0.0                   0.0                 0.0
       108 MM. TABLE'Y'              0.0                   0.0                 0.0
       133 MM. TABLE'Y'              0.0                   0.0                 0.0
       159 MM. TABLE'Y'              0.0                   0.0                 0.0
                                     0.0                   0.0                 0.0


   CHANNEL  6 NOW     10 BUCKETS
```

FIG 5.19

were available although it was considered essential to the system that data could be displayed in graphical form. To this end a software package GOODS was prepared which enabled point plots to be prepared using the lineprinter. (Details of the GOODS program are given in Appendix M3). Point plots are not particularly useful for distinguishing trends but once the data is in this form, the plotted points can readily be connected by pen. By enabling data co-ordinates to be plotted automatically against flexible scales the task of graph preparation was thus reduced to a simple clerical task.

Table 8 of the PSALM output - fig. (5.19) - is based upon the first quarter of 1972. This particular facility was added after the trial period and nominally after the research had ended. Actually the system support manuals (Appendices M1 and M2) were prepared during the first months of 1972 and arrangements were made for C.A.D. staff to operate the system at this time. Special coding sheets were also prepared containing the pre-coded card identification codes (Appendix M1).

During the development and testing of the PSALM system there was close collaboration with the C.A.D. manager. As a result of this collaboration a number of changes were made to the initial system.

Thus it was agreed that stock would be regarded as an input to the system rather than as a variable derived from production and demand figures. The reason for this was discussed in 2.2.3 (iii).

Secondly the format of the tables output by PSALM were changed to include product codes and factory codes at the request of the C.A.D. manager. He would have been satisfied to lose the alphameric descriptions as he was completely familiar with the product codes. Had this practice been adopted it would have been in line with many other computer generated listings used at the Company. However experience had shown that the value of such listings was limited to a very small number of people familiar with the codes used and consequently the printouts were virtually incomprehensible to anyone outside the specialist area concerned. It was agreed that the alphameric description and the codes should be used. Use of product codes would help in the production of information for use by other programs in future system development.

It was also agreed to impose a limit on the variation the program could apply to the smoothed production level, even though this would mean that the buffer stock would not be produced by the end of the horizon period. At the time of the trial period, stocks were approximately half their "minimum" level, for reasons discussed in the next section. This meant that PSALM was calculating smoothed production levels which were well in excess of those set out in the production budgets and which could not be met by production. To avoid this situation the program was amended after the trial period so that the smoothed production level could never fall below 85% of the total production budget and could never exceed 125% of the budget.

Because of the stock shortage the minimum overall stock level for the company was entered in the model as 900 tons at the suggestion of the C.A.D. manager. The minimum overall stock level corresponding to the minimum values shown in Table 1 of the PSALM output - see Table (4.5) or (4.6) for individual values - was 1286 tons. This also helped to keep the smoothed production level within reasonable bounds.

Finally the C.A.D. manager asked that orders to bring stock levels up to their minimum should not be less than 10.0 tons. This would prevent uneconomic production orders of, say, 0.1 tons being raised by the system.

5.4 TRIAL RESULTS & DISCUSSION

The overall impression gained from the results of running the system on the results of the 4th quarter of 1971 was one of considerable confusion, almost chaos, in the production/stock control situation.

Undoubtedly a great deal of this was due to the fact that a metric range of products had just been introduced and it was the intention of the Company to discontinue its imperial size products. The changeover period occurred during the period of the trial results and thus production was initially divided between both product ranges and in addition stocks of the metric size product still had to be built up.

There was no way of anticipating how rapidly customers would change to the new product range given that imperial size stocks were still available at builders' merchants and other outlets. Since the two product ranges were approximately equivalent, it might seem at first sight that at least the total demand for metric and imperial sizes would be the same as if only the imperial size product range were available. If this were the case then the total forecast of demand could be obtained using the relationship derived by Marketing Department which was described in 4.1. Unfortunately even this approximation was unlikely to be adequate as the wholesale trade were expected to place abnormally large orders in order to build up their own stocks. Thus individual estimates of expected demand varied within the Company and further complicated the already uncertain picture.

Bearing the above in mind, one of the immediately apparent observations to be made from the PSALM print-outs was that stocks of all items in each of the Table 'X', 'Y' & 'Z' product ranges were below the official minima for the whole 13 week period.

Table (5.2) shows the production budgets for the trial quarter. One of the immediate problems of implementing this within the PSALM framework was that budgets were not specifically given for all products. Thus it can be seen in Table (5.2) that the official production budget contains entries such as "others" and "over 54 mm", while the PSALM system operates at a more detailed level of control. However it was decided

Table (5.2) PRODUCTION BUDGETS FOR FINAL QUARTER 1971
(TABLES X, Y &Z)

Factory No.	Product Type	Size (mm)	Budget (tons)
1	BS 659/Table X	(15 (22 (others	33 20 5
	BS 3931/Table Z		7
	BS 1386/Table Y		2
2	BS 659/Table X	15 22 28 35 42 54 Over 54	110 20 40 10 10 10 4
	BS 3931/Table Z		18
	BS 1386/Table Y		25
3	BS 659/Table X	15 22	10 85
4	BS 659/Table X	15	25

that the specific allocations obtained from the system were not too important as it was recognised that the principle of weekly adjustments to the production budget would not be implemented immediately. It was decided therefore merely to enter those production budgets specifically made for individual products. This would account for 373 tons of the weekly budget of Table 'X', leaving a further 9 tons not specifically allocated. The 27 tons of Table 'Y' and the 25 tons of Table 'Z' were ignored. Instead of allocating this 9 tons in an arbitrary manner it was expedient to enter the total weekly budget as (373 + 9 = 482) on the datafile. This had the effect of spreading the 9 tons in proportion across the products already allocated. (Further details may be found in Appendix M1, section 5.1.2).

A more fundamental problem was that in the production budget no differentiation between Table 'A' and Table 'X' was made. It was necessary to assume therefore that the whole of the budget would be Table 'X'. Hopefully this largely explains the deviations from production budget recorded in the quarter which are presented in fig. (5.20). In fact the overall picture is one of production deficit and this is reflected in the fall in stock over the period-fig. (5.14).

Apart from the general production deficit, the variation of weekly production achieved is shown in fig. (5.20). The difficulty of controlling stocks is, of course, as much related the variability of production as to the variability of demand. However it should be possible to control the variability of production to a greater degree than that observed during the trial period. Some reasons for the inability of C.A.D. to control production levels have been discussed in 2.2.2.

Finally it is interesting to note that all the curves in fig. (5.20) have approximately the same shape. This may be caused by the fact that

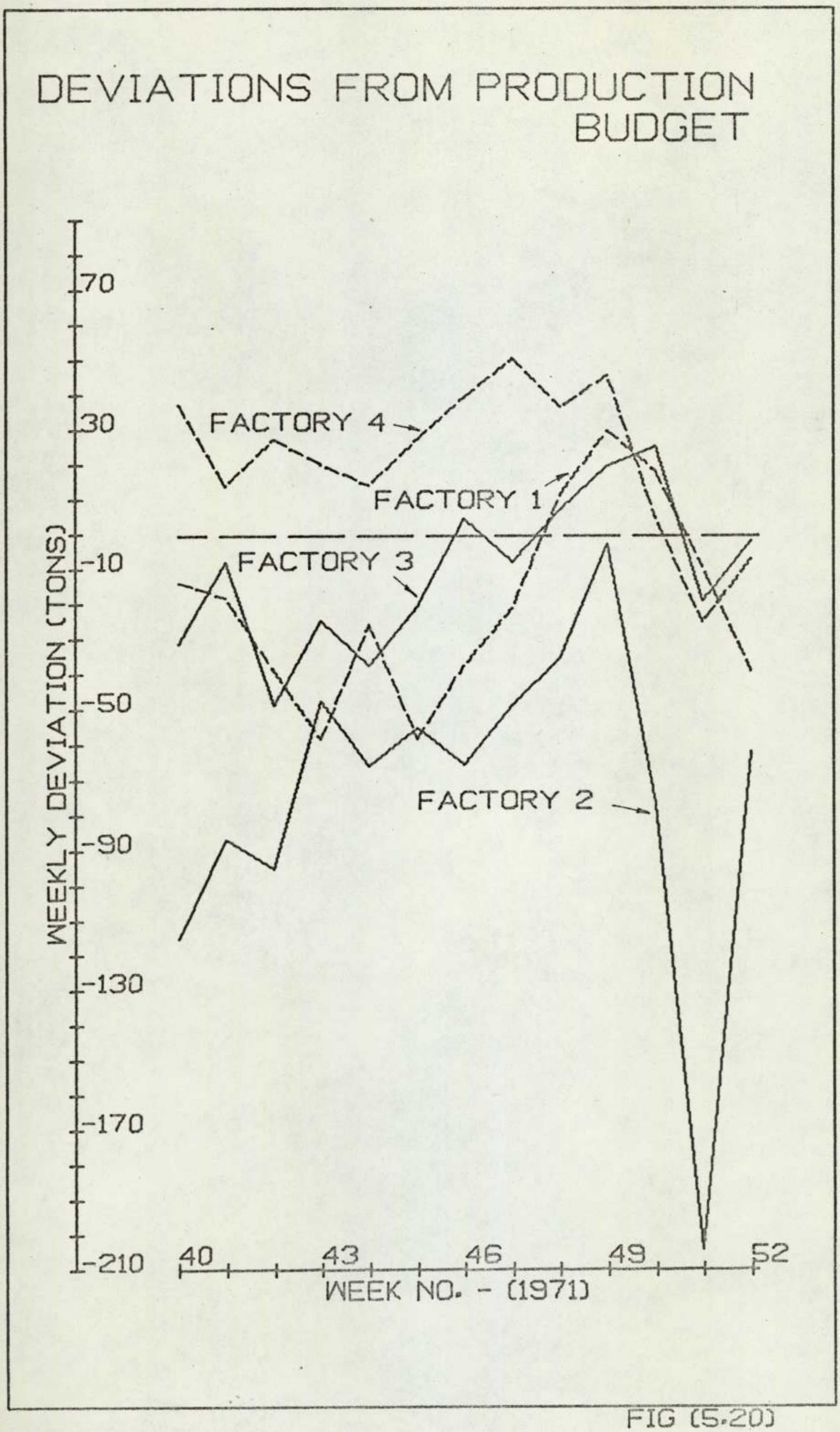

DEVIATIONS FROM PRODUCTION BUDGET
WEEKLY DEVIATION (TONS)
70
30
-10
-50
-90
-130
-170
-210
FACTORY 4
FACTORY 1
FACTORY 3
FACTORY 2
40
43
46
49
52
WEEK NO. - (1971)

FIG (5.20)

factory 2 produces most of the starting stock for products in the Table 'X', 'Y' and 'Z' product groups. If so it indicates where further investigation would be profitable, since the suggestion is that the deviations from production budgets originate in the common production process at Factory 2. The difficulties of stock control are underlined by the fact that the factory with the largest production budget also has the greatest deviations from that budget.

Figure (5.21) shows the production requests (allocations) that PSALM has generated. Since no attempt was made to implement these, they merely serve to show how the system would work assuming that production departments could and would respond to weekly monitoring by C.A.D. The greater magnitude of the deviations in fig. (5.21) compared with those of fig. (5.20) is caused by there being no limit imposed to the smoothed production level during the period of the trial. Thus as the end of the period approached the smoothed production level rose sharply as the system tried to achieve the minimum stock level by the end of the period. (See Appendix M1, section 2.2.1). This effect appears as increasing deviations from production request in fig. (5.21) in weeks 46, 48 and 50 for the major producer Factory 2, and to a lesser extent a similar effect is noticeable for the next largest producer, Factory 1. The oscillatory effect in the deviations is caused by the system requesting lesser amounts of production in alternate weeks in anticipation of stocks becoming available as a result of the previous week's production request. In the first week of the trial period there was no production request and the deviation shown is thus the actual production achieved. In the second week the production request was set equal to the material programmed for completion in that week - see for example, Table (4.1). In the third and subsequent weeks the deviations are those measured by the system.

Figure (5.22) reconstructs a printout obtained from running the PSALM system on Table 'A' data at week number 40 in 1971, prior to

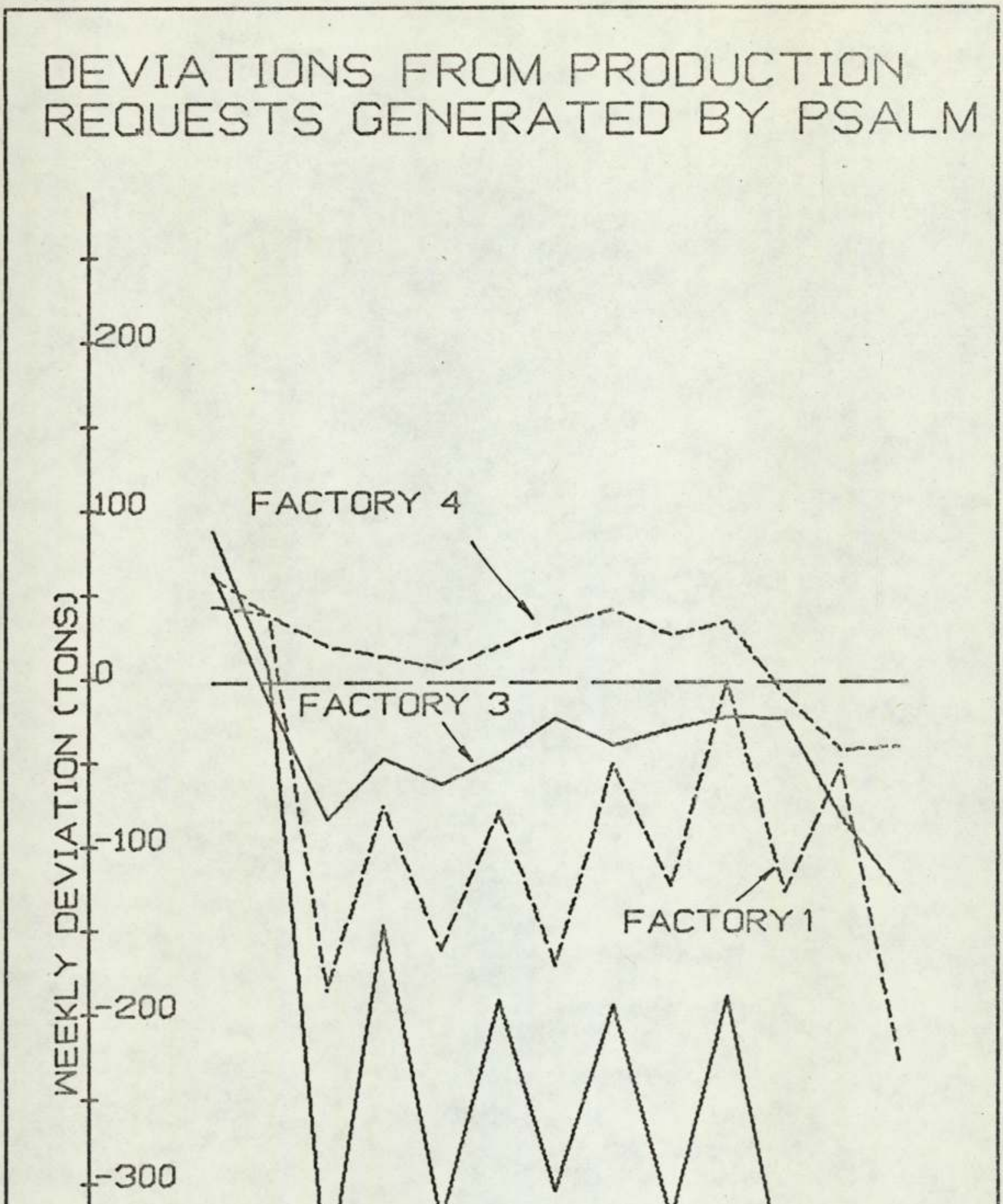

DEVIATIONS FROM PRODUCTION
REQUESTS GENERATED BY PSALM
WEEKLY DEVIATION (TONS)
200
100
0
-100
-200
-300
-400
-500
FACTORY 4
FACTORY 3
FACTORY 1
FACTORY 2
40
43
46
49
52
WEEK NO. - (1971)

FIG (5.21)

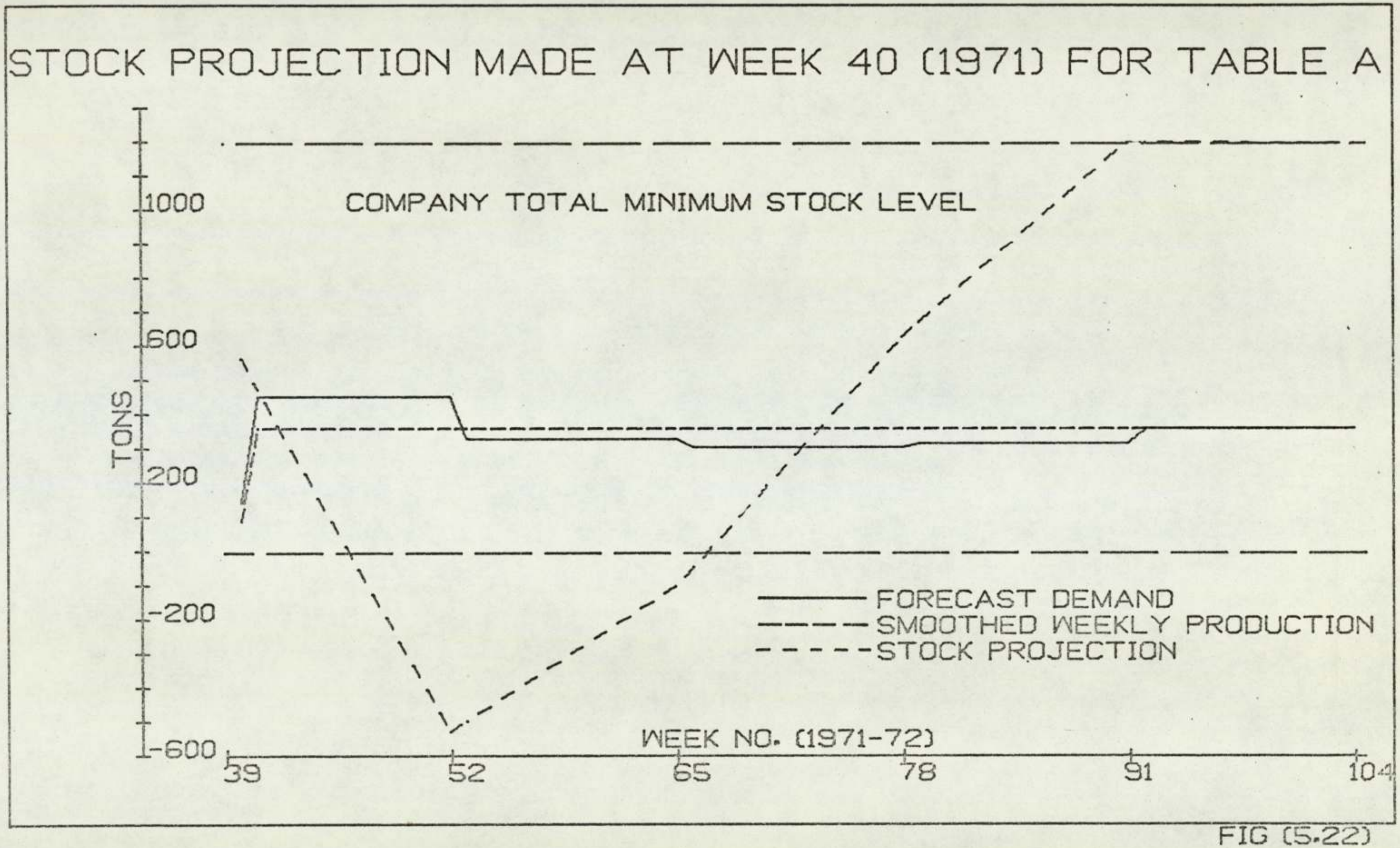

STOCK PROJECTION MADE AT WEEK 40 (1971) FOR TABLE A
COMPANY TOTAL MINIMUM STOCK LEVEL
TONS
1000
600
200
-200
-600
WEEK NO. (1971-72)
39
52
65
78
91
104
FORECAST DEMAND
SMOOTHED WEEKLY PRODUCTION
STOCK PROJECTION

FIG (5.22)

conversion to metric data. It shows that a short term deficit in stock is expected to occur by the end of the year if the shown levels of production and forecast demand are maintained. In fact such a shortage did occur, but the accuracy of prediction is clouded by the complicating factors of the effects of metrication already discussed. However fig. (5.22) does serve to show how the system should be used to predict the medium term future (the horizon period is only limited by the period of the forecast, of course) and exemplifies with real data the schematic presentations of 5.2.

Fig. (5.14) shows the situation at the end of the trial period with the stock projection for the first quarter of 1972. It may not be compared with fig. (5.22), of course, because fig. (5.14) refers to metric products. It is suggested however that if PSALM had been implemented <u>before</u> metrication had commenced, two versions could have run side by side, one for metric products and one for imperial sizes. This could have been a great help to monitoring the progress of metrication and evaluating the effects of various levels of future demand.

The difficulty of obtaining any reliable forecasts of demand led to the use of the cusum analysis facility being temporarily shelved.

6. REVIEW

This chapter begins by summarising the essential differences between the manual and computer systems for production smoothing and allocation (6.1). Although Part 1 has concentrated on the functional aspects of production control, much of the work was carried out against a background of considerable organisational change and inevitably this had an effect on the work carried out. Some of these factors are mentioned in section 6.2. Finally 6.3 outlines a programme of future work that was submitted for consideration. This programme would explicitly take into account more variables than the PSALM solution. Additionally an optimising solution to allocation is described and its validity is discussed.

6.1 COMPARISON OF MANUAL AND COMPUTER SYSTEMS

The manual system was rudimentary, largely unelaborated and depended considerably on the skill of the allocator. Production allocation and stock reviews took place weekly and were based on the information shown in Table (4.1). Such a table was prepared weekly for the allocator who would use the information contained therein to make his allocations. If necessary, reference could be made to past tables. No formal attempt was made to monitor the variation of demand and production through time by extracting trends from the data. A global view was taken of stock and production budgets were only partially differentiated with respect to individual products.

On the other hand the PSALM system provides a consistent, logical and yet simple means of reviewing and adjusting production levels in accordance with specific Company policies. It requires only the same input information as the manual system to provide everything the manual system provides. Additionally it automatically scans stock levels indicating excesses and deficiencies of stock and raises replenishment orders where necessary. Information about the three product groups used for production smoothing is presented on the same sheet and stock and production summaries are given by product type and manufacturing site. Instead of providing only point estimates of the major variables (ie stock, production and demand), PSALM reproduces past trends and a simple projection of future stocks is made. The future trends may be manipulated to allow for expected events such as annual shut-down of sites for maintainance and the need to create stock to cover holiday periods when production levels are low. Allocations of production to manufacturing sites are adjusted automatically to allow for unexpected depletions of stock and provision has been made to monitor forecasts of demand, production budgets and/or any other variable whose expected value is known. Finally a record of deviations of production from the budget enables responsibility for stock deficiences and excesses to be allocated according to performance. The occupation time for one run of PSALM was approximately 7-8 mins on the Company's I.C.L. 1903A (32k) computer.

In addition to providing a more systematic approach to allocation, important statistics are accrued describing the mean demand and its variance for each of the 45 products individually monitored. Primarily intended for re-setting the limits in the cusum analysis, this information is useful in assessing changes in demand in the absence of a better forecasting system. Also of great importance is the fact that the data input weekly on cards to the system is in an ideal form for additional analyses of the type reported in Chapter 4. Before the advent of PSALM this data had to be extracted manually from sheets such as that shown in Table (4.1). This is a very time consuming exercise and as such constitutes an effective deterrent to such analyses being carried out.

Adopting the PSALM system represents an important advance in the control of allocation. However the system may be improved further by (a) gaining a firmer grasp on production and so reducing a source of variation potentially under internal control and (b) to improve both the reliability of the forecasts of total demand and demand for individual products. When the foregoing has been achieved the door is open to more ambitious improvements in allocation procedures.

6.2 EXTRANEOUS FACTORS

One extraneous factor that significantly affected the research was metrication, but this has already been discussed in 5.3 and 5.4. The remaining factors were concerned with organisational aspects.

During the course of the study the position of allocator was occupied by three separate people. Each had rather different views of the job and different requirements of the new allocation system, which introduced additional design requirements in order that the system could retain its utility for the allocator.

Finally, during the test period, a major re-organisation was being planned to reconstitute the Company on a product division basis. Previously product division boundaries were relatively flexible, for example, Fittings Division would obtain its tube stock by internal transfer from Tubes Division. Under the new scheme, not finalised when the research ended, each operating division was to be entirely independent of each other. For example, in the above case, if Fittings Division required tubes it would be required to produce them itself. Under the new conditions, the scope of the C.A.D. function was not known at the time the research ended and its future role in load balancing for the Company had not been defined. Clearly changes on this scale could affect the validity of the PSALM system and its future was uncertain,as were the positions of the C.A.D. personnel. This did not facilitate implementation of the system. (Post Script: The allocator mainly involved with implementing PSALM was appointed to another division and the responsibility for domestic tube production allocation has been returned to the C.A.D. manager first involved with the research).

6.3 FUTURE WORK

The following proposals have been made that work should be carried out:-

(i) To establish (a) stock levels related to demand, and (b) a review procedure for amending these as patterns and variability of demand change.

(ii) To produce stock holding costs versus service level curves for individual products in order to support decisions to provide very high service levels where this is done.

(iii) To examine manufacturing processes in order to establish product cost versus production batch size curves, and thus to obtain a minimum cost batch size and an acceptable range about this value.

(iv) To use the results of (iii) to establish the product cycle time ie how frequently batches of particular items must be scheduled for production in order that in combination with the stock, average demand may be met.

(v) Use the results of (i - iv) to produce optimum production schedules, say, at weekly intervals using linear programming techniques.

A number of linear programming solutions may be devised to solve the problem of production allocation. For example, the fact that some of the products are made on common machines was mentioned in chapter 5. It was also stated that the efficiency of these machines depended the product being processed. The problem of allocation may be formalised in the following way. Assume that products P_j ($j = 1, \dots.\ n$) are to be made on some or all of machines M ($i = 1, \dots.\ m$) and let $a_{ij} \geqslant 0$ represent the time taken to produce unit weight of product P_j on machine M_i in unit time. Then if each machine has associated with it spare capacity C_i the problem may be expressed as follows:-

Products P_j / Processes M_j	P_1	P_2	P_3	P_j	P_n
M_1	$a_{11}x_1$ +	$a_{12}x_2$	....	$a_{1j}x_j$	$a_{1n}x_n \leq C_1$
M_2 $\vdots$	$a_{21}x_1$ +	$a_{22}x_2$	....	$a_{2j}x_j$	$a_{2n}x_n \leq C_2$
M_i $\vdots$	$a_{i1}x_1$ +	$a_{i2}x_2$	....	$a_{ij}x_j$	$a_{in}x_n \leq C_i$
M_m	$a_{m1}x_1$ +	$a_{m2}x_2$	....	$a_{mj}x_j$	$a_{mm}x_n \leq C_m$
	c_1x_1 +	c_2x_2	....	c_jx_j	$c_nx_n = Z$

where C_j (j = 1, ... n) is the contribution (or profit) associated with unit weight of x_j.

The problem, therefore, is to maximise Z subject to the capacity constraints (C_j). The above could be made more realistic by adding further constraints to x_j to prevent excessive stocks being accumulated or to ensure that a minimum amount of x_j was made.

The above scheme is similar to one advanced by Oldfield (1970) but not explored in any depth.

The output from such a model would be similar to that obtained from PSALM, namely a production allocation that would vary on a weekly basis. Unless the Production Departments were to accept the principle of responding on a weekly basis to production requests, there is no reason to suspect that such a scheme would be any more successful than the existing PSALM scheme which is computationally more simple.

Obviously the linear programming allocation system could be incorporated into the PSALM framework. Although it would require more data input (concerning plant capacities) each week, the allocations produced by the scheme outlined above would be more in tune with production capabilities in a given week, since the plant capacities would indicate the state of readiness of equipment. PSALM currently obtains this information from the production budgets which are, in effect, forecasts of plant availability made, in general, some considerable time in advance of a particular week of manufacture. In consequence these will be less reliable indicators of plant availability in a given week than would estimates of machine spare capacity collected specifically for that purpose just prior to the week of interest.

In the above model spare capacity is utilised until none exists. By definition, therefore, the model is operating in an area of diminishing returns. That is to say, the output may be expected to be less than proportional to the inputs, although the linear programming model assumes linearity of response. While good approximations to linearity may be obtained in practice, the theoretical problem needsto be borne in mind.

Eilon (1962) has discussed some alternative linear programming solutions to the production allocation problem.

In the final analysis, however, the form of model that will be most successful is the one that is accepted by management for implementation, and it is this theme that Part 2 will pick up and develop.

PART II: THE IMPLEMENTATION OF COMPUTER-BASED MANAGEMENT INFORMATION SYSTEMS

Part II is concerned with the generalised problem of obtaining successful implementation of management information systems. After a review of the literature in Chapter 7 it is concluded that a means of systematically investigating user expectations is required. Chapter 8 presents the essential propositions of Personal Construct Theory which is used in Chapter 9 to develop a test for revealing user expectations. The results of using the test are presented and discussed in Chapter 10 while Chapter 11 draws conclusions from the research as a whole.

7. LITERATURE REVIEW

The practice of management is largely an exercise in decision making and the purpose of a management information system (M.I.S.) is to make information available to management in order to facilitate decision making. M.I.S. design therefore is intimately concerned with the particular form of decision making employed by management and in particular by individual managers.

Because of the relevance of decision making to M.I.S. design, section 7.1 contains a short statement of the basic ideas of decision theory and their practical relevance to systems design. Decision theory is an attempt to set human decision making behaviour within a conceptual framework. Section 7.2 describes two possible alternative approaches to human decision making in the management context. Section 7.3 explains why existing M.I.S.s tend to be concerned with quantitative aspects of management. A review of human factors aspects of computer system design is given in section 7.4 with particular reference to man-computer interaction. The reason why man-computer problem solving requires the system to ideally be designed for specific users is explained. Finally, section 7.5 describes some practical problems arising in the design of management information systems.

7.1 DECISION THEORY

Since the practice of management is largely comprised of decision making, an examination of the theory of decision making is an obvious requirement in any attempt to improve or assist practising managers.

The theory divides decisions into three categories, namely, those decisions involving riskless choice, those involving risky choice and finally decision making in the face of uncertainty (Edwards, 1954). By riskless choice is meant choice between alternatives where the decision maker is free to select that alternative whose outcome suits him best. Risky choice, on the other hand, is used to describe choice between alternatives whose outcomes have known probability of occurence but where the decision maker exercises no control over the occurence. Finally, uncertain choice describes the situation where a choice must be made between alternatives having outcomes whose probability of occurence is unknown.

In the theory of riskless choice, the decision maker is assumed to make his choice between possible outcomes in accordance with the utility each outcome affords him. The assumption is made that a decision maker will act so as to maximise his utility. Here then in theory is a rational procedure for decision making under conditions of riskless choice.

Following the work of Von Neumann & Morgenstern it is also possible in theory to derive the utility of outcomes under conditions of risky choice. Thus the expected utility of an outcome may be calculated as the product of its utility and its probability of occurence. In general the expected utility (EU) of a decision dependent upon 'n' risky outcomes is given by,

$$EU = \sum_{i=1}^{n} p_i \, u_i \quad \text{where } u_i = \text{utility of } i^{th} \text{ event}$$

$$p_i = \text{probability of occurence of } i^{th} \text{ event.}$$

Once the expected utility of each set of outcomes has been established the Von Neumann&Morganstern theory predicts that the decision maker will select the set of outcomes which maximises his expected subjective utility.

Finally, when decision making is taking place under conditions of uncertainty, Von Neumann & Morganstern (1947) have developed the Theory of Games which attempts to give guidance upon how to select the most effective strategy based upon consideration of pay-off matrices drawn up from the strategies available.

The foregoing is a most cursary summary of decision theory but it serves to present the features necessary for the following discussion. Reviews of the field are to be found in Longbottom (1972), Edwards & Tversky (1967), Edwards (1961) & Edwards (1954).

The foregoing has shown that knowledge of individual utility is a fundamental requirement if decision theory is to be used in practical situations and Mosteller & Nogee (1951) concluded that it was indeed feasible to measure utility experimentally. However Longbottom (1972) suggests that " ... it is doubtful whether an individual manager's utility functions will, or even should, be used in organisational decision making". He bases this remark upon a consideration of the practical problems involved in developing utility curves, which arise from unfamiliarity of the process and the hypothetical nature of the questioning. Neither is it clear, he suggests, whose utility function should be used in decision making - the shareholders', the managing director's or the individual manager's. His reservations are supported both by the variety of views and findings reported in Edwards & Tversky. (ibid) and their lack of precision, for example, Edwards states that " ... the subjective expected utility model is clearly wrong in detail".

Since many if not most of the decisions that are taken by a practical decision maker would be categorised as "risky" in the terminology of decision theory, the question of subjective assessment of probability must also be considered.

Once again a number of fundamental problems present themselves. Firstly, how should the subjective probabilities be treated; for example, should they range between 0 and 1 by analogy with statistical probability, and should subjective probabilities be addititive in the sense P(A) + P (not A) = 1, again by analogy with statistical probability? Secondly, how do people adjust their estimates of probability in the light of new information? Bayes Theorem (see for example Kendall, 1966b) describes in mathematical terms how a new estimate of probability P(H/D) may be calculated as to the truth of some hypothesis (H) - initially having probability P(H) - in the light of new data (D) having probability P(D/H). In fact the hypothesis (H) being tested is always in competition with other hypotheses, for example (not H), thus the formal statement of Bayes Theorem considers (r) hypotheses of the form (H_i), where i=1, ... r ($r \geqslant 2$). It may be written as follows,

$$P(H_i/D) \propto P(D/H_i)\ P(H_i)$$

Bayes Theorem is widely used as a model of human information processing. Its use is advocated by Gustafson et. al. (1969) as a practical means of reducing actuarial data in medical diagnosis. The authors quote other research that has shown the Baysian approach to be superior to multiple discriminant analysis and principle axis factor analysis in diagnosis. The main practical difficulty in the approach advocated is seen by the authors as being that of estimating the liklihoods - P(D/H). In contrast, Phillips et. al. (1966) found that while their subjects had shown some Bayesian decision making behaviour, by and large they had exhibited a marked degree of conservatism in that they tended to underestimate big changes in posterior probability - $P(H_i/D)$ - compared with the theory.

In realistic decision making situations, the choice that has to be made is multidimensional in its nature. In an industrial context, for example, a decision to introduce automation with the intention of gaining greater productivity might require a reduction of staff. This in turn could cause disruption of production by virtue of trades union activity. In coming to such a decision, a manager would have to assess his subjective expected utility of each outcome separately; that is an increased return by virtue of increased productivity and a possible loss by virtue of industrial action should it occur. He would then have to establish the overall subjective expected utility by adding together the individual utilities. In so doing he would be combining utilities calculated along different dimensions of his utility space. In the theory of riskless choice it is usual to assume that the utility of a multi-dimensional alternative is the sum of the utilities of its component parts. Thus Yntema & Torgerson (1960) discuss three methods of conveying a man's decision rules to the computer, each of which is concerned with establishing the relative utilities of the human decision maker and converting these to a decision function that can be used by the machine. The authors emphasise that interactions between the main effects can be ignored. This idea is embodied in the experiments of Phillips et al (ibid) but are challenged by Keeneg (1968) who suggests that a quasi-seperability utility function should be introduced yielding,

$$U(x,y) = U(x,o) + U(O,y) + K.U(x,O).U(O,y)$$

where K is an arbitrary constant, $U(x,y)$ represents the utility of (x and y), $U(O,x)$ represents the utility of (x) etc.

The above expression would replace the seperable utility function, $U(x,y) = U(x,O) + U(O,y)$ implied by Phillips et al.

However perhaps it is in decision making under conditions of uncertainty that existing theory could be of most practical value, but in fact it is in this particular area that its application is most questionable. One of the fundamental principles of game theory developed by Von Neumann & Morganstern to treat decision making under uncertainty is the min-max principle. This postulates that under uncertainty a decision maker will attempt to minimise his maximum loss. That is, he will select the strategy from those available to him that has the least unattractive outcome if it is not successful. By applying this principle it is possible to select a 'best' strategy within the definitions of the theory. However the theory was developed for application in a gaming situation where the idea of opponents and competition are relevant notions, while in many practical situations of decision making under uncertainty there can be little suggestion of the decision maker being in competition with anyone, except perhaps with nature. It would be a particularly cynical outlook to suggest that nature always competed against man's objectives in a manner comparable to a player competing against his opponent(s). Furthermore it is often difficult to cast real life decision making into the required pay-off matrix form required to apply game theory, so much so, that one writer on the subject has remarked that it is an interesting methodology that requires useful applications. (Williams,1954).

The foregoing serves to show that, at least for the present, decision theory has little to offer in a practical way to M.I.S. design. While it is claimed to have predicted real life decision making behaviour, the basic models, while elegant in conception, do not appear to be rich enough to predict the actual decision making behaviour of an individual human being. By analogy with atomic theory, while the model of an atom as a nucleus having orbital electrons is a good predictor of the macroscopic behaviour of materials, it is inadequate to represent the observed behaviour of individual atoms.

7.2 ARTIFICIAL INTELLIGENCE & CYBERNETICS

Brief mention should be made of the possibility of developing intelligent machines. Conceptually, at least, if machines could be built which were able to exhibit intelligence then they could be set to the task of decision making.

Conventional machines are used by man to amplify or extend his capabilities. For example, they are used to amplify his strength, endurance, speed of working etc. By similar reasoning it might be argued that if machines could be built that exhibited intelligent behaviour of even a limited kind, then the <u>possibility</u> of developing machines that were even more intelligent than man would have been demonstrated. Clearly much revolves around the definition of intelligence in this context, but since 1833 when Babbage designed his analytical engine and demonstrated that the execution of logical operations could be performed by machine, machines capable of exhibiting increasingly complex behaviour, for example learning behaviour (Leondes and Mendel, 1969), have been built. One of the best known and most tangible aspects of artificial intelligence research has been the development of chess playing computer programs, (Zobrist and Carlson, 1973). However, as yet the software remains inferior to the best human exponents of the game and doubts have been expressed about the probability of this situation radically changing in the next few years (Rosen, 1973). Once again it is concluded that the capability of practical decision making in an industrial situation is even more remote.

A rather different use of cybernetics in management is described by Beer (1959, 1966, 1967, 1972). He points to the similarity of all living organisms in terms of their information processing organisation and suggests that if human organisations are to be viable then they too must arrange themselves in a comparable way. In the cybernetic scheme of things it is the type of organisation structure adopted by management rather than its content that is of paramount importance. For example the method used by a decision maker to make his decisions (which might include linear programming, statistical analysis and forecasting techniques) would be of less importance - often black box decision functions are used - than the way in which he received the information upon which the decisions were to be based and the way in which the results of the decision making were distributed to the rest of the organisation.

In a recent book Beer (1972) has used the above philisophy to develop a model of the firm based upon the neuro-physiological organisation of the human body. The model has five hierarchies of command between which suitably filtered information flows. Beer sees existing techniques of O.R. being used to process and filter the large amounts of information that flow in any organisation and suggests that computers would enable the processing to be carried out at an acceptable speed. Although the author claims to have implemented his ideas in a number of practical situations, details do not appear to have been published. In one particular case the above approach has been implemented on a national scale (Hanlon, 1973) although its future is not clear now that the sponsoring government has fallen. Thus, although Beer makes a very persuasive case for his cybernetic approach to management organisation, for the present at least, the verdict must be 'case not proven'.

7.3 PRACTICAL DECISION MAKING, EDP & MANAGEMENT SCIENCE

Although decision theory has so far failed to satisfactorily provide an explicit account of how human decision making takes place, the human decision maker is nevertheless able to take extremely complex decisions with a considerable amount of success. Furthermore he is usually able to do this without any of the high degree of quantification of the variables that decision theory requires. In such situations, the human decision maker uses the process known as judgement. Good (1962) has discussed a number of aspects of judgement in management situations.

In mathematical terminology, judgement may be described as the resultant of a weighted function of all the variables an individual perceives to be relevant to the decision at hand. However individual perceptions of the dimensions of a given problem vary greatly and a further complicating factor is that there would seem to be no need for perceptions to be conscious in order to be included in the weighted function. Thus, while this mathematical conceptualisation may be a useful way of describing the operation of judgement, unless some means can be found of establishing the variables an individual perceives to be relevant to a problem, it offers no practical solution toward understanding how human decision making actually takes place. In effect this is restating the problems encountered in attempts to apply decision theory.

The area of perception, judgement and understanding will be returned to in the discussion of problems arising in M.I.S. design (7.5).

Recent interest in M.I.S.s has grown from the increasing use of computers in large organisations in combination with the greater speed and capacity of modern computers. While information has always been available to management for decision making it has often not been readily available and because of this, decisions would frequently have to be taken without managers having the opportunity to take into account all the relevant known (ie potentially available) facts. It seemed, therefore, that this situation could be remedied by using computers to provide such information as managers required both speedily and accurately. However, before this could be done the information required by managers had to be identified, and therein lay the problem. Because managers use judgement to such a large degree in their decision making and because, as the earlier discussion pointed out, it is very difficult to establish just what relevant information is required to support sound judgement in each individual case, M.I.S.'s have often failed to find general acceptance.

It is no accident therefore that M.I.S.s have found most application to date in areas of decision making concerned with readily quantifiable variables. Typically, M.I.S.s concerned with financial variables are amenable to a computer based approach because data such as cash flows are quantifiable and can therefore be readily manipulated by a computer. Additionally, in the case of financial decision making, the rules whereby decisions must be made are well established over a wide range of decisions, and thus the information provided by a new computer-based financial M.I.S. is more likely to be accepted by management than a M.I.S. pertaining to, say, manpower planning or marketing strategy where the basic principles are not as widely agreed upon. Hence management decision making in the latter type of application has tended to rely upon human judgement.

Since computers cannot exercise judgement, management scientists have attempted to find alternative ways of making decisions which can be carried out by a computer. This is one reason why mathematical modelling is currently enjoying such a widespread use in management science. Computers are extremely adept at manipulating such models, in general much more so that human beings. Providing that sufficiently accurate models can be developed, managerial judgement can be to some extent superceded by computer-based decision making which may be carried out with more speed, greater accuracy and with a higher degree of consistency, the management scientist would argue. It is reasonable to speak of computer-based decision making in terms of the definition of decision making implied in 6.1, namely, that decision making is the selection of one option from the range of options available. On this basis a computer can obviously make decisions, providing the rules of selection are known to it.

Effectively what happens in such a situation is that management decision making is moved to a higher level, whereby the manager decides only the strategy upon which his decisions are to be based and the computer then grapples with the detailed decision making. The extensive linear programming and simulation models used by large oil companies (see, for example; Wagle, 1969) to 'optimise' their decision making are examples of an application of the above approach and they may be regarded collectively as an example of a "symbiotic relationship" that Licklider (1960) suggested should ideally exist between man and his computers.

The essence of his idea was that if man and the computer shared the task of decision making by doing what each did best, then the combined results should in practice be better than those possible by each agency acting independently. Licklider pointed out that many problems can be pre-formulated and in theory at least a computer is not required, but in practice the effort to think through problems in the detail necessary to establish the validity of a particular approach is often prohibitive.

The "man-computer symbiosis" concept, therefore, suggests that great potential exists for man to increase the power of his decision making. Its weakness however, lies in the particularly stereotyped way problems must be presented to the computer before this power can be harnessed. Because the formulation of problems for computer processing is, for the present at least, a complex task it is usually carried out by experts. It is not done by the manager who will ultimately want to use the product of the computer processing for his decision making. Peace and Easterby (1973) have argued that this practice can give rise to the situation where a manager either does not understand or simply does not accept the terms in which his problem has been framed for use by the computer.

7.4 HUMAN FACTORS CONSIDERATIONS IN COMPUTER SYSTEMS

The traditional role of human factors research has been to investigate the relationship between man and machine, usually with the objective of optimising their joint performance. In one sense a computer system is merely a hardware system and as such it may be regarded as similar to other hardware systems, for example, a printing press, a milling machine or a motor car. If this approach is adopted, the familiar "knobs and dial" methodology is as appropriate a means of investigation of the man-machine interface in a computer system as it is in any other equivalent hardware system.

However to regard a computer system exclusively as a hardware system is an inadequate concept. Although computers have the potential to assist in the human decision making process, the key to realising that potential lies largely in software and not hardware design (Parsons, 1970) although the latter must not be ignored. Thus a computer should be regarded not merely as a man-machine system, but as a man-machine-logic system (Grace, 1970). Failure to explicitly consider the role of software in man-computer interaction has without doubt contributed to the failure of human factors practitioners to realise man-machine symbiosis, a fact upon which Nickerson (1969) comments at length.

There appears to be another reason, however, why progress in this field has not been as rapid as it should, and this stems from the position traditionally ascribed to man in man-machine system studies. Perhaps because many human factors practitioners originally trained as engineers, it is usual for the function of man in a man-machine system to be described in terms not very different from those used by equipment designers (Gagne, 1963). While this approach is quite reasonable when man-machine interaction is confined largely to physiological aspects as is the case, for example, in production line working, it is patently unsuitable in areas where interaction involves psychological considerations (Jordan 1962, 1963). In the study of computer aided problem solving there can be no doubt that human psychology must be taken into account and in such cases, as Bowen (1967) remarks, " ... man does constitute a different kind of resource in a system and ... his contribution for good or ill is qualitatively different from the contribution of other system components".

If man-computer interaction introduces a new dimension into human factors concepts, clearly new methodologies are required to explore, classify and quantify the variables concerned. Knowles et al (1969) discuss the use of models in system design and the problems of predicting system performance at the design stage. In particular, they note the difficulty of designing for people in the system. Wulfeck and Zeitlin (1962) also call for the use of "new psycho-physical methods or efficient modification of old ones" in order to provide data for predicting the effectiveness of new system designs. Effectiveness here must be taken to include acceptability of the new system to the user.

Sackman (1970) suggests that an interdisciplinary approach incorporating the methods of behavioural science might enable better man-computer communication to be achieved. An example of how such methods might be applied in practice is to be found in Bare (1966) who uses the Osgood Semantic Differential to elicit from 100 subjects their attitudes toward machines, thus permitting explicit account to be taken of psychological aspects of man-machine interaction.

In fact this approach has some similarity to that developed in the next chapter, but use of the Osgood Semantic Differential confines attention to global attitudes whereas in the study of computer aided problem solving it is individual differences that are of importance.

The importance of taking into account individual human differences in any given man-machine system design may be established by considering the relative contribution man and machine separately make to the overall system effectiveness. Thus in the case of a capstan lathe operator most of the system integrity is vested in the machine and the operator is required to do little if anything that could be affected by individual human differences between operators. Thus in designing such a man-machine system it is sufficient to design for the average man. The reverse is the case in a man-computer problem solving situation. The man is free to formulate the problem in his own highly individualistic way while the machine's function is limited to that of processing data according to the particular formulation employed. Clearly in the latter situation, the effectiveness of the combined man-computer system is largely controlled by the man component. If the individual's freedom to evaluate the various formulations of his problems is artificially constrained by inadequacies of the computer system, then the overall man-computer system effectiveness will be reduced.

While the foregoing example serves to demonstrate the importance of considering individual human differences in man-computer decision making situations, the literature suggests that allowing for such differences is a particularly difficult thing to do in practice (Ware, 1964; Grant and Sackman 1967; Shackel, 1969).

The next section describes a particular example of man-computer decision making, the computer-based management information system.

7.5 PROBLEMS ARISING IN M.I.S. DESIGN

A case has already been made supporting the idea of system design for individual users in man-computer problem solving situations and it is the particular case of an essentially one-to-one man-computer relationship that is considered below. In so doing the definition proposed by Shackel (1969), which regards man-computer interaction a "very direct, close coupled computer usage", is used. Man-computer interaction may often be better described as "man-software" inter-action (Parsons, op cit). This is particularly true for management information systems where probably the only "hardware" the user sees is printout and his only mode of response a "run request" sheet.

However, whatever form the human interaction with a particular system takes, we may assume that for successful implementation the user should ideally have confidence in the accuracy of the inputs, should accept the validity of the operations performed on the data and should be satisfied with the form the output takes. It follows that to achieve this result, the system designer must identify the appropriate inputs, ascertain the requisite operations to be performed on the input and finally arrange for the output to be presented in an effective and acceptable form.

At this point a number of practical problems may arise. A management information system is generally regarded as a job aid, and to fulfil that objective some reasonably clear idea of the job to be aided must be obtained. Management information systems are characterised by complex human information processing, which is not open to direct observation or the use of task description techniques. Formalisation of procedures in task description form may produce 'text book' accounts which in reality are not followed, and often are not as effective as their "fast and dirty" alternatives. Knowledge of the latter can often give valuable insight to the analyst.

A related problem is that many functions may never have been previously formalised, so that once this process has been carried out neither the analyst nor the potential user can be certain that the formalisation is a valid one, in that it will be able to deal with real life situations. In addition, the effect of asking the potential user to give a formal description of how he performs his function may expose him to the risk of having <u>his</u> version compared to some other statement of function that may also exist within his organisation.

An even more intractable issue is when the potential user is not able to externalise the protocols he observes in fulfilling his function even though his behaviour suggests that these exist quite markedly in some internal form. This situation has been observed in a parallel context by Bainbridge et al (1968). Thus it may mean that matters of great importance turn on aspects that, to the analyst, seem relatively unimportant or even irrelevant because they are not stressed by the potential user to whom the effects are so commonplace as to be unworthy of mention.

Before discussing how some of these problems may be solved, it is convenient to consider the nature of a desirable solution and in particular to introduce the notion of "technical" and "total" solutions.

A technical solution is said to have been achieved when a system has been designed that can accurately represent the status of relevant physical variables, and produces valid analyses and predictions based on these data. On the other hand a "total solution" is said to have been achieved when additionally the results of the technical solution are available in an understandable form for the user to assimilate and use.

These ideas are presented graphically in the Venn diagram of Figure (7.1). In practice, a technical solution is often all that is ever achieved and the relationship between technical and total solution is usually grasped more by intuition, than by formal procedures, which are few and far between in this area.

It is also widely accepted that the total solution should be a dependent function of the technical solution, rather than the converse, deriving from a recognition of the relative inflexibility of physical variables in the real world compared with human adaptability. It is assumed that if the physical constraints imposed on a problem can be resolved (ie a technical solution) the human aspects will be able to adapt accordingly. While we may hope that the man will adapt, in practice he may not do so and in such cases it may be better to work toward a technical solution from the standpoint of the total solution.

To elaborate this total solution concept, consider a managerial situation where it is desired to improve the profitability of a small operating division controlled by an individual manager. Further, let us assume it has been decided that the improvement in profitability is

VENN DIAGRAM TO ILLUSTRATE THE CONCEPT OF A TECHNICAL AND TOTAL SOLUTION

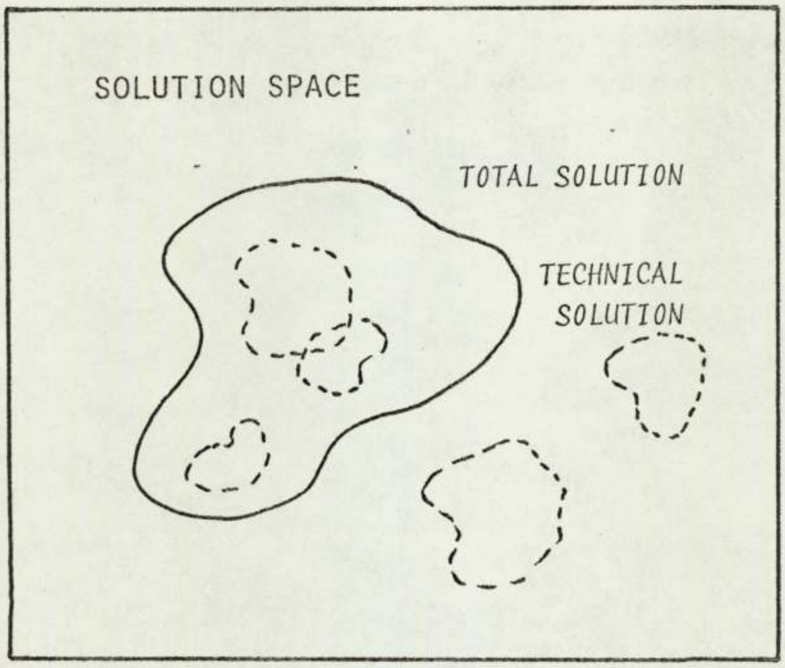


FIG (7.1)

to be obtained by improved methods of control, these to be devised and implemented by outside experts in collaboration with the present manager, who will ultimately assume responsibility for its ongoing operation.

The experts may be able to achieve the desired standard of performance in a number of ways independently of human factors aspects and so achieve a technical solution. The required return on investment might be achieved by better calculation of product mix made on the basis of, say, a linear programming model; or alternatively it might be achieved by the introduction of new stock control policies where stock levels are based upon statistically determined customer service levels, the latter being related to sales volume. Here only physical variables enter into the control schemes. Regardless of whether the manager approves of such schemes, there are many reasons why he may nod in apparent agreement while the experts are around, only to quietly dismantle their work when they ultimately withdraw. A non-numerate manager, for example, whose experience is largely confined to line management may fail to appreciate the conceptual elegance of regarding his area of responsibility in terms of the linear functions implied by one of the solutions suggested above. In such circumstances, the linear programming approach would constitute a technical solution that was not contained in the manager's total solution space.

On the other hand, a technical solution that might fall within the total solution space of our hypothetical manager could be the designing of incentive pay schemes for his shop floor workers. Because of his previous experience such a solution would constitute an approach with which material sympathy existed, and with outside help at the design stage it could be expected to achieve the profitability requirements, not only at the outset when the experts were in control, but as a continuing system. In this way a total solution would have been achieved.

The foregoing is a practical recognition of the fact that while it is sometimes possible to retrain or replace a senior manager by one more suited to operating a projected system, there are many more cases where it is neither possible nor desirable. In these latter circumstances, a better total solution can be achieved by adopting a scheme more acceptable to the ultimate user, although possibly less efficient in purely technical terms. The criterion is acceptability to the user, and as Fig. (7.1) indicates this may simply mean choosing one of a number of technically equivalent solutions. By doing so nothing may be lost technically, but much may be gained totally.

A computer based management information system is a system of man/software logic and its elements are a mixture of both tangibles (eg hard data processed by the logic) and intangibles (eg ideas and concepts inherent in the logic). Unfortunately the way in which the user perceives these elements and their inter-relation within the system is an exclusively cognitive operation and is thus not susceptable to direct observation, yet we need these perceived inter-relationships to help define the user's criteria of acceptability of a system.

It follows that if the user's internal criteria for assessing system elements and their perceived inter-relationships could be established there would seem to be considerable justification for using them in the design phase. Thus, provided the hard data may be adequately treated, so achieving a range of technical solutions, the final total solution may take its shape from the individual criteria and expectations of the user. The problem is whether these expectations can be established in a reasonably unbiased way. The remaining chapters of this thesis attempt to show that they can.

8. PERSONAL CONSTRUCT THEORY

The reasons for taking into account individual user expectations when designing man-computer problem solving systems such as the M.I.S. have now been described and subsequent chapters will show how this can be attempted and will record the degree of success achieved. However, since the methodology employed derives from origins that are not widely known, the present chapter is devoted to a review of the relevant theory.

The Theory of Personal Constructs was first enunciated by Kelly (1955) and it is based upon a philosophical standpoint he terms as constructive alternativism. Constructive alternativism postulates that man's perception of truth and reality is limited only by his ability to construct alternative interpretations of the events that he perceives. Kelly discusses the reasonableness of his philosophical position at some length, but it is sufficient for present purposes to move immediately to the Theory of Personal Constructs and to present its main contentions. Kelly presents his theory as a fundamental postulate and eleven corollaries. This classical form of presentation is typical of the elegance of the theory itself and its method of development in the original work. The Theory is concerned with the psychology of man and is regarded by psychologists as being presented at a relatively high level of abstraction (Bannister and Mair, 1968). It is perhaps surprising therefore, that the theory leads directly to a practical, test methodology that was devised by Kelly for use in psychotherapy.

Although Kelly was concerned with clinical psychology, he saw the potential value of his theory and its associated techniques in other fields including that of management. It is with the latter situation that the present study is concerned of course. In the outline of the theory below, the review by Bannister and Mair (ibid) is closely followed.

8.1 FUNDAMENTAL POSTULATE

"A person's processes are psychologically channelized by the ways in which he anticipates events."

Here Kelly is postulating that an individual's actions - mental and physical - are constrained by the way he predicts events will occur. External events exist in their own right but they are channelized by each individual into certain patterns and groupings in order that he might better predict the future.

To obtain these groupings an individual uses his personal construct system. A personal construct may be regarded as a decision function; it is a means whereby certain events, things or people may be determined as being alike or different according to some criterion and yet distinct from other events, things or people.

8.2 CONSTRUCTION COROLLARY

"A person anticipates events by construing their replications."

It is suggested that in the course of time certain events are perceived to be similar and others dissimilar in respect of various characteristics. More importantly it is argued that, in terms of human perception, the identification of similarity suggests the existence of contrast. For example "light" implies the existence of "dark". Obviously if such a scheme is to be used to categorise events, the possibility of events being neither light nor dark must be allowed.

This corollary suggests that man categorises events as they repeatedly occur by a system of constructions that depends upon contrasts, that is, bi-polar or dichotomous relationships. Not all events can be associated with a particular dichotomous relationship and thus a number of (possibly related) such relationships are used by an individual to anticipate events.

The difference between the above classification and that used in formal logic should be noted. Thus the (light-not light) relationship in formal logic includes within "not light" not only "dark" but "human", "motor car", "bucket" etc.

8.3 INDIVIDUALITY COROLLARY

"Persons differ from each other in their construction of events."

Faced with the same set of events it is likely that two individuals will interpret their meaning differently. This occurs because their construction systems are likely to be different.

In terms of personal construct theory, this corollary explains why in computer-assisted problem solving the computer system must be able to accommodate individual requirements.

8.4 ORGANISATION COROLLARY

"Each person characteristically evolves, for his convenience in anticipating events, a construction system embracing ordinal relationships between constructs."

The separate constructs that are used to classify events for the purpose of prediction are not independent one of another but are interrelated by the individual in order to improve the quality of his predictions. What is more, some constructs are more important than others - according to an ordinal relationship. Kelly suggests, therefore, that every individual erects a hierarchy of constructs in order to introduce some degree of consistency into his decision making.

8.5 DICHOTOMY COROLLARY

"A person's construction system is composed of a finite number of dichotomous constructs."

Sections 8.1 - 8.4 have introduced the notion of constructs to explain how individuals interpret events, have discussed some characteristics of constructs and have suggested that these constructs are hierarchically arranged in order that effective decision making may be achieved. This corollary re-emphasises the bi-polar nature of constructs. It states that events grouped by a particular construct must fall exactly into one category (pole) or the other. Thus in the case of the construct "light dark", events must be classified so that they are regarded as being either light, or dark, or alternatively it is decided that the construct does not apply. The possibility of an event being categorised between the poles of the construct is not allowed by the theory.

This does not preclude the existence of mental scales along given dimensions however. Thus any event could be ordered as the lighter or darker of a pair by using the "light-dark" construct, and in this way a whole series of events could be ordered on a light-dark scale by repeated operation of the construct on the events. A construct is merely a device used by individuals to interpret events. It does not affect the separate existence of events which may themselves exist in a finely differentiated state upon an interval scale.

8.6 CHOICE COROLLARY

"A person chooses for himself that alternative in a dichotomised construct through which he anticipates the greater possibility for the elaboration of his system."

In this corollary the "elaboration of his system" may be taken to mean the definition and extension his system. Indeed this was how Kelly first propounded the Choice Corollary.

The suggestion is that an individual will prefer an alternative that either better defines or reinforces his existing construct system or provides a new extension that is compatible with the existing system. As Bannister & Mair (ibid) point out, this choice is not an intellectual one but rather is governed by the individual's awareness of the possibilities involved. Because of this the individual may not choose that construct which is objectively best for him. It must also be remembered that the choice corollary is concerned with the choice between constructs, not the objects differentiated by constructs.

8.7 RANGE COROLLARY

"A construct is convenient for the anticipation of a finite range of events only."

The range corollary explains why individuals use a whole system of constructs to deal with all the events they experience in real life.

Kelly defines two terms to describe a construct, the "focus of convenience" which covers a set events that are easily subsumed by the construct and the "range of convenience" which covers a wider set of events that may be subsumed by the construct but with which the construct may deal less effectively.

In the study described in Chapter 9, only one sub-system of constructs is being explored, namely that relating to an individual's job. As will be seen, the relatively limited range of convenience of some of the constructs in this sub-system had to be taken into account in deriving the final form of the repertory grid test.

8.8 EXPERIENCE COROLLARY

"A person's construction system varies as he successively construes the replication of events."

Since the fundamental postulate claims that an individual's processes are directly related to his anticipation of events, it follows that as anticipation becomes realisation the individual will re-assess the effectiveness of his construct system in terms of its predictive capability.

8.9 MODULATION COROLLARY

"The variation in a person's construction system is limited by the permeability of the constructs within whose range of convenience the variants lie."

The permeability of a construct is its capacity to subsume new events. Generally speaking the more super-ordinate a construct is, the more capable it is of accommodating new events. For example, the construct "light-dark" is super-ordinate to the construct "black-white" and is also more permeable because it could therefore subsume the new construct "pink-indigo" and a host of other colour shades whereas the construct "black-white" could not.

8.10 FRAGMENTATION COROLLARY

"A person may successively employ a variety of construction sub-systems which are inferentially incompatible with each other."

It is this corollary that enables Personal Construct Theory to explain the inconsistency of behaviour that may be observed in individuals from time to time. The individual may or may not be aware of these inconsistencies. In the former case he may be acting quite consistently with respect to some super ordinate construct that is not known to the observer, in the latter he may simply not be aware of the inconsistencies that exist within his own construct system.

8.11 COMMONALITY COROLLARY

"To the extent that one person employs a construction of experience which is similar to that employed by another, his processes are psychologically similar to those of the other person."

If two individuals have developed similar construct systems (or sub-systems) then their anticipations arising from events occuring in their shared experience will be similar. It should be noted that similar previous experience of events is not a necessary condition to achieve similar construct systems.

This corollary makes an important contribution to the theoretical basis of the repertory grid test discussed in chapter 9.

8.12 SOCIALITY COROLLARY

"To the extent that one person construes the construction processes of another, he may play a role in a social process involving another person."

This corollary is concerned with inter-personal behaviour. It explains why, for example, an adult will explain things to a child in a manner different to that which he would use to explain the same thing to another adult. He attempts to construe the construction process of the child and acts accordingly, that is, he plays an appropriate role.

8.13 SUMMARY

Each individual has a construct system which is composed of both related and unrelated hierarchies of constructs and this system is peculiar to himself. He is able to re-arrange its organisation both in terms of the construct hierarchies to be found at a given time and in terms of the elements gathered within the range of a particular construct, at will. The desire to re-organise arises when the existing organisation of constructs fails to anticipate events in the real world with acceptable accuracy. Thus each individual interprets the events he perceives by use of his personal construct system and in this way tries to anticipate the likely outcome of future events. When he is unsuccessful in his predictions, he modifies the construct system in order that, hopefully, its subsequent operation will be more consistent with reality.

However not all constructs are easily changed. The stability of the construct system is maintained by core constructs. Such constructs are those that have stood the test of time, as far as the individual is concerned. Experience has shown him that these constructs are reliable in adequately anticipating future events. Core constructs underpin the whole system, they are unlikely to be changed and unwitting external attempts to do so may prove disastrous.

Although an individual expands his construct system to cover the whole of his activities, Personal Construct Theory was developed to deal with those problems arising out of inter-personal relationships. In particular Kelly used it in psychotherapy, although he saw that it had potential application in management situations. The theory is used here to try to understand managerial expectations of computer based information systems.

Since, according to the theory, an individual's construct hierarchy is his only way of interpreting external events and thus deciding his future course of action, it may be deduced that construcs will exist that map on to all his occupational activities. If these constructs and their relationships could be defined in some meaningful way, they could be used by a management information system designer to service some of the user's requirements not defined by the physical world, as was discussed earlier. In addition the designer would be given a systematic diagnostic tool with which to evaluate the relative importance of the various elements involved in the complete work activity of the user, since the weighting device of the user (his construct system) would be available. Within the context of a management information system, the elements concerned might be, for example, the order entry procedure, the sales ledger, administrative policies, etc., etc.

The Repertory Grid Methodology was devised by Kelly to elicit personal constructs. Specific tests may be produced in various ways, and some examples of these different methods are quoted by Bannister and Mair. However, the fundamental principle is similar in all cases. The subject is asked to differentiate on the basis of similarity and dis-similarity between individual elements grouped into sets of three elements (triads). The elements themselves may be given or, more preferably, the subject may provide them himself in response to questions. He is then asked, "In what way are two of these three objects (or persons) similar and opposite to the third". The basis of the distinction is then taken to be the construct. For example, on being asked to differentiate in the above way between 'profit and loss account', 'weekly sales figures', 'annual salary review', the subject might respond by saying that 'profit and loss' and 'weekly sales figures' were control functions and 'annual salary review' was not a control function. It is important to establish that the subject perceives a genuine dichotomous relationship even if this is only a "****** - NOT ******" relationship, before the construct is accepted as genuine. There is a significant difference between the "Control - Not Control" relationship and saying that 'profit and loss' and 'weekly sales figures' are similar by virtue of being control functions and that 'annual salary review' is perceived as having no relationship with 'profit and loss' and 'weekly sales figures' whatsoever. In the latter case 'profit and loss' and 'weekly sales figures' are in the range of a construct having "control" as one pole and 'annual salary review' is in the range of another, possibly quite unrelated construct.

9. INVESTIGATING SYSTEM USER EXPECTATIONS AT THE DESIGN STAGE

Using the theory outlined in the previous chapter section 9.1 explains the basis of grid tests and shows how personal constructs may be elicited and understood. Section 9.2 is concerned with the development of a specific test for use in M.I.S. design, while the results of using the initial design are presented in 9.3 together with discussion and suggestions for improvement to the test content. Finally 9.4 indicates how the results obtained may be analysed and is concerned in the main with the principles of factor analysis as they relate to repertory grid analysis.

9.1 THE REPERTORY GRID TEST

The repertory grid test takes its name from the original application of Personal Construct Theory in psychotherapy where it was used to establish the roles various individuals played in a subject's construction of life. It was envisaged that a subject had a repertory of constructs which he used to establish the role he would attempt to play with those about him. The purpose of the test was to elicit a table, or grid, of these important constructs.

In order to understand the theoretical basis of a repertory grid test, it is necessary to consider how communication between individuals takes place in terms of construct theory. It is a consequence of the fundamental postulate of Kelly's theory that an individual can only attach meaning to information by operating on it with his construct system. In order that two individuals derive the same meaning from the same information, one of two alternative situations must occur. Either the individuals must have identical construct systems which is unlikely (Individuality Corollary), or one of the individuals concerned must subsume the construct system of the other within his own system (Commonality Corollary).

In the latter case one individual is able to play a role with the other (see 8.12 - Sociality Corollary) and effective communication can take place to the extent that the receiver of the information is able to use a similar system of constructs for decoding the message to that used to transmit it. Since the receiver of information will not in general be able to completely subsume the construct system of the transmitter, information exchange will not be perfect and distortions will occur.

In many exchanges of information the receiver will only have to subsume parts of the transmitter's construct system (see 8.10 Fragmentation Corollary) in order that effective communication can occur. This provides an explanation of how individuals may converse effectively on some topics and yet be at cross-purposes on others. In such a situation the communicators have been successful in subsuming only some of each other's relevant construct sub-systems.

The repertory grid test is a formalisation of the communication procedure described above, in which the experimenter attempts to subsume the structure (inter-relationships between constructs) and content (labels given to constructs) of the subject's construct system. This is done with aid of a set of constructs known as elements because they are sufficiently basic to be readily and identically comprehended (subsumed) by subject and experimenter alike. The subject is asked to differentiate between groups of the elements in a particular manner and to indicate the basis of his differentiation in each case. Since by definition superordinate constructs are developed by individuals to distinguish between subordinate constructs, providing the differentiation procedure is carried out properly, for each of the groups sorted in this way the basis of differentiation will be a super ordinate construct of the subject. This then is the basis of the repertory grid test.

However the problem of obtaining the elements remains, and without these the more important super-ordinate constructs cannot be obtained.

The simplest solution from a practical point of view would be for the experimenter to provide the subject with a set of elements, but this would give rise to at least two objections. Firstly the subject might not recognise the element at all, or secondly, he might construe it within a different context to that envisaged by the experimenter. This latter outcome would not be too serious since it is a feature of the test to establish such usage, but it could well detract from the selection of sorts that had been planned.

It is generally accepted that the best method of obtaining elements is to ask the subject to supply these himself in response to general questioning. The implicit assumption here is that to the extent that individuals (in this case the subject and experimenter) can meaningfully communicate, their construct systems will show some degree of similarity. By speaking generally about the area of interest the experimenter enables the subject to identify the corresponding sub-system within his own (the subject's) construct system. When requested to do so, the subject can then select from that sub-system an element which, for him, particularly typifies it. It is important that this element, produced by the subject, is similarly construed by the experimenter, although its implications in general will not be immediately clear to the latter. A category of constructs which can be readily construed without distortion of meaning are those relating to tangible entities, for example people, animals and inanimate objects. It is the experimenter's knowledge of the area being probed and his similar construction of the subject's low order constructs (ie elements) that permits him to subsume (ie interpret) the information regarding the content and structure of the subject's construct system within his own.

Proceeding in this way the experimenter can elicit elements from a number of different parts of the subject's construct system, the particular parts chosen depending on the objectives of the study. Having obtained the elements by general questioning in the above way, super-ordinate constructs are elicited from them by a process known as "triad sorting" which will be discussed next.

A personal construct has been described as the means whereby an individual can differentiate events as being similar or opposite in some sense. (It will be remembered that events which may not be classified in this way are regarded as falling outside the range of convenience of the construct). Thus if a subject is asked to differentiate between individual elements arranged in groups of three (triads), on the basis of a perceived similarity between two of them which is in some way seen as opposite to the third, the basis of differentiation is one of the subject's personal constructs. The subject is literally asked, "In what way would you regard two of these elements - in practice the word 'elements' would not be used, instead the actual elements would be given - as being similar and opposite in some sense to the other?" Depending on the elements available and the triad grouping selected, the experimenter may obtain a variety of constructs from a subject, which may then be set out in a grid form as shown schematically in fig (9.1).

The number of elements selected in the first stage of the test is a matter of compromise. The purpose of the exercise is to obtain a representative set of elements from the construct space of the subject. Thus the larger the sample the smaller the chance of bias, other things being equal, but this desirable feature has to be balanced against the problems of processing a large number of elements through the remainder of the test. In practice 20-25 elements seems to be a reasonable number to acquire.

The number of constructs obtained in the second stage of the test depends upon the elements selected in the triad sorts, the number of triad

A SCHEMATIC REPERTORY GRID

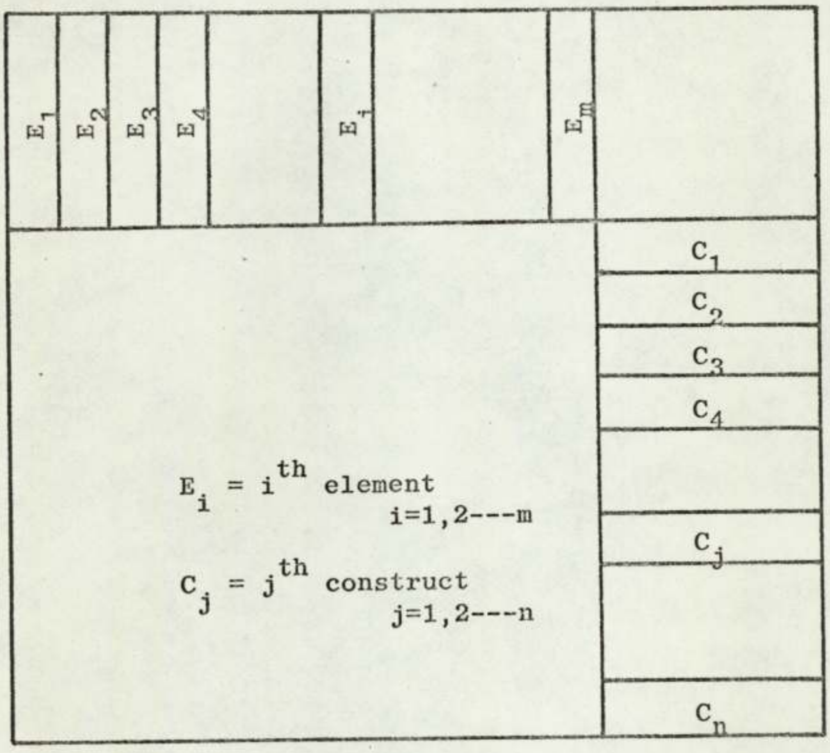


FIG (9.1)

sorts put to the subject and his ability to uniquely differentiate between individual triads. The theory predicts that the number of constructs that may be elicited is finite (see 8.5 - Dichotomy Corollary); and experience has shown that the number of constructs that may be obtained by the foregoing procedure is relatively small, seldom greater than 20-30 and often considerably less. This may be expected to be particularly the case when only a sub-system of the subject's construct space is being investigated, as in the present study. Thus while the number of unique triads possible in a selection of 20 elements is large (nCr = 1140, for n = 20 elements and r = 3) only a small number of these will be required to elicit all the constructs available. In consequence it may be found that some triads cannot be subsumed within the subject's construct system, while others are subsumed under constructs already elicited and are therefore non-productive in eliciting new constructs.

The final stage of the test involves placing scores in the body of the table shown in fig. 9.2. This is done row by row, and each element is scored according to the degree it is in accordance with the construct on that row. A five point scale was used throughout this study with scale values 2, 1, 0, -1, -2. Positive scores indicated that the element referred to one pole (designated "side entry A" on the grid) and negative scores that it referred to the other (side entry B). The greater the absolute score given, the greater the extent to which the appropriate entry applied. Zero was used to indicate those elements to which neither side entry applied. The admission of constructs which do not relate to all the elements broadens the scope of the technique but can give rise to difficulties in the analysis of the grid. Detailed discussion of grid analysis is presented in 9.4.

Having described the basis of the repertory grid test its relevance to the present study may now be considered. The Fragmentation Corollary suggests the possibility of studying only parts of an individual's construct system, in particular that sub-section which relates to his working situation. Accurate information about this sub-system would permit his expectations of new systems to be anticipated. In this way potential mis-matches between his expectations and the new system's capabilities could be identified and adjusted.

However this approach has a potential difficulty which becomes evident from a consideration of the relevant theory. In limiting the investigation to a sub-system of the individual's construct system, the study is being confined to an investigation of relatively sub-ordinate constructs (see 8.4 - Organisation Corollary), since sub-systems are bound together to form the overall construct system by super-ordinate constructs. This must be contrasted with Kelly's practical use of the theory, which was concerned with the investigation of super-ordinate constructs. The latter, of course, have rather different properties to those of sub-ordinate constructs. In particular a super-ordinate construct is associated with greater permeability (see 8.9 - Modulation Corollary).

One of the basic questions to be answered in this study was whether the test methodology devised by Kelly for use with super-ordinate constructs would be equally useful when applied to less permeable, sub-ordinate constructs which by definition have less flexibility in their application.

9.2 DEVELOPMENT OF A REPERTORY GRID TEST TO AID M.I.S. DESIGN

This section describes the relatively formal development of a specific repertory grid test for users of M.I.S.s. This form of the test is the basis of the final and more flexible test format used in Chapter 10.

9.2.1 Elicitation of Elements (ie Sample Constructs)

The questions used to elicit these elements were grouped into categories describing the areas within the subject's construct system to be investigated (see Table 9.1). They were also presented to the subject in this order. The subject was not explicitly told that the questions had been grouped by common theme, although it was recognised that he might perceiv e this. The reason for this procedure was that once the subject had focussed on some internal sub-system of constructs, all the elements required from that system would be elicited at the same time. It was assumed that the subject might otherwise experience confusion if he were asked to repeatedly focus and re-focus on different parts of his construct system.

9.2.2 Elicitation of Super-ordinate Constructs

As 9.1 explained, super-ordinate construct elicitation is carried out by a process of triad sorting. It was anticipated that the sorts could be pre-determined by a consideration of the questions used to elicit the elements. Figure (9.2) shows schematically how this was to be done. The categories C1 - C4 relate to those set out in Table (9.1). The symbols Q1, Q2 etc represent the elements obtained from questions 1, 2, 3 etc in each of the categories C1 - C4 of Table (9.1). The three circles drawn on each row of the grid indicate the elements to be used in the triad sort for that particular row. In this way it was hoped to elicit constructs that showed how the subject mentally perceived the relationships between each category, and at a finer level of resolution, how he related various aspects within each category.

It was felt however that if the grid were laid out in the form shown in fig (9.2) then the subject might readily perceive the nature of the relationships being investigated. Under these circumstances he might feel tempted to give responses that he thought were in some way expected of him.

Table (9.1) QUESTIONS USED TO ELICIT ELEMENTS IN FIRST FORM OF REPERTORY GRID TEST

CATEGORY	QUESTION
(1) Analysis of job	1. What is the most important task in your job? 2. What is the most difficult task in your job? 3. What is the most interesting task in your job? 4. Which task in your job do you consider could most likely be done by computer if it were suitably programmed? 5. Which task do you consider least likely to be capable of being done by a computer? 6. Is there a calculation or series of calculations you regularly have to perform that is particularly tedious? 7. What is the most boring task you have to perform? 8. Which task or aspect of your job would be most difficult to explain to an outsider? 9. Which task or aspect of your job would be easiest to explain to an outsider? 10. What do you regard as the best way to communicate factual data at work? 11. What is the most important information you receive? 12. What is the most important information you generate?
(2) Impressions of Computer/Management Techniques	1. What is the best application of computers you have heard of? 2. What is the best application of computers you have had personal experience of? 3. What is the best management technique you know of? 4. Give an instance of an application of computers you consider least likely to be useful. 5. Is there an application or area of your job which is particularly unstructured? 6. Indicate some quantity or figure which most exemplifies in your own mind, the subject of statistics.
	1. Give an example of an application or task in industry at which computers (are or would be) superior to a human.

Table (9.1) Contd.

CATEGORY	QUESTION
(3) Humans in relation to computers	2. Give an example of the reverse situation in industry i.e. where a human would be superior to a computer. 3. If you had a job to do at work that you knew could be done by computer would it be easy or difficult to get the job done? Ques. What particular aspect of getting the job done would you expect to be most easy/difficult?
(4) Relevant Human characteristics in the work situation	1. What is the attribute you associate with the most able manager you know? 2. What is the attribute you associate with the least able manager you know? 3. What is the attribute you associate with the best sub-ordinate you have had? 4. What is the attribute you associate with the worst sub-ordinate you have had?

THE SELECTION OF TRIADS IN VARIOUS CATEGORIES

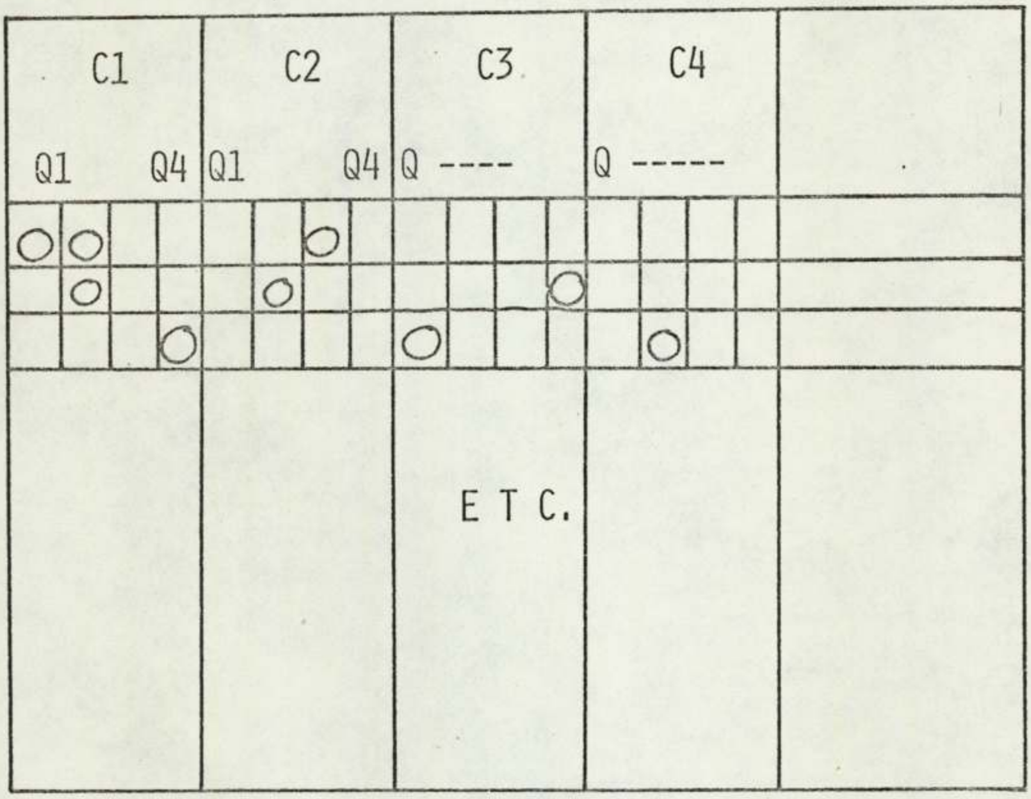


FIG (9.2)

The problem of reaching beyond formal procedures to obtain the real-life situation was discussed in 7.5. In order to reduce the possibility of the subject perceiving any such structure in the grid and its associated triad sorts, after the sorts had been chosen the columns and rows of the grid were randomised.

Section 9.1 has discussed the fact that more unique triads of elements may be created than are needed to elicit the required number of constructs. Furthermore the procedure described above was ultimately discontinued. For these reasons the particular sorts chosen in this stage of test development have no special significance and are not presented.

9.2.3 Test Administration Procedure

It was anticipated that the test could be carried out in one session, with the possible reservation that the subject might be permitted to complete the scoring of the elements against the constructs on a separate occasion. Thus the questions would be asked in the element elicitation stage and the elements themselves would be entered directly on to a grid upon which the triad sorts had been previously marked. When the elements had been obtained the subject could proceed directly to the construct elicitation stage based upon the triads indicated, and finally to the scoring stage.

However it was recognised that the whole procedure would be completely unfamiliar to the subject and it seemed reasonable to suppose that the test would run more smoothly if the subject could be given an overview before he commenced his own test. For this reason an example case was devised but relating to a different subject area to avoid the possibility of pre-empting the actual test.

9.3 INITIAL FINDINGS, DISCUSSION AND INTERIM CONCLUSIONS

Having devised the test procedure it was necessary to try it out on a number of suitable subjects. Ideally these subjects should have been practising managers holding positions comparable with the ones upon whom the test was ultimately to be used in industry. However subjects like this are difficult to obtain and it was not wished to expose the managers in the sponsoring organisation to the test until it was in its final form. It was decided therefore to give the test to a non-managerial subject since this would give an opportunity to assess the administration procedure, while the results would give an indication of how specific the test form had to be to a particular type of job in order to give useful results.

Two subjects were found who were willing to carry out the test and who were well known to the author. This last fact would enable a comparison to be made between the results of the test and the author's personal assessment of the subjects' attitudes to their work. Obviously this would furnish no objective results but would be a useful guide in assessing the test at this early stage.

In carrying out the test a number of procedural difficulties became evident.

Considering first the elicitation of elements, the subjects sometimes wished to give similar answers to the various questions used to obtain elements. For example, the most important task might be regarded as the "most satisfying task". This immediately affected the pre-selected triad sorts, some of which had to be discarded in order to avoid duplication.

A second problem was concerned with the size of the grid which was drawn on an A3 size (11¾'x16½")piece of paper. This size of paper is the largest that can readily be copied using conventional office facilities which thus avoids reproduction problems. This paper size is also about the largest that can comfortably be manipulated by a subject seated at a desk. Its drawback was that to enter the elements the grid had to be turned on its side in order to have sufficient room to make the entry and even when this had been done the space available was limited. This can be seen from the completed grids (Figs. 10.1-3, 10.5-7, 10.9-10). One of the two initial subjects complained about the difficulty of scoring the elements when they were perpendicular to the body of the grid.

It was decided therefore that the element elicitation questions would be better given at a straightforward question-and-answer session between subject and experimenter, with the experimenter taking a written note of the subject's answers. The resulting elements could then be written on to the grid by the experimenter at a later stage. This also solved another small practical problem, since the subjects would sometimes think of a more appropriate element in response to a previous question during the element elicitation stage, necessitating amendment of a grid element entry. This tended to make the grid look untidy and added to the problem of legibility already mentioned.

A consequence of deciding to enter the elements on the grid at a later stage was to break up the test between the element elicitation stage and the construct elicitation stage. In fact this was quite convenient as it became obvious during the trial tests that the full test would take too long to administer in one session. Experience showed that element elicitation would take approximately 30 mins, construct elicitation 45-60 minutes and element scoring 40-50 mins.

However in the final form of the test there was another important reason for separating the element elicitation and construct elicitation stages. In fact this modification of the procedure was developed as a result of administering the test at the sponsoring Company in two further pilot tests carried out before approaching the practising manager. Since this was the only change in procedure subsequently introduced it is convenient to mention it here. It was found that during the eliciting of constructs, which is regarded as the most difficult part of the test, the subject was often having difficulty keeping the elements concerned in the triad sorts in his "mind's eye". The subject would tend to ask questions like "what was the third thing again?" or "what were the other things?" To overcome this problem, the elements were typed separately on to cards. Thus each triad sort was presented to the subject on three separate cards.

Not only did this help him to simultaneously visualise the elements concerned, but it also appeared to help him verbalise the constructs. It was found helpful at the construct elicitation stage to suggest to the subject that he arranged the cards so that two of the elements were similar and yet opposite in some sense to the third, before he tried to express why they were perceived in that way. This is consistent with Kelly's suggestion that some constructs might be pre-verbal. In such a case, forcing the subject to label these verbally may introduce a degree of inaccuracy but, providing he recalls the construct he had in mind at the time he scores the construct against the elements, this does not matter. The construct label is thus seen as an aide-memoir or an intervening variable in the analysis.

Even with the aid of the element cards, subjects found the construct elicitation stage difficult and would discuss the nature of a "right" answer with the experimenter. His task was to establish whether the qualities of contrast were contained in the subject's differentiation of the elements in each triad. This was particularly difficult in cases where the construct concerned was of the "X - not X" variety. The latter situation is discussed in section 8.13.

Because the experimenter had to be alive to the above problem it again was impractical to permit the subject to enter his responses directly on to the chart. The subject's responses were recorded on paper by the experimenter and subsequently copied on to the grid. This was then given to the subject to score in the manner described in 9.1.

A final point on the test administration procedure is concerned with the order in which the element elicitation questions were asked. It became evident that there was no need to ask questions within categories at the element elicitation stage since subjects had no difficulty in responding to the questions whatever the order of presentation.

Except where stated otherwise the foregoing changes to the test administration procedure were carried out after giving two tests in the manner described at the start of the present section. Detailed discussion of these tests is deferred until Chapter 10 where the results of all the tests given are presented and discussed. However the interim conclusions drawn from the first two tests are presented here in anticipation of the final test procedure described in section 10.1 which is based on these.

The interim conclusions were as follows:-

(i) As anticipated, the questions for element elicitation had not been altogether relevant to subjects engaged in university research occupations and to this extent the scope of the test had been reduced.

(ii) Despite the observations of (i) above, the test had produced results consistent with the experimenter's expectations of one subject's responses and in accordance with its hypothesised performance with the second subject (see 10.1.1 & 10.1.2).

(iii) "Object elements" such as the "sales ledger" and "Personality elements" such as "Honesty" jointly fall within the range of very few super-ordinate constructs and comparisons between these categories of elements in the form of triad sorts are unfruitful. Thus triad sorts of elements drawn from categories 1-3 of Table 9.1 are not productive in eliciting constructs when combined with elements from category 4.

(iv) "Object Elements" have fewer constructs associated with them than do "Personality Elements" ie they are constructs of lower order.

(v) Because of (iii) and (iv) the present test form will not cover most of the aspects of the subject's job relating to interaction with other persons. Thus if resistance to implementation arises from inter-personal considerations, then the test in the form described above will probably not be of much use.

9.4 THE FACTOR ANALYSIS OF REPERTORY GRIDS

Factor Analysis considered as a methodology, rather than as one of the many techniques loosely given this name, is particularly useful in the understanding of repertory grids. Section 9.4.1 describes how a repertory grid may be interpreted in geometric terms and explains how it may be simplified by factor analysis. Section 9.4.2 explains the basic principles of factor analysis and distinguishes between the basic approaches. Specific terms used in the discussion and interpretation of the grid in chapter 10 are introduced. Finally the actual analysis procedure used in this study is explained in section 9.4.3.

9.4.1 Geometric Interpretation of a Repertory Grid

A completed repertory grid may be simply regarded as a 'mxn' data array where 'm' is the number of constructs elicited from 'n' elements. The row entries could then be regarded as defining the position of 'm' construct vectors of an 'n' dimensional element space. If the constructs are not independent then these vectors could be represented in a space of reduced dimension by defining new reference axes that are independent (ie orthogonal) and then resolving the original vectors into independent components along the new axes defining the reduced space.

Ignoring for a moment the mathematical and statistical complexities, the purpose is to understand how the constructs elicited from the elements are themselves inter-related, if at all. If two or more constructs have been given effectively similar scores on the same set of elements it is obviously important to detect this situation, since the constructs must be closely related even though they have been given different names. In a large matrix of numbers such as that obtained in a completed grid such relationships are difficult to pick out by inspection and factor analysis techniques provide one systematic way of doing this. Thus by judicious grouping of the elicited constructs it is sometimes possible to establish more super-ordinate constructs that are, in contrast to the former, quite independent. This has been done for the grids described in Chapter 10.

It is also possible in general to re-scale the co-ordinates of the original construct vectors to correspond to the new reference axes (vectors) and in this way graphical plots may be drawn to visually demonstrate the derived relationship. However this has not been done in the present study, partly for technical reasons associated with the computer software package used and partly for theoretical reasons described in 9.4.3.

9.4.2 Factor Analysis

A large number of specific techniques exist for those who wish to apply the general principles of factor analysis and the question of which to choose has not always an easy one. Harman (1967) quotes Cureton on this subject as follows,

> "Factor theory may be defined as a mathematical rationalisation. A factor-analyst is an individual with a peculiar obsession regarding the nature of mental ability or personality. By the application of higher mathematics to wishful thinking, he always proves that his original fixed idea or compulsion was right or necessary. In the process he usually proves that all other factor-analysts are

dangerously insane, and that the only salvation for them is to undergo his own brand of analysis in order that the true essence of their several malodies may be discovered. Since they never submit to this indignity, he classes them all as hopeless cases, and searches about for some branch of mathematics which none of them is likely to have studied in order to prove that their incurability is not only necessary but also sufficient".

Cureton (1964)*

However the theory is now relatively well developed and many of these earlier problems have disappeared.

The starting point for a factor analysis is a table listing variables $V_1 \ldots V_n$ whose values at equivalent times are given by each of the rows $1 \ldots m$. This is shown in fig. (9.4). Typically results a_{ij} for $i = 1, \ldots m$ are obtained from the value of variables V_j for $j = 1, \ldots n$ in the ith trial in a series of 'm' experiments. The purpose of factor analysis is to attempt to explain the observed variation between the variables V_j in terms of a smaller number of hypothetical, usually independent variables, or factors as they are known, $F_1, F_2 \ldots F_l$ where $l < n$. In practice some meaning is often ascribed to the factors depending on the variables associated with each.

The foregoing may be expressed more formally by representing the actual variables V_j as linear combinations of the factors F_k where $k = 1, \ldots l$, thus yielding,

$$V_j = w_{j1} F_1 + w_{j2} F_2 + \ldots w_{jl} F_l \ldots\ldots \text{Equn. } (9.1)$$

where w_{jk} ($k=1, \ldots l$) are coefficients known as factor loadings.

However two fundamentally different objectives can be pursued in attempting to derive the factors, namely, (a) reduction of variance and (b) best reproduction of correlation between the original variables.

* See Harman (1967)

TABLE OF VARIABLES SUITABLE FOR FACTOR ANALYSIS

V_1	V_2	V_3	- - -	V_j	- - - - - - - - - - - -	V_n
a_{11}						a_{1n}
a_{21}						
a_{31}						
a_{i1}				a_{ij}		
a_{m1}						a_{mn}

FIG (9.4)

The reduction of variance approach uses the basic relationship already given in Equ. (8.1) where it can be seen that the variance associated with a particular variable is shared across all the factors. The factors may be regarded as common factors. The particular technique usually associated with this approach is that of Principal Component analysis.

However the classical factor analysis model assumes that only part of the variance is shared among the common factors, while the remaining variance associated with each variable is ascribed to a specific factor S'_j and an error factor E_j each of which is independent of the other factors. This model yields the following relationship

$$V_j = w_{j1} F_1 + w_{j2} F_2 + \dots w_{jl} F_l + b_j . S'_j + e_j . E_j \quad \text{Equ. (9.2)}$$

where b_j & e_j are coefficients

In practice it is unusual to be able to separate the specific and error effects and these are combined into the specific factor U_j, giving,

$$V_j = w_{j1} F_1 + w_{j2} F_2 + \dots w_{jl} F_l + d_j U_j \quad \text{Equ. (9.3)}$$

where d_j is a coefficient.

The total variance s_j^2 associated with V_j, which is standardised, may be expressed as follows,

$$s_j^2 = 1 = h_j^2 + (b'_j)^2 + e_j^2$$

$$= h_j^2 + d_j^2$$

$$\text{where} \quad h_j^2 = w_{j1}^2 + w_{j2}^2 + \dots w_{jl}^2$$

$$= \text{common variance}$$

Given the above account of the two basic models of factor analysis, the actual process may readily be understood in geometrical terms. If in the table of results depicted in Fig. 9.4 the entries for each variable are standardised, then the individual entries may be regarded as describing a vector of unit length existing in a space whose dimensionality is potentially 'm', the number of trials.

Using the geometry of n-dimensional space it is possible to show (see, for example, Harman 1967) that such vectors are separated from one another by angles whose cosines are equal numerically to their Pearson's product moment correlation coefficients. This is shown in two dimensions in fig. (9.5).

In the simple case of fig. (9.5) where all the variable vectors fall in one plane, the four vectors can be resolved into components on two independent axes F_1 & F_2 so producing a complete explanation of 4 variables in terms of only two hypothetical factors. This is the purpose of factor analysis, although in practice the vectors would seldom lie in one plane. Usually some degree of approximation is required before the variables may be represented by a smaller number of factors.

9.4.3 Discussion & Selected Procedure

The starting point of a factor analysis is to compute the correlation coefficients between variables. Factor analysis theory requires that the Pearson product moment correlation coefficient be used but it is only sensible to use this when the initial data can be represented on at least an interval scale. However it is hard to see how the scores in a repertory grid may be regarded as being picked off an interval scale. Instead it seems prudent to regard these scores as being only ordinal in character. Under such circumstances a rank order correlation coefficient must be used to correlate the variables.

Kendall (1948) has discussed a number of rank order correlation coefficients and prefers in a situation similar to that occuring in repertory grid where it is desired to correlate the constructs, to use a correlation coefficient of his own derivation, the 'tau - τ' coefficient. However, despite the suggested theoretical superiority of this coefficient, another coefficient 'Spearman's rho - ρ' was used in this application. The basic reason for this choice was the fundamental similarity between the derivation of Spearman's rho and Pearson's product moment coefficient

GEOMETRIC REPRESENTATION OF VARIABLES IN FACTOR ANALYSIS

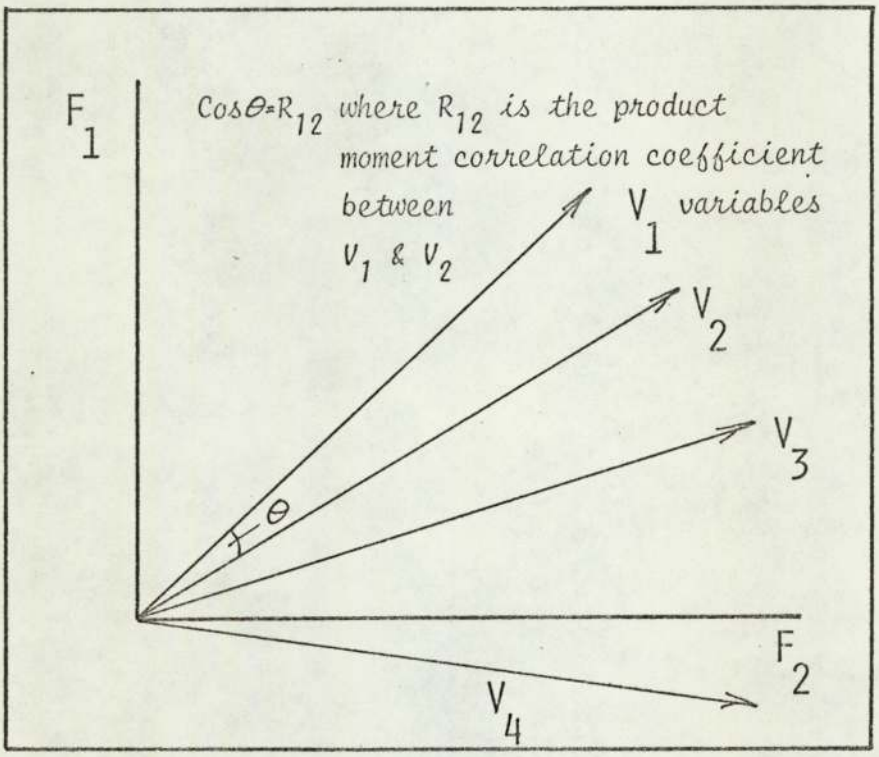

F_1
$\cos\theta = R_{12}$ where R_{12} is the product moment correlation coefficient between V_1 & V_2
V_1 variables
V_2
V_3
θ
F_2
V_4

FIG (9.5)

'r'. In fact Kendall (1948, chapt. 2) shows them to be special cases of an expression for a generalised correlation coefficient R,

$$R = \frac{\sum xy}{\sqrt{x^2 . y^2}}$$ where the x & y are individual attributes of the variables X & Y

If the 'x' & 'y' are deviates from their respective means, the above expression yields Pearson's product moment correlation coefficient. If however the results or readings derived from a test are equal to their ranks (so that a reading value 1 has a rank of 1 & a reading of 2 has a rank of 2 etc) then the expression for the calculation of Spearman's rho gives the same numerical result as obtained with the same data with the product moment correlation coefficient. If this were universally true the discussion about the type of scale to which the scores obtained in the grids belonged would be academic. Unfortunately however if tied ranks are obtained the numerical correspondence between the two coefficients disappears. Despite this, Spearman's rho was still used in preference to Kendall's tau coefficient because of its fundamental relationship with the product moment coefficient.

The product moment coefficient has great significance in the theory of factor analysis and its properties are utilised in deriving the fundamental relationships, for example, that $r = \text{Cos}\,\theta$ where θ is the angle between two variable vectors as discussed in the previous section. If, however, another coefficient of correlation is used, these relationships are no longer appropriate. On the other hand it is equally inappropriate to use a coefficient that will indicate spurious relationships by assuming interval scaling, simply in order to proceed with the analysis.

A formal description of the procedure adopted therefore is as follows. Firstly the relationships between the constructs were calculated using the rank order correlation coefficient, Spearman's rho.

New constructs were then defined having the same labels or names as the original constructs but which were inter-related in such a way that their product moment correlation coefficients were numerically equal to the rho coefficients of the corresponding, original constructs. The vectors representing the newly defined constructs could not be represented in terms of their co-ordinates because the explicit form of the transformation is indeterminate by virtue of the ordinal relationship. However they were defined to be of unit length having co-ordinates in standardised form. The factor analysis (or principal components analysis) was then carried out using the transformed vectors.

The two approaches to factor analysis described in section 8.4.2 are quite different in principle, in addition to their different ways of treating the variance associated with a particular variable. In the classical factor analysis approach a linear model is fitted to the data, but before this can be done the experimenter must decide how many factors need to be fitted, that is he must decide the dimensionality of his model. However the method of Principal Components (reduction of variance approach) proceeds by fitting the first factor (known as a principal component in this case) to the data in such a way as to explain the most variance. The next axis is then placed orthogonally and positioned to account for the greatest amount of the remaining variance. To explain the greatest possible amount of variance the fitted axis, or component, must subtend as small an angle θ as possible with the variable vectors. In this way $\cos\theta$ is maximised and since $r = \cos\theta$ r too is maximised. However the explained variance is given by r^2 and so by minimising θ the variance is maximised. The benefit of this approach is that an invariant solution is obtained. In classical factor analysis no single orientation is 'best' and the actual orientation obtained is determined by the particular technique used.

In practice many researchers subsequently rotate the axes initially obtained to obtain a better "fit". A drawback to the method of Principal components is that the analysis ultimately yields as many components as the original number of variables. However the first few components derived explain most of the variance and hence the remaining components are usually ignored.

Slater (1965) favours the invariant principal components solution for grid analysis but in the present study factor analysis was preferred. In fact the procedure adopted was to first carry out a principal components analysis to suggest how few factors were required to explain most of the variance and then to carry out a factor analysis based on Lawley's method of maximum liklihood (see Harman, 1967 Chapter 10) to extract this number of factors. The factor analysis model was preferred because it does not assume that all the variance associated with a particular variable must be distributed in the common factor space, thus permitting some constructs to have a high specific variance (d_j^2 - Equ. 9.3). This seems to avoid the statistical straitjacket of principal components and is more compatible with personal construct theory.

It is possible in general to recompute the co-ordinates of the original vectors relative to the new common reference axes (factors). However because of the ordinal nature of the construct scores this could not be justified in the present study.

10. RESULTS OF REPERTORY GRID TESTS

This chapter discusses the results of the grid tests administered during the research and presents the relevant findings. In the first two tests the main pre-occupation was with establishing the test and analysis procedures, and for this reason the discussion is largely confined to procedural aspects rather than particular grid findings - although these are not ignored. The results of the first two tests are presented in 10.1. A revised element and construct elicitation procedure was introduced after the first two tests and was used in the remaining tests. This is described in 10.2. Also presented in 10.2 are some general comments and findings from other tests used to polish the presentation of the test procedure. The three tests described in 10.2 were administered to practising managers not associated with the C.A.D. function although two of the managers were from the sponsoring Company. The penultimate section (10.3) presents the results of the tests with two of the successive managers in C.A.D. which is followed by discussion (10.4).

All the managers at the sponsoring Company gave their consent for the results of their tests to be published in a research thesis. It was explained that the results of the tests would not be available in any other way and in particular that they would not be discussed with other members of the Company.

In analysing the grids using various computer programs it was necessary to identify the grids by codes. In the following sections these codes have been retained and thus the tests are referred to by the names KGRID 1, KGRID 2 etc.

10.1 THE INITIAL TESTS

Since the repertory grid test procedure requires the subject to score every element according to a particular construct, inspection of the completed grid can indicate areas requiring further investigation or clarification directly. However in the initial tests the subject's scoring of the elements is of little relevance to the present study except insofar as it exemplifies the procedure to be employed in the general analysis of a grid. Consequently tests KGRID 1 & KGRID 2 concentrate mainly on analytical aspects rather than content.

10.1.1 Test KGRID 1

Although inspection of a grid can yield insight directly, it is difficult to be systematic in such an approach and thus to be certain of noting all the observable information. Furthermore, the constructs elicited from the subject may be inter-related, additionally complicating a simple analysis as discussed in 9.4.1 where it was explained how factor analysis could simplify the interpretation of correlated constructs.

In analysing the first repertory grid, interest focussed on comparing the results obtained using the method of analysis selected for the present study with those obtained from alternative methods. The effect of extracting different numbers of factors was also examined.

Analysis of the grid commenced by calculating Spearman's 'rho' rank correlation coefficient between all the constructs elicited using program RHØ (see appendix P6). The results obtained are presented in Table (10.1) where they may be compared with Pearson's product moment correlation coefficient calculated for the same data considered as being on an interval scale.

As is to be expected, differences are found between the two coefficients although in general these are relatively small. The greatest differences are to be seen in the correlations involving constructs 2 & 6. As described in 9.4.3, the factor analysis procedure adopted postulated the existence of new construct vectors, related by Pearson product-moment correlation coefficients equal numerically to the Spearman's rank correlation coefficient actually

COPY KGRID1

test no. 1R date 8.3.71
15.3.71
22.3.71

notes:- Elements elicited on 8th March. Most of constructs elicited on 15th March and remainder on 22nd March.

1.Understanding errors on computer runs	2.	3.Numerical control	4.Cyclic strain hardening exponents	5.Communication with supervisor	6.Retrospective thinking	7.Resentful of my position	8.Void linkage	9.	10.Mand tests	11.Fractional factorial experiment	12.Conscientiousness	13.	14.Crack propagation characteristics	15.Current literature	16.Clear assessment of situation	17.Personnel selection	18.Critical Path Analysis	19.Delta 'K' calculation	20.Automatic stores control	21.Stereoscan work	22.	23.Computing specimen area	24.Photographic evidence	25.Structural deformation	side entry A	side entry B
-2		+2	+1	-2	-2	-2	0		+2	0	-2		0	0	-2	-2	+1	+2	+2	+2		+1	0	0	1.Machine	Man
+2		+2	+2	+2	-2	-2	0		+2	+2	+2		+2	+2	+2	+2	+2	+2	+2	+2		+1	+2	-2	2.Useful	Not useful
-2		+2	-2	-2	-2	-1	+1		+2	+1	-2		+1	-2	-2	-1	-1	-2	+2	+2		-1	-1	0	3.Practical	Intellectual
-2		-2	+1	0	0	0	+1		+2	+1	0		+1	-2	0	0	0	0	0	+2		-1	+1	+1	4.Experimental	Not experimental
+2		-2	-2	+2	+2	+2	-2		-2	-2	+2		-2	0	+2	+2	-2	-2	-2	-2		-2	-2	-2	5.Human	Not human
+2		-2	+2	+1	0	0	+2		+2	+1	+2		+2	+1	+1	-2	-2	+2	-2	+2		+2	+2	+2	6.Job	Not job
+1		-2	0	0	0	0	0		0	+1	0		0	+2	+1	-1	-2	0	-2	0		0	0	+2	7.Academic	Industrial
0		+2	+2	0	0	0	0		0	+1	0		0	0	0	-2	+1	+2	+2	0		-1	0	0	8.Computer applications	Not computer applications
+2		-2	-2	+2	+2	+2	-2		-2	-2	+2		-2	0	+2	+2	-2	0	-2	0		0	0	0	9.Human factors involved	
+2		-2	+2	0	0	0	-2		-2	+1	0		-1	+2	0	0	+1	+2	-2	-2		+2	-1	0	10.Theoretical	Practical
-2		+2	+2	-2	-2	-2	0		0	+1	-2		0	0	-2	-2	+2	+2	+2	0		+1	0	0	11.Computer aspects	Human aspects

FIG (10.1)

Table (10.1) PEARSON'S PRODUCT MOMENT AND SPEARMAN'S RANK CORRELATION COEFFICIENTS FOR CONSTRUCTS OF KGRID 1

Construct Nos. → ↓	1	2	3	4	5	6	7	8	9	10	11
1											
2	.28 .25										
3	.62 .58	.08 - .01									
4	.22 .24	- .09 - .02	.39 .45								
5	- .89 - .86	- .19 - .09	- .62 - .64	- .37 - .47							
6	- .04 .12	- .03 .04	- .17 - .08	.35 .45	- .13 - .29						
7	- .33 - .37	- .25 - .17	- .34 - .36	.04 .07	.19 .21	.70 .47					
8	.55 .54	.20 .33	.20 .12	- .04 .02	- .46 - .45	- .16 - .20	- .31 - .26				
9	- .80 - .76	- .27 - .20	- .67 - .64	- .29 - .39	.89 .88	.07 - .06	.32 .29	- .55 - .57			
10	- .22 - .23	.05 .01	- .76 - .73	- .44 - .44	.21 .23	.20 .09	.40 .41	.00 .02	.27 .27		
11	.89 .87	.30 .22	.42 .40	.06 .09	- .88 - .86	- .15 - .02	- .38 - .35	.71 .71	- .85 - .83	.02 .05	

Key

A B

A = Pearson's product moment correlation coefficient.
B = Spearman's rank correlation coefficient

computed between the constructs. It seemed worthwhile to investigate whether the derived Pearson product-moment coefficients (r) would differ significantly from those computed directly from the scores when the latter were considered to be on interval scales. The results of the comparison which was carried out using Fisher's Z transformation - Moroney (1956, p 312-14) - are presented in Table (10.2). It can be seen that the differences between the transformed correlation coefficients is not more than one standard error of the difference ($Z_1 - Z_2$), which may be assumed to be distributed normally. Thus the correlation coefficients are not significantly different although different numeric results will be obtained from each of course. If the above finding is generally true, which seems not unlikely, the use of the product moment correlation coefficient computed directly from the ordinal scores may be justified when computational facilities for calculating Spearman's rank correlation coefficient are not available.

In order to decide how many factors to extract in the factor analysis, Kaiser's criterion (Kaiser, 1960) was applied. This requires that a principal components analysis is carried out and the number of eigenvalues greater than unity of the correlation matrix (having ones in the leading diagonal) is found. This number is then taken as the number of factors to be extracted. The eigenvalues for KGRID 1 are shown in Table (10.3) which shows that three factors should be extracted. In fact 4 factors were extracted since the fourth eigenvalue was close to unity and in the initial enthusiasm it was feared that potentially valuable information might be lost.

The loadings for a principal components solution (first 4 components), a four factor solution and a two factor solution are shown in Table (10.4). It was interesting to compare the four factor solution with the principal components solution since Slater (1965) favours the latter.

Table (10.2) COMPARISON OF CORRELATION COEFFICIENTS GIVEN IN TABLE (10.1) FOR SIGNIFICANT DIFFERENCE USING FISHER'S Z - TRANSFORMATION

Construct Nos.→ ↓	1	2	3	4	5	6	7	8	9	10	11
1											
2	.10										
3	.19	.27									
4	.06	.21	.22								
5	.39	.31	.10	.37							
6	.48	.21	.28	.36	.50						
7	.14	.25	.07	.09	.06	1.08					
8	.04	.42	.25	.18	.04	.12	.16				
9	.31	.22	.16	.34	.14	.39	.10	.09			
10	.03	.12	.20	.00	.06	.34	.04	.06	.00		
11	.27	.26	.07	.09	.25	.40	.10	.00	.21	.09	

Method: $Z = 1.15 \log \frac{(1 + r)}{(1 - r)}$

Let Z_1 and Z_2 be the transforms of the product moment correlation coefficient and rank correlation coefficient respectively. Then $(Z_1 - Z_2)$ may be assumed to be distributed normally with standard error, $\sqrt{\frac{2}{N - 3}}$ where N=21 is sample size

The entries in the body of the above table are calculated from,

$$\frac{Z_1 - Z_2}{\sqrt{\frac{2}{N - 3}}} = 3 (Z_1 - Z_2)$$

Table (10.3) EIGENVALUES GREATER THAN UNITY FOR RANK CORRELATION COEFFICIENTS OF TABLE (10.1)

Component No.	1	2	3
Eigenvalue	4.79	1.96	1.73
% of Total Variance	43.53	17.78	15.75
Cumulative % of Total Variance	43.53	61.31	77.06

Table (10.4) PRINCIPAL COMPONENTS ANALYSIS & FACTOR ANALYSIS OF KGRID 1

Construct No.	Component/Factor No.											
	1			2			3			4		
	C	4F	2F	C	4F	2F	C	4F	2F	C	4F	2F
1	41	89	90	06	- 08	- 12	10	- 16		05	- 08	
2	11	19	20	27	23	19	07	- 22		- 90	- 87	
3	33	49	49	- 27	- 52	- 56	- 33	- 58		07	13	
4	19	21	20	- 52	- 74	- 77	06	- 02		- 24	- 14	
5	- 42	- 93	- 88	12	36	35	- 19	- 01		- 15	00	
6	02	08	05	- 43	- 58	- 52	50	59		- 16	- 26	
7	- 20	- 32	- 33	- 25	- 22	- 16	49	61		02	- 08	
8	28	65	66	38	42	36	13	- 14		- 07	- 12	
9	- 42	- 87	- 87	00	19	23	- 08	18		- 06	05	
10	- 18	- 04	- 04	34	55	59	53	74		19	- 04	
11	40	98	95	25	16	13	20	00		20	00	

Note: Decimal point omitted - all table entries X10 $^{-2}$

C = Component Loadings
4F = Factor Loadings with four factors extracted
2F = Factor Loadings with two factors extracted

In fact the four factor solution gives significant loadings on all the variables having significant loadings in the principal components analysis, with additional loadings in the first and second factors extracted. In deciding whether a loading was significant, the simple procedure of rejecting as insignificant all loadings less than 0.3 was applied. The rationale behind this was that any loading less than 0.3 explained less than 10% of the common variance on a given factor. Although more rigorous tests are available for computing the statistical significance of factor loadings - see, for example Burt and Banks (1947), Holzinger and Harman (1941) - the approach adopted seemed consistent with the relatively low order of measurement (ordinal) obtained in the basic data. In each case the loadings are greater in the four factor analysis compared with the principal components analysis, suggesting that the factor analysis is a better discriminator between variables.

The two factor analysis was carried out after a more detailed examination of the four factor results discussed below. It seemed possible that only two factors might adequately describe the data if extracting only two factors caused the variation in the common factor space to be redistributed. As the following discussion will show, this would probably have been an oversimplification. However it is mentioned here because, in the event, the loadings on the two factors extracted were very similar to the loadings on the first two factors of the four factor solution obtained initially. Thus a redistribution in the common factor space had not been achieved, instead the common variance associated with the third and fourth factors in the four factor solution had merely been added to the specific variance as Table (10.5) shows.

Although the information elicited in this particular test has no relevance to the present study, it is appropriate to consider the factors derived as these gave an early indication of the kind of results that were

Table (10.5) SPECIFIC VARIANCE ASSOCIATED WITH CONSTRUCTS IN TWO AND FOUR FACTOR ANALYSES

Construct No.	Four Factors		Two Factors	
	Common Variance	Specific Variance	Common Variance	Specific Variance
1	.84	.16	.81	.19
2	.86	.14	.08	.92
3	.87	.13	.54	.46
4	.62	.38	.63	.37
5	1.00	.00	.98	.02
6	.76	.24	.27	.73
7	.53	.47	.13	.87
8	.63	.37	.57	.43
9	.83	.17	.81	.19
10	.85	.15	.35	.65
11	1.00	.00	.99	.01
	79.9%		56.0%	

to be expected from tests completed by managers. To this end the factor loadings have been arranged in descending order in Table (10.7) and the signs reversed where appropriate. (Reversing the sign of all the loadings of a bi-polar factor simply reverses the label of the factor).

The first factor in Table (10.6) may be termed a "man-machine" bi-polar factor. The construct "practical/intellectual" loads on the machine (computer aspects) end of the scale and is consistent with the subject's use of the computer as a practical tool for the analysis of results.

The second factor is bi-polar and at first sight appears to have "Theoretical - Experimental" as its poles. However further examination of the constructs loading on to this factor suggests that theoretical has a rather special meaning. Thus "human" and "computer application" are associated with "theoretical". Upon examining the grid element scores for "Human/Not human" it is found that positive scores (relating to the "human" pole of the construct) are given to elements such as "personnel selection" (judgement) and "retrospective thinking" (reflection). Furthermore the construct "computer application", which also loads on the "theoretical" pole of the factor, is used in the context of computational aspects rather than machine aspects. The construct "Theoretical/Practical" subsumes under its theoretical pole, elements such as "Delta 'K' calculations" (a fracture mechanics term) and "understanding errors on computer runs". Used in the above way the three constructs may be collectively described as "cerebral activities" while the other pole of the factor may be described as "practical activity" giving a "Cerebral-Practical" bi-polar factor. It is interesting to note that "Job/Not Job" loads on the "practical" pole of the factor.

The latter observation is even more interesting when in factor 3, the construct "Job/Not Job" loads on the factor pole containing the

Table (10.6) CONSTRUCTS HAVING SIGNIFICANT LOADINGS IN KGRID 1

Factor No.	Construct No.	Construct	Factor Loading	Specific Variance
1	11	Computer aspects/Human aspects	.98	.00
	1	Machine/Man	.89	.16
	8	Computer Application/Not Computer Application	.65	.37
	3	Practical/Intellectual	.49	.13
	7	Academic/Industrial	- .32	.47
	9	Human factors involved/	- .87	.17
	5	Human/Not human	- .93	.00
2	4	Experimental/Not experimental	.74	.38
	6	Job/Not Job	.58	.24
	3	Practical/Intellectual	.52	.13
	5	Human/Not Human	- .36	.00
	8	Computer application/Not computer application	- .42	.37
	10	Theoretical/Practical	- .55	.15
3	10	Theoretical/Practical	.74	.15
	6	Job/Not job	.59	.24
	7	Academic/Industrial	.61	.47
	3	Practical/Intellectual	- .58	.13
4	2	Useful/Not useful	.87	.14

constructs "Theoretical/Practical" and "Academic/Industrial". This factor may be termed "Theoretical-Practical" by expanding the meaning of theoretical from its narrow use within the grid, to its wider more colloquial use. In this dimension the job is seen to have a theoretical or academic bias as distinct from a practical or industrial flavour.

The fourth factor contains only one construct "Useful/Not Useful". Since it is not associated with any other construct there are no clues to indicate whether the words have their usual meaning. Assuming they do, it may be significant that the construct "Job/Not Job" does not load on to it. It will be remembered that by Kaiser's criterion only three factors should have been extracted. Had this been done, and assuming the same effect on the factor loadings as was observed when only two factors were extracted, then the construct "Useful/Not useful" would not have appeared in the analysis.

In summary, the above results were encouraging. Some procedural difficulties had been exposed as described in chapter 9, but a grid had been obtained and analysed. Based on a purely subjective assessment the results had given a fair picture of the subject's job and his attitude toward it. There was every reason to suppose that the test would be even more successful when applied to the kind of subject for which it had been designed. The subject of KGRID 1 was employed as a University experimental officer and spent his time upon largely experimental work in the field of fracture mechanics. The test had been able to expose a small number (3, perhaps 4) of super-ordinate constructs about which the subject structured his attitudes toward work.

Before making changes to the test it was decided to carry out a further test using exactly the same procedure. This would give an indication of whether the difficulties encountered in administering the first test were due entirely to the test procedure or whether they were to some extent due to the subject.

10.1.2 Test KGRID 2

The subject selected for this test was a university lecturer who had previous managerial experience and was known to be highly articulate. It was hypothesised on the basis of these two facts that a more complete grid would be obtained in his case and probably a larger number of superordinate constructs (factors).

Proceeding in exactly the same way as described in 10.1.1 a principal components analysis was carried out on KGRID 2. This yielded five eigenvalues greater than unity (cf. 3 eigenvalues $>$ 1.00 for KGRID 1) thus confirming both hypotheses, since the subject also provided eighteen constructs (cf. 11 constructs from KGRID 1).

The correlations between constructs are given in Table (10.7), the eigenvalues are given in Table (10.8) and the factor loadings for the five factors extracted are given in Table (10.9). The significant factors are arranged in descending order in Table (10.10).

It is proposed to comment only briefly on the interpretation of the results. The first general comment to be made is the richness of the constructs elicited. However these are highly correlated and the five factors extracted have been named as follows:-

Factor No.	Description
1	Science - Philosophy
2	Purposeful - Haphazard
3	Measure of people - Measure of things
4	Quantitative - Qualitative
5	Systems - Not Systems

The value of the above terms is only in their ability to describe the clusters of constructs gathered at each factor pole. (Factors

COPY KGRID2

test no. 2R date

notes:- Elements elicited 7.4.71
Constructs elicited 8.4.71
Grid scored over week-end
(Easter) 9-13.471

1.Response in terms of productive runs	2.Participation in seminars/discussions	3.Budget preparation	4.Compilation of works lists	5.Human Relations in general	6.Indecisiveness	7.Unreliability	8.Resource scheduling	9.Committee type meetings	10.Meeting student's expectations of personal development	11.Spurious accuracy	12.Trustworthyness	13.	14.System performance parameters	15.Examination marks	16.Perception in terms of sensitivity to other people	17.Personnel selection	18.Costing techniques	19.Previous Midland Bank system	20.Computer application at N.A.S.A.	21.Strategic Planning	22.Planning of educational process	23.National Health real time patient recording systems	24.Face to face communication (with scratch pad)	25.Personal relationships within system	side entry A	side entry B
-2	0	+2	+1	+1	-2	-2	+2	+1	+1	0	+2		+2	+2	+1	+2	+2	+1	+2	+2	+2	-2	+2	+1	1.Control	Lack of control
+1	+1	-2	-2	+2	+2	+2	-1	+2	+2	0	+2		+1	-1	+2	+2	-1	-1	-1	0	+2	+1	+2	+2	2.Human factors	Lack of human factors
-2	+2	+2	+2	0	-2	-1	+2	+1	+2	0	0		+2	+2	0	+2	+1	+2	+2	+2	+2	-2	0	+1	3.Objectives clear	Objectives not clarified
-2	-2	+1	0	+2	0	0	+2	0	-1	0	0		0	0	-2	+2	0	+1	+2	+2	+2	+2	-2	+2	4.Long term	Short term
+2	-1	+2	+1	-1	-2	-1	+1	0	0	+2	-1		+1	+1	-2	0	+1	+1	+1	+1	0	0	-2	-1	5.Objective	Highly personalise(subjective)
-2	+2	+2	-1	+2	+2	+2	+1	+1	+2	+1	+1		+1	0	+2	+2	-2	-2	-2	+2	+2	-2	+2	+2	6.Subject to personal philosophy	Personal philosophy subject to it
-2	-2	-1	-2	+2	+2	+2	+1	-2	+2	0	0		+2	-2	+1	-2	-2	-2	+2	-1	+1	+2	-2	+2	7.Continuous activity	Discrete activity
+2	0	+1	+1	-2	0	0	+1	0	+1	+2	0		+1	+1	-2	-1	+2	0	0	+1	+1	0	-2	-2	8.Judgement based on quantification	Judgement based on values
+1	0	-2	+2	+2	+2	+2	+1	+1	+1	-2	+2		-1	+1	+2	+2	-2	-1	-1	-1	+1	+1	+2	+2	9.Measures of people	Measures (or manipulation) of data
-1	+1	+1	+1	+2	+2	+1	+2	+2	+2	-2	+1		+2	+1	+2	+2	-1	+1	+2	+2	+2	+2	+2	+2	10.Complex judgemental process	Simple judgemental process
+1	-2	+1	-2	+2	-2	-2	+2	+1	+1	-2	-2		+2	-1	-2	+1	-1	+2	+2	+2	+2	+2	-2	+2	11.Systems	Non-systems
+1	+1	+2	+2	+2	+2	+2	+2	+1	+1	0	0		+2	+2	+1	+2	+2	+2	-1	+1	+2	+1	+2	+2	12.Bread	Jam
+2	0	+2	+2	0	0	0	+1	+1	0	+2	0		+1	+2	-2	-1	+2	+1	0	0	+1	0	0	0	13.Tend to give rise to spurious accuracy	Accuracy debateable
-2	+2	-1	-1	+2	+2	+2	-1	+2	+1	-2	+2		-1	-1	+2	+1	-1	-2	-1	+1	+1	+1	+2	+2	14.Art	Technique
+2	-2	+1	+1	-2	-2	-2	-1	-2	-2	+2	-2		-1	+1	-2	-2	+1	+1	0	-2	-2	-2	-2	-2	15.Closed	Open-ended
+1	-2	+2	-2	-2	-2	-2	+1	-1	-2	0	0		+1	-2	-2	-1	+2	+2	-1	+1	0	0	-2	-2	16.Financial	Not financial
-2	-2	+1	+2	-2	-2	-2	-1	-2	-1	-1	-2		-1	-1	-2	-2	+1	-1	-1	-2	-2	-2	-2	-2	17.Mechanistic	Probabilistic
+2	0	-1	+2	0	0	+1	-2	0	+1	+2	+1		+2	+2	0	-2	+1	+2	-2	-1	-1	-1	0	0	18.Evaluation of past	Evaluation of future

FIG (10.2)

Table (10.7) RANK CORRELATIONS* BETWEEN CONSTRUCTS IN KGRID 2 ($X10^{-2}$)

Construct No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1																		
2	- 24																	
3	62	- 46																
4	01	04	- 27															
5	22	- 08	40	- 07														
6	02	60	02	- 29	- 58													
7	- 20	34	- 23	- 04	- 29	25												
8	10	- 59	24	15	80	- 42	- 22											
9	- 22	66	- 40	01	- 74	39	17	- 65										
10	09	44	01	11	- 32	40	53	- 38	18									
11	27	- 15	35	08	31	- 17	32	02	- 30	36								
12	20	- 05	18	- 06	- 08	22	- 07	- 08	23	- 11	12							
13	09	- 66	19	09	77	- 54	- 41	79	- 54	- 50	01	19						
14	- 17	78	- 36	05	- 93	66	27	- 74	67	32	- 27	- 01	- 72					
15	08	- 80	19	- 11	85	- 67	- 37	68	- 58	- 52	00	04	83	- 90				
16	34	- 51	20	11	69	- 46	- 21	55	- 70	- 24	43	- 04	48	- 64	50			
17	32	- 79	52	- 14	68	- 46	- 15	59	- 55	- 31	04	18	66	- 72	75	35		
18	- 30	- 13	- 16	- 08	21	- 36	- 23	36	- 06	- 51	- 38	04	49	- 31	48	- 04	31	

*Spearman's rho

Table (10.8) EIGENVALUES GREATER THAN UNITY FOR RANK CORRELATION COEFFICIENTS OF TABLE (10.7)

Component No.	1	2	3	4	5
Eigenvalue	7.66	2.62	1.77	1.21	1.08
% of Total Variance	42.54	14.55	9.85	6.7	5.98
Cumulative % of Total Variance	42.54	57.09	66.94	73.64	79.62

Table (10.9) FACTOR ANALYSIS OF KGRID 2

			Factor Loadings ($X10^{-2}$)				
Construct No.	Common Variance	Specific Variance	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5
1	.41	.59	13	59	10	18	10
2	.76	.24	- 82	- 21	- 05	- 17	- 06
3	.95	.05	26	80	24	28	29
4	.23	.77	- 07	- 13	17	- 34	- 26
5	.95	.05	91	16	28	04	- 15
6	.58	.42	- 66	25	- 05	- 18	21
7	.28	.72	- 36	- 03	11	- 02	- 37
8	.97	.03	76	06	40	- 44	04
9	.66	.34	- 65	- 16	- 44	- 08	09
10	.44	.56	- 50	16	22	18	- 30
11	.85	.15	06	50	22	25	- 70
12	.95	.05	02	58	- 70	- 34	- 09
13	.84	.16	86	06	- 04	- 31	02
14	.93	.07	- 93	- 13	- 11	08	16
15	.99	.01	97	- 08	- 12	06	01
16	.59	.41	57	16	29	- 03	- 40
17	.74	.26	77	30	- 04	06	22
18	.48	.52	44	- 31	- 23	- 25	26

Table (10.10) CONSTRUCTS HAVING SIGNIFICANT LOADINGS IN KGRID 2

Factor No.	Construct No.	Construct	Factor Loading	Specific Variance
1	15	Closed/Open-ended	.97	.01
	5	Objective/Highly personalised	.91	.05
	13	Tends to spurious accuracy/Accuracy debatable	.86	.16
	17	Mechanistic/Probabilistic	.77	.26
	8	Judgement based on quantification/ Judgement based on values	.76	.03
	16	Financial/Not Financial	.57	.41
	18	Evaluative of past/Evaluative of future	.44	.52
	7	Continuous/Discrete	- .36	.72
	10	Complex judgemental/Simple judgemental	- .50	.56
	9	Measure of people/Measure of data	- .65	.34
	6	Subject to personal philosophy/ Personal philosophy subject to it	- .66	.42
	2	Human factors/Lack of human factors	- .82	.24
	14	Art/Technique	- .93	.07
2	3	Objectives clear/Objectives not clear	.80	.05
	1	Control/Lack of control	.59	.59
	12	Bread/Jam	.58	.05
	11	Systems/Non-systems	.50	.15
	17	Mechanistic/Probabilistic	.30	.26
	18	Evaluative of past/Evaluative of future	- .31	.52
3	8	Judgement based on quantification/ Judgement based on values	.40	.97
	9	Measure of people/Measure of data	- .44	.66
	12	Bread/Jam	- .70	.95
4	13	Tends to spurious accuracy/Accuracy debatable	- .31	.16
	4	Long term/Short term	- .34	.77
	12	Bread/Jam	- .34	.05
	8	Judgement based on quantification/ Judgement based on values	- .44	.03
5	10	Complex judgemental/Simple judgemental	- .30	.56
	7	Continuous/Discrete	- .37	.72
	16	Financial/Not financial	- .40	.41
	11	Systems/Not Systems	- .70	.15

having all positive (or all negative) loadings must still be regarded as being bi-polar because of the dichotomous nature of the constructs. In Tables (10.6) & (10.10) the constructs are listed according to one pole. Reversing the construct, for example changing "Job/Not Job" to "Not Job/Job", would tend to move the construct toward the opposite pole of the factor). The real insight into the meaning to be ascribed to the various factors is gained by careful consideration of the constructs gathered at the factor poles in terms of their element scores in the manner indicated in 10.1.1.

The construct "Bread/Jam" is used by the subject to indicate "basics" and "trimmings". Virtually all of its variance (95%) is distributed in the common factor space and thus it is useful as a measure of other constructs. It loads on the "purposeful" and "quantitative" ends of factors 2 & 4 and indicates that the subject sees these aspects of his universe as basic. However the construct loads most heavily on the "measure of people" factor (factor 3) suggesting a significant regard for "people" as distinct from data, which are more usually associated with quantitative methods.

Factor 5 was named "systems - not systems" after the construct having the greatest loading in that factor. This decision was supported by the fact that the construct with the next highest loading was financial/not financial, since finance in its various forms provides some of the most widely dispersed systems in human society. To find a manager having systems as a super-ordinate construct must be a fortunate occurence indeed for a system designer charged with the task of introducing a new system.

Finally, to obtain a measure of the stability of the subject's scoring, he was asked to rescore his grid 13 weeks later. The scores on each construct were then compared using the sign test. The results

COPY KGRID2-1

test no. 2R-1 date 6.7.71

notes:- KGRID2 re-scored 13 weeks after initial scoring

(Scored by column)

1.Response in terms of productive runs	2.Participation in seminars/discussions	3.Budget preparation	4.Compilation of marks lists	5.Human relations in general	6.Indecisiveness	7.Unreliability	8.Resource scheduling	9.Committee type meetings	10.Meeting students' expectations of personal development	11.Spurious accuracy	12.Trustworthiness	13.	14.System performance parameters	15.Examination marks	16.Perception in terms of sensitivity to others	17.Personnel selection	18.Costing techniques	19.Previous Midland Bank system	20.Computer application at N.A.S.A.	21.Strategic planning	22.Planning of educational process	23.National Health real time patient recording system	24.Face to face communication with scratch pad	25.Personal relationships within system	side entry A	side entry B
-2	+1	+2	+1	+1	-2	-2	+2	-1	+2	-1	+2		+2	+1	+1	+1	+2	-2	+2	+2	+2	-2	+1	+1	1.Control	Lack of control
+1	+2	-1	+1	+2	+1	+1	0	+1	+2	0	+2		0	+1	+2	+2	-2	-1	+2	+2	+2	0	+2	+2	2.Human factors	Lack of human factors
+2	+2	+2	+2	+1	-2	-1	+2	0	+2	0	+2		+2	+1	0	+2	+2	-2	+2	+2	+2	-2	+2	+2	3.Objectives clear	Objectives not clarified
0	-1	+1	-2	+1	0	0	+1	0	+2	0	0		0	0	0	+2	0	-1	+2	+2	+2	+2	-2	+2	4.Long term	Short term
0	0	+2	+1	-2	-2	-1	+1	0	-1	0	0		+1	+1	-2	-1	+1	0	+1	+1	+1	0	-2	-2	5.Objective	Highly personalised(Subjective)
0	0	+1	-2	+2	+2	+2	+1	-1	+1	+1	+1		+1	+1	0	+2	+1	0	0	+1	+1	0	-1	0	6.Subject to personal philosophy	Personal philosophy subject to it
0	0	-1	+1	+2	0	0	+1	-2	+1	0	0		0	0	+2	-2	-1	+1	+2	+2	+1	+2	-2	+2	7.Continuous activity	Discrete activity
+1	0	0	+2	-1	0	0	0	0	0	+2	-2		0	+1	-1	-1	+1	0	0	0	0	0	-1	-2	8.Judgement based on quantification	Judgement based on values
0	+1	-2	-1	+2	0	+2	0	0	0	0	+2		+1	+1	+2	+2	-2	-1	0	0	0	0	+2	+1	9.Measure of people	Measure (or manipulation) of data
+1	+2	+2	-2	+2	0	0	+2	+1	+2	0	+2		+2	0	+2	+2	+1	+2	+2	+2	+2	+2	+1	+2	10.Complex judgemental process	Simple judgemental process
+1	+1	+2	0	+1	0	0	+2	+1	+1	0	-2		0	-2	0	+1	+1	+2	+2	+2	+2	+2	-2	+2	11.Systems	Non-systems
+2	0	+2	+1	0	0	0	+1	0	0	0	0		0	0	0	0	+2	+1	-1	0	0	0	0	0	12.Bread	Jam
0	0	+1	-2	-2	0	0	0	0	0	+2	0		0	+1	0	0	+2	0	0	0	0	0	0	0	13.Tend to give rise to spurious accuracy	Accuracy debatable
+1	0	0	-2	+2	+1	0	+1	+1	+2	-2	0		-1	0	+2	+1	0	0	0	+1	+1	0	+2	+2	14.Art	Technique
+1	-1	+1	+2	-2	-2	0	-1	-1	-2	+2	0		0	+2	-2	-2	+1	-1	-2	-2	-2	-2	-2	-2	15.Closed	Open-ended
+1	-2	+2	-2	-2	0	0	+1	0	-1	0	0		0	0	0	-1	+2	+1	-2	+1	+1	+1	0	0	16.Financial	Not financial
0	-2	-1	+2	-2	0	0	0	-2	-2	0	-2		0	0	-2	-2	0	-2	-2	-2	-2	-2	-2	-2	17.Mechanistic	Probabalistic
0	0	0	+2	0	0	0	0	0	0	0	+2		0	+1	0	0	0	0	0	0	-1	0	0	0	18.Evaluation of past	Evaluation of future

FIG (10.3)

of this are presented in Table (10.11) where it can be seen that only the scores on construct 12 (Bread/Jam) are significantly different from the initial scoring. Almost certainly the subject had forgotten the context in which he originally used this unusual term. Otherwise, within the limits of statistical variation, the scoring had remained constant. This stability is in accordance with more rigorous measurements of grid reliability reported in Bannister & Mair (1968, Chapter 7). Clearly if significant shift of construct scoring were to occur over relatively short periods, there would be little justification for using grids as a basis for system design work.

10.2 THE FINAL TEST FORMAT: SOME FURTHER TESTS

Following the administration and analysis of tests KGRID 1 and KGRID 2, the element elicitation questionnaire shown in fig (10.4) was adopted. This dropped the personality categories in accordance with the interim-conclusion reported in 9.3(iii). Additionally, triad sorts were based upon inspection of the elements elicited and were not selected in a pre-arranged order.

A further three tests were carried out using the new format. This gave practice in administering the test before approaching the C.A.D. manager. The relevance of the test was increased by using the limited number of managers available and the results are presented in KGRID 3, KGRID 4 and KGRID 6. (As indicated by the numbering system, test KGRID 6 was administered after the first C.A.D. manager had been interviewed but before the second was appointed). The correlation coefficients between constructs and the factor analyses for these grids are presented in Tables (10.12) - (10.17) for completeness but no detailed analysis is given. However some general points arose in the course of these tests that are worthy of note.

Thus, in the course of administering test KGRID 3, the element elicitation was carried out on one day and construct elicitation on the

Table (10.11) RESULTS OF SIGN TEST ON TEST-RETEST SCORES

Construct No.	Plus	Minus	Match	Probability of Chance Occurence
1	3	6	15	.51
2	7	6	11	1.00
3	6	3	15	.51
4	4	4	16	.73
5	3	7	14	.34
6	5	11	8	.21
7	7	5	12	.77
8	4	9	11	.27
9	5	9	10	.42
10	6	7	11	1.00
11	8	3	13	.34
12	1	18	5	$<.001$
13	2	10	12	.04
14	8	7	9	1.00
15	7	3	14	.34
16	10	3	11	.09
17	7	5	12	.77
18	7	8	9	1.00

Conclusions: Only in the case of construct no. 12 (Bread/Jam) is there a significant difference between the first and second grid test scores.

Fig. 10.4 ELEMENT ELICITATION QUESTIONS IN FINAL TEST FORM

Questions

1. If you had a job that could be done by computer which particular aspect would be most difficult to actually get the job done?

2. What is the most satisfying task in your job?

3. Give an example of an application or task in industry at which, in your opinion, a computer is (or would be) superior in performance to a human.

4. Is there a calculation or series of calculations you regularly have to perform.

5. What is the most difficult task in your job?

6. Give an example of a short term decision you regularly have to make in the context of your job.

7.

8. Which task or aspect of your job would be easiest to explain to an outsider?

9. What is the most boring task you perform?

10. What is the most important task in your job?

11. Indicate some quantity or figure which most exemplifies in your own mind the subject of statistics.

12. Is there some task you perform which involves financial decision making and if so what is it?

13. Which task in your job do you consider as least likely to be capable of being done by computer?

14. What do you regard as the most important information you generate?

15. What do you regard as the most important information you receive?

16. Give an example of a long term decision you have to make in the context of your job.

17. Give an example of an application in industry where a human is superior in performance to a computer.

18. What is the "best" management technique you know of?

19. What is the "best" application of computers you have had personal experience of?

20. What is the "best" application of computers you have heard of?

21. What is the most interesting task in your job?

22. Is there an application or area of your job which is particularly unstructured, and if so what is it?

23. Give an instance of a specific application of computers that you consider as least likely to be useful.

24. What do you regard as the best way to communicate factual data at work?

25. Which task or aspect of your job would be most difficult to explain to an outsider?

COPY [illegible]

test no. 3R **date** 18.5.71

notes:-

18.5.71 - Elicitation of elements
19.5.71 - Construct elicitation and scoring

1.Testing a running computer programs	2.Successfully completing a job on own	3.Producing a detailed sequencing plan	4.	5.Dealing with people of low intelligence	6.Making decision on reliability of Co. fitting on Dutch tube	7.	8.Calcn. of means & std. deviations from num.data	9.	10.Getting subordinates to do what ought to be done	11.Concept of probability	12.Project evaluation	13.Problem on what to calculate a std.devn.	14.Probabilities and sensitivity analysis	15.Estimate of fittings forecasts	16.Work content of section 5 years from now	17.Product mix decision making	18.Sensitivity analysis	19.Stop-cock component breakdown program	20.I.C.L. Data base	21.Presenting results in a form acceptable to other people	22.Copper price analysis	23.Sales order processing	24.Verbal communication	25.Explaining theory of statistics	side entry A	side entry B
+2	-2	+2		-2	+2		-1		+1	-2	-2	-2	-2	+2	0	+1	-1	+2	0	-2	+2	+2	0	0	1.Specific tasks	General situation
+2	0	+1		-2	+2		+2		-2	+1	0	+2	+1	+2	0	+2	+1	+2	+1	-2	+2	+2	-2	-2	2.Technical	People
-1	+2	+1		+1	+2		0		+2	0	-2	+2	0	0	0	0	0	+2	0	+2	-2	0	+2	+2	3.Gives personal satisfaction	Independant of personal satisfaction
0	0	0		0	+2		0		+2	0	0	+1	0	0	0	0	0	+2	+2	0	-2	-2	0	0	4.Successful	Unsuccessful
0	+2	0		0	0		0		-2	0	0	0	0	0	-2	0	0	-2	0	+2	0	0	0	0	5.External to dept.	Internal to dept.
-2	+2	+2		+2	-2		-2		+2	+2	0	+2	+2	-2	+2	-2	+2	-2	-2	+2	0	0	+2	+2	6.Critical analytical technique	
-1	0	0		+2	-2		-2		+2	-2	-1	-2	-2	-1	+2	-1	-1	+2	0	+2	-1	0	+2	-2	7.Managerial	Professional
0	+2	0		+2	-2		-2		+2	0	-1	-2	-2	-1	+1	0	0	+2	0	+2	+2	0	+2	-2	8.Ego	Technician
0	0	0		0	0		0		0	0	0	0	0	0	0	0	0	0	0	0	0	-2	0	0	9.Valuable	Not valuable
0	-2	0		+2	+2		0		+2	0	0	-2	0	0	0	0	0	0	0	-1	0	0	+2	-2	10.Influencing people as a person	Influencing people as a technician
+2	+1	+2		0	+2		-2		0	-2	+2	+2	-2	+2	0	+2	-2	+2	+2	0	+2	+2	0	-2	11.Practical	Theoretical
+2	0	+2		0	+2		+2		0	-2	+2	-2	-2	+2	0	+2	-2	+2	-2	-1	+2	0	0	0	12.Results	Techniques
0	0	0		0	0		0		0	0	0	0	0	0	0	0	0	0	0	0	0	-2	0	0	13.Do not annoy	Annoy
-1	0	+2		0	+2		-1		0	-2	+2	-2	+2	+2	0	+2	-2	+2	+2	-1	+2	+2	0	-2	14.(Successful) Application	Theoretical technique
-2	0	-2		0	-2		+2		0	+2	+2	-2	+2	-2	0	+2	+2	-2	-2	0	-2	-2	0	+2	15.Abstract	Particular
-2	+2	+2		+2	+2		-1		+2	-1	-1	-2	-2	-2	+2	+2	+1	+2	0	+2	+2	+2	+2	+2	16.Achievement	
-2	+2	-2		+2	+2		+1		+2	+2	-1	+2	+1	+2	+2	-2	+2	-2	-2	+2	+2	-2	+2	+2	17.Not computers	Computers
-2	+2	+2		+2	-2		-2		+2	+2	-1	+2	+2	-1	+2	+2	+1	+2	-2	+2	-1	0	+2	+2	18.Complex	Simple
+2	+2	+2		+2	+2		+2		+2	+1	+2	+1	+1	+2	+2	+2	+1	+2	-2	+1	+2	+1	+1	+1	19.My work at Company	Not Company
0	-2	-1		+2	-2		-1		-2	-2	0	-2	-2	-2	-2	-2	-2	-2	-2	-2	+2	+2	-2	-2	20.Inferiority	Superiority

FIG (10.5)

Table (10.12) RANK CORRELATIONS• BETWEEN CONSTRUCTS IN KGRID 3 $(X10^{-2})$

Construct No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1																		
2	51																	
3	- 11	- 37																
4	00	- 02	52															
5	- 40	- 09	05	- 36														
6	- 56	- 72	38	- 15	12													
7	12	- 46	26	06	- 22	19												
8	06	- 39	19	- 09	- 07	24	85											
9	31	- 06	- 06	18	- 45	- 17	28	22										
10	57	54	- 16	13	- 04	- 59	12	03	02									
11	63	41	- 21	- 11	- 17	- 62	01	- 01	24	52								
12	49	32	- 25	00	- 16	- 50	19	04	34	63	48							
13	- 63	- 43	- 11	- 20	11	35	- 26	- 18	- 08	- 75	- 26	- 38						
14	27	- 35	46	- 07	- 09	20	58	60	19	03	18	18	- 09					
15	- 35	- 43	37	- 01	13	55	- 03	15	00	- 46	- 33	- 50	18	13				
16	- 36	- 52	54	03	- 09	73	33	36	- 18	- 35	- 37	- 24	29	38	27			
17	36	21	- 13	- 01	- 27	- 33	15	18	28	33	82	33	- 14	20	- 07	- 13		
18	19	19	- 52	- 56	07	- 19	05	08	19	28	42	21	- 20	02	- 33	- 38	33	

•Spearman's rho

Table (10.13) FACTOR ANALYSIS OF KGRID 3

Eigenvalues >1	5.48 (30.4%)	3.40 (18.9%)	2.07 (11.5%)	1.51 (8.4%)	1.16 (6.4%)

			Factor Loadings $(X10^{-2})$				
Construct No.	Common Variance	Specific Variance	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5
1	.64	.36	63	23	03	19	39
2	.73	.27	49	- 10	- 51	12	45
3	.97	.03	- 40	75	20	- 40	13
4	.97	.03	- 25	80	- 42	24	- 12
5	.32	.68	- 14	- 30	07	- 37	27
6	.75	.25	- 66	- 10	42	- 31	- 16
7	.95	.05	- 04	35	80	43	- 07
8	.78	.22	- 04	20	80	31	- 06
9	.26	.74	22	21	10	30	- 26
10	.76	.24	54	24	- 09	39	51
11	.99	.01	96	20	02	- 08	- 04
12	.44	.56	51	11	03	36	18
13	.83	.17	- 26	- 35	- 05	- 37	- 71
14	.63	.37	09	36	68	- 17	04
15	.34	.66	- 39	05	16	- 39	- 11
16	.58	.42	- 46	20	46	- 30	- 16
17	.75	.25	77	26	12	- 03	- 27
18	.53	.47	53	- 42	23	11	12

COPY KGRID4

test no. 4R **date**

notes:-

24.5.71 - Element elicitation.
25.5.71 - Construct elicitation. Scored during following week.

1.Specifying one's requirement of a computer system in systems terms	2.Drawing of conclusions from market research studies	3.Analysing market research forms	4.	5.Economic forecasting in terms of the National Economy	6.Short term margins forecasts	7.	8.Preparation and explanation of sales control graphs	9.Editing market research questionnaires for consistent completion	10.	11.	12.Evaluating market research proposals	13.Long term forecasting	14.Market research information	15.Sales volumes and margins	16.Decision to buy desk calculator	17.Personnel selection	18.Systematic long range planning	19.Market research program	20.I.B.M.'s company planning computer application	21.	22.Forecasting fittings sales	23.	24.Graphical presentation of numerical data	25.Procedural problems in market research form design	side entry A	side entry B
-2	+1	-2		+2	+2		-2	-1			+2	+2	-2	-2	+1	+2	+2	-2	+2		+2		-2	+2	1.Balancing alternatives	Not balancing alternatives
-2	-1	+1		+2	+2		-2	+2			+2	+2	+1	-2	+1	+2	+2	-2	+2		+2		-2	+2	2.Evaluation	Not evaluation
0	-2	0		-1	-2		0	-2			+2	-2	0	0	-1	+2	-2	0	-2		-2		0	0	3.Requires judgement qualitatively	Requires judgement quantitatively
-2	-2	-2		-2	-2		-2	-2			-2	-2	-2	-2	-2	-2	-1	+2	+2		-2		-2	-2	4.Specific computer application	Not a specific computer application
-2	0	0		0	0		-2	0			0	0	+2	+2	0	0	0	0	0		0		-2	-2	5.Items of information	Presentation of information
+2	-2	-2		-2	-2		+2	-2			-2	-2	0	0	+1	+1	+2	0	+1		+2		-2	+1	6.Explanation of reasoning	Actual reasoning
-2	-1	+2		-2	-1		+2	+2			-2	-2	0	+2	-2	-2	-2	-1	-2		-2		+2	-1	7.Clerical task	Non-clerical task
0	-1	-2		+2	+2		-1	-1			0	+2	-2	-2	0	+1	+2	0	+1		+2		0	0	8.Future	Past
0	+2	+1		0	+2		0	-1			+1	+1	0	0	+2	-1	+1	0	+1		+2		0	0	9.Applying judgement to things	Applying judgement to people
+2	-2	+1		-2	-2		+1	+1			-2	-2	+2	+2	-1	-1	-2	+1	-1		-2		+2	0	10.Information	Interpretation of information
+2	-1	-1		+2	+2		-1	-2			-2	+2	0	0	-2	+1	+2	-1	+2		+2		-1	-1	11.Continuous decision making	Discrete decision making
+1	-1	-1		-2	-1		-2	-1			+2	-2	0	0	+1	+2	-2	+2	+2		+1		-2	-2	12.Problems requiring an individual approach	Problems requiring a general approach
+2	-2	0		-2	-2		+2	-1			-1	-2	+2	+2	-2	-2	-2	+2	-2		-2		+2	+1	13.Presentation of items of information	Use of items of information
-2	-2	+1		-2	-2		+2	-1			-2	-1	+2	+2	-1	-2	-2	+2	-2		-2		+2	-1	14.Non-application of judgement	Application of judgement
0	+1	-2		+1	0		-2	+2			+2	+1	-2	-2	+2	+2	+2	-1	+2		0		-2	0	15.Comparison of data	Collection of data

FIG (10.6)

Table (10.14) RANK CORRELATIONS• BETWEEN CONSTRUCTS IN KGRID 4 ($X10^{-2}$)

Construct No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1															
2	90														
3	-38	-38													
4	10	04	-22												
5	05	17	-15	10											
6	01	-10	07	29	-25										
7	-70	-52	19	-23	-02	-31									
8	79	63	-43	25	-14	14	-74								
9	35	19	-53	06	15	-05	-34	30							
10	-85	-70	51	-07	-12	14	63	-70	-62						
11	39	30	-38	22	10	28	-47	65	15	-29					
12	-01	-03	28	27	34	20	-34	-07	05	07	-08				
13	-82	-73	62	-08	-22	08	65	-68	-56	87	-36	-02			
14	-75	-63	36	-07	04	-10	75	-64	-38	72	-46	-22	74		
15	70	67	-33	23	09	03	-69	48	17	-61	02	26	-73	-70	

•Spearman's rho

Table (10.15) FACTOR ANALYSIS OF KGRID 4

Eigenvalues >1	6.71 (44.7%)	1.86 (12.4%)	1.57 (10.5%)	1.34 (8.9%)

			Factor Loadings (X10^{-2})			
Construct No.	Common Variance	Specific Variance	Factor 1	Factor 2	Factor 3	Factor 4
1	.96	.04	96	18	13	02
2	.90	.10	84	35	27	- 06
3	.53	.47	- 45	- 28	48	- 16
4	.10	.90	16	- 26	- 02	07
5	.15	.85	07	14	- 15	- 32
6	.31	.69	06	- 52	14	14
7	.95	.05	- 82	51	- 01	07
8	.90	.10	84	- 13	00	41
9	.66	.34	43	04	- 68	- 06
10	.88	.12	- 87	- 19	28	03
11	.58	.42	46	- 24	- 06	56
12	.48	.52	08	- 48	09	- 48
13	.89	.11	- 88	- 16	28	12
14	.71	.29	- 82	13	06	11
15	.78	.22	76	- 06	10	- 44

COPY KGRID6

test no. 6R **date** 16.6.71

notes:- Element and construct elicitation carried out on above date.
Scoring carried out 4.7.71

1.Devising procedures for accommodating the variability in fuel events	2.Resolving the competing requirements of the factories & the divn.	3.Airline reservation system	4.	5.Selling the concepts of computer modelling & M.I.S.s to senior management	6.Obtaining hot rolled coils in excess of supply	7.	8.Production allocation problem	9.	10.	11.	12.Stockholding/Transportation problem of semi-finished stock	13.Fostering the use of clerical work measurement within the divn.	14.Statement of production requirements to be used in long term plan	15.Sales forecasts	16.Decisions on the future structure of the computer network in the division	17.Labour relations	18.Linear programming	19.Scheduling the cold mill	20.Oil industry application of linear programming	21.	22.	23.Use of computers to establish authorship of Shakespeare's works	24.Graphical representation of facts	25.	side entry A	side entry B
+1	+2	0		+2	+2		+2				+2	-2	0	-2	+1	-2	0	+1	0			0	+1		1.Outcome under my control	Outcome not under my control
+2	-2	0		+2	-2		-2				0	+2	-2	0	+2	0	+2	0	0			0	0		2.Management Services responsibility	Operations Planning responsibility
+1	+2	+1		+2	0		+2				+1	-1	+1	+2	+2	+2	+1	+1	+1			-2	0		3.Important	Not important
0	+2	+2		0	0		+2				+2	0	0	+1	0	0	0	+1	+2			-2	0		4.Useful application of computers	Not a useful application of computers
+2	+2	-2		+2	+2		+2				+2	+1	+2	+2	+2	-1	+2	+2	-2			-2	+2		5.Within my personal experience	Not within my personal experience
+2	+2	+2		0	+2		+2				+2	+1	+2	+2	0	-2	+2	+2	+2			+1	+2		6.Concerned with numbers (Quantitative)	Not concerned with numbers (Qualitative)
-1	+1	+2		-1	+2		+2				+2	0	-1	0	-2	0	0	+2	0			0	0		7.Short term	Long term
+2	+2	-2		+2	+2		+2				+2	+1	+1	-2	+1	-2	+1	+2	-2			-2	+1		8.Job I do	Job done by others
0	-2	+2		0	-1		-1				-1	0	-2	-1	0	0	+2	+2	+2			+2	0		9.Actual computer application	Not an actual computer application

FIG (10.7)

Table (10.16) RANK CORRELATIONS* BETWEEN CONSTRUCTS IN KGRID 6 (X1$\bar{0}^{2}$)

Construct No.	1	2	3	4	5	6	7	8	9
1									
2	- 19								
3	12	- 04							
4	19	- 39	39						
5	58	- 07	32	01					
6	21	- 46	- 17	54	32				
7	26	- 55	- 15	60	- 08	48			
8	83	- 11	12	15	72	25	22		
9	- 29	54	- 34	- 17	- 57	- 19	- 03	- 42	

*Spearman's rho

Table (10.17) FACTOR ANALYSIS OF KGRID 6

Eigenvalues >1	3.39 (37.7%)	2.04 (22.6%)	1.29 (14.4%)

			Factor Loadings (X$\bar{10}^{2}$)		
Construct No.	Common Variance	Specific Variance	Factor 1	Factor 2	Factor 3
1	.72	.28	81	24	08
2	.72	.28	- 29	65	- 47
3	.15	.85	19	- 24	- 23
4	.42	.58	23	- 22	56
5	.69	.31	76	- 05	- 32
6	.33	.67	32	- 17	44
7	.88	.12	27	- 08	89
8	.96	.04	95	23	- 03
9	.92	.08	- 60	71	22

next. Although the subject claimed that he understood the requirement that a construct should involve a similarity and a difference that were related in some way, it rather seemed that some of the constructs offered did not correspond to this condition. The subject also had difficulty in finding significance in the elements when asked to carry out the triad sorts. He said it would have been helpful to have had available the questions used to elicit the elements, and on three or four occasions these were given when the subject appeared to be on the verge of giving up.

However the elements did not appear to be of a different quality to those obtained in KGRID 1 and KGRID 2, although admittedly some of the questions used to elicit them did not seem to be very appropriate to the subject's job. In fact the impression obtained from the elements before commencing construct elicitation gave the impression that the triad sorts would be fairly reasonable, which could not have been said for the earlier tests.

The subject attempted to score the elements immediately after finishing the construct elicitation stage. However he had immense difficulty as he found that he could no longer recall precisely what he had meant by either the constructs or the elements. The above behaviour was extreme and atypical. However it was similar in some ways to test KGRID 6.

In KGRID 6 the subject only produced 9 constructs, which was the least number obtained in the limited series of tests. Of these, some were closely related to the questions used to elicit the elements. For example, an element was elicited from the question "What is the most important task you do". When that element was used in the sort the construct given was "important - not important". This may have been due to the fact the element and construct elicitation stages were carried out in the same day and thus the subject could remember the initial questionnaire.

Undoubtedly KGRID 3 and KGRID 6 were the least successful tests carried out, which is why they have been described in some detail above. It may be relevant that these were the only tests carried out on subjects who understood the principles of the test and its purpose.

The construct elicitation stage of the test was always the most demanding of the subject. The subject of KGRID 4 remarked that after completing this stage of the test he felt mentally exhausted and that he had felt disinclined to undertake tasks involving mental effort for 2-3 hours afterwards.

10.3 TESTS WITH THE C.A.D. MANAGERS

It was considered of great importance that this test went as smoothly as possible, since anomalous conditions could not be rectified as there was only one manager. The intention was to administer the test before the work of developing a planning system was commenced and to re-test when the planning system had been implemented. Any changes in attitude observed would be noted and analysed. In the event, however, this experimental design was ruined by the C.A.D. manager being changed during the implementation phase. As the new manager was already aware of the developments, the chances of measuring changes in attitude in the remaining months of the research were limited. Instead it was hoped that in the course of implementing the system the test could be re-applied using increasingly specific elements to both obtain more detailed design information from the manager and to gauge his understanding of system at various stages of development. This latter aspect is discussed in 10.4.

Sadly, lack of computer facilities made it impossible to analyse the grids obtained in KGRID 5 and KGRID 7 in the manner described for the earlier tests. The results presented in 10.3.1 and 10.3.2 were analysed using the program INGRID 67 developed by Slater (1967). The significance of using the latter procedure on the results obtained is also discussed in 10.4.

Fig. (10.8) INTRODUCTORY NOTES FOR REPERTORY GRID TEST

General

The purpose of the research is to study the problems of computer system design with particular respect to human factors. More specifically it considers the problem of design to meet individual needs and requirements with the context of a total system specification. This test is the tool that is being used to identify some of these needs and requirements, and one of the major objectives is to evaluate the utility of the test for this purpose.

Specific Notes

(i) This is mainly the test of a test not of the individual, but to achieve the former it is necessary to do the latter.

(ii) There is no correct or "right" answer to the questions. In all cases it is the individual's opinion that is sought, and that is the "right" answer if any is to be regarded as such.

(iii) In each case a specific example is required. Only the participant has to understand what this or its implications is in any detail so that, although he may wish to elaborate in order to check he has understood the question for example, the final answer is required in as brief a form as possible.

(iv) In some instances the participant may wish to give the same answer that was given in response to a previous question. When he indicates that this is the case it may be possible to modify the question to obtain an acceptable answer.

10.3.1 Test KGRID 5

As an introduction to the test, the information given in fig (10.8) was communicated verbally with the subject. The element elicitation stage went very smoothly which was gratifying as, finally, the subject was the manager holding the job for which the questions were designed.

In a 20 minute conversation after the 60 minute session spent eliciting the elements, it appeared that the subject's high motivation to collaborate stemmed from the opportunity it appeared to offer to him to discuss the technical aspects of his work. In the two years the manager had been in post in his previous job (controlling fittings stock), such an opportunity had not arisen. He seemed to regard the grid test as providing a chance to re-appraise his work using the test as a touchstone.

Construct elicitation took place the following day. The subject readily grasped the idea of construct formation, more so than earlier subjects, and during the course of the session he occasionally checked himself after deciding that the construct just given did not correspond to the requirement of similarity and oppositeness. He also realised that after a few constructs had been given, the same constructs tended to suggest themselves again and worked hard without prompting to give alternative constructs where possible.

However he did have a little difficulty in understanding what was required of him in the scoring of the elements.

Table (10.18) presents the product moment correlation coefficients for KGRID 5 and Table (10.19) shows the results of principal components analysis, which are discussed in the following paragraphs.

The practice of attempting to name factors (in this case principal components) has been continued. Where examination of the grouping of constructs within the factors enables an appropriate word or phrase to be used to describe the grouping itself rather than the individuals within

COPY KGRID5

test no. 5R date

notes:-

8.6.71 - Element elicitation.
9.6.71 - Construct elicitation. Scored in following week.

side entry A	1.Obtaining confirmation that original problem is being satisfied but in manner desired	2.Mathematical [illegible] of stock control	3.Numerical control of production of complex components	4.Load balancing	5.Interfacing with other departments	6.Deciding the validity of certain production orders	7.	8.Explaining how stock is controlled at outside distribution depots	9.Writing input sheets for computer	10.Monitoring staff action	11.Variance	12.Calculating manufacturing batch quantities	13.Investigating customer orders where computer cannot give satisfactory delivery	14.Production orders on factory	15.Sales forecasts	16.Deciding values of fixed stock parameters on computer	17.Communicating	18.Management by exception	19.Marconi system of integrated stock control and purchasing	20.Marconi(Canada) Production Control System	21.	22.Policy for distribution outlets & what should be stocked at these	23.Speech recognition by computer	24.Presenting information by person to person communication	25.Explaining the concept of forecasting and monitoring	side entry B
1.Interests me	+2	+2	-2	+1	+2	+1		+1	-2	+2	+1	+2	+1	+2	+2	+2	+2	+2	-2	-2		+2	-2	+2	+1	Does not interest me
2.Not specific to my own personality	0	+2	0	-2	-2	-2		+2	+2	-2	0	-2	-2	+2	0	+2	-2	-2	0	0		-2	0	-2	+1	Specific to my own personality
3.Capable of future solution	+1	0	0	+2	-1	+2		0	+2	+1	0	+2	+1	+2	0	+1	-2	+1	0	0		+1	0	+1	+2	Insoluble at any time
4.Worth doing (Good)	+2	+2	+2	+1	+2	+1		+1	+2	+2	+1	+2	+2	+2	+2	+2	+2	+2	+2	+2		+2	-2	+2	+2	Waste of time (Bad)
5.Computer will assist with this problem	-2	+2	+2	+1	-2	-2		-2	-2	-1	+2	+2	-1	+2	-1	-1	-2	+2	+2	+2		+1	-1	-2	-1	Computer cannot assist with this problem
6.Within my control	-2	+1	-2	+1	-1	+2		+2	+2	+2	-2	+2	+2	+2	-2	+2	-1	+1	-2	-2		-1	-2	+1	+2	Not within my control
7.Long term	+1	+1	0	+1	+1	-2		-1	-2	+2	-2	-1	-2	-2	-2	-1	-1	-1	0	0		-1	+2	-1	-1	Short term
8.Fairly easy to do	-2	+1	-1	-1	-2	+1		+1	+2	-1	+1	+1	+1	+1	-2	+1	-2	+1	+1	-1		-1	-2	-1	+1	Fairly difficult to do
9.Illogical	-2	-2	-2	-2	-2	-2		-2	-2	-2	+1	-2	-1	-2	-1	-2	-2	-2	-2	-2		-2	+2	-2	-2	Logical
10.Personal	+2	-1	-2	-1	+1	+1		+1	-2	+2	-2	-1	+1	-1	+1	-1	+2	+1	-2	-2		+2	+1	+2	+1	Impersonal
11.Can be written down	+1	+2	+2	+1	+1	+1		+2	+2	+1	+2	+2	+1	+2	+2	+2	+1	+1	+2	+2		+1	-1	-1	+2	Cannot be written down
12.Past	-2	+1	+1	-2	-1	-1		-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	+2	+2		-2	-2	-1	-1	Present
13.Have been involved with	+2	+1	-1	+1	+2	+2		+2	+2	+1	+1	+2	+2	+2	-1	+2	+2	+2	+2	-1		-1	-2	+2	+2	Have not been involved with
14.Associated with computer	+2	+2	+2	+1	-1	+1		+2	+2	-1	+2	-1	-1	+2	-1	+2	-1	-1	+2	+2		-2	+2	-2	-2	Not associated with computer
15.Outside this company	+2	+1	-1	+1	+1	+2		+2	+2	+1	+1	+2	+2	+2	+1	+2	+1	+1	-2	-2		+2	-2	+1	+2	In this company
16.Can be delegated	-2	+1	0	-1	-1	+1		+2	+2	-1	0	+1	+1	+2	0	+2	-1	-1	0	0		0	0	+1	+1	Cannot be delegated
17.Requires decision making	-1	+1	0	+2	+1	+2		-1	+1	+2	0	+2	+2	+2	0	+2	+1	+2	0	0		0	0	+1	-1	Does not require decision making

FIG (10.9)

Table (10.18) PRODUCT MOMENT CORRELATION COEFFICIENTS OF CONSTRUCTS IN KGRID 5 ($X10^{-2}$)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1																	
2	- 27																
3	32	- 09															
4	35	- 04	- 04														
5	- 18	07	- 22	13													
6	36	09	62	19	- 20												
7	- 14	- 12	- 30	- 32	02	- 28											
8	- 07	41	28	19	26	63	- 53										
9	- 27	04	- 06	- 79	02	- 39	09	- 16									
10	56	- 47	20	- 07	- 63	09	13	- 47	- 03								
11	- 06	51	- 12	51	40	09	- 35	49	- 36	- 60							
12	- 50	26	- 71	32	49	- 25	07	20	- 25	- 59	45						
13	42	00	26	43	- 27	60	- 30	52	- 47	08	15	- 11					
14	- 53	65	- 24	- 32	24	- 19	14	23	21	- 65	32	34	- 11				
15	72	01	68	36	- 36	66	- 47	34	- 34	35	16	- 61	58	- 33			
16	- 10	58	27	02	- 03	59	- 56	70	- 07	- 33	29	11	21	21	30		
17	32	- 36	34	15	11	57	- 12	26	- 20	- 06	- 11	- 14	34	- 17	27	13	

Table (10.19) PRINCIPAL COMPONENT ANALYSIS OF KGRID 5

Component No.	1	2	3	4	5
Eigenvalues >1	4.88	4.20	2.28	1.41	1.05
% of Total Variance	28.73	53.43	66.82	75.13	81.31
Construct No.	Component Loadings (X10^{-2})				
	(1)	(2)	(3)	(4)	(5)
1	- 70	33	- 24	- 08	- 15
2	15	- 67	37	- 47	13
3	- 69	10	43	17	- 16
4	- 42	- 33	- 77	- 17	- 09
5	35	- 46	- 27	51	- 38
6	- 81	- 27	20	21	30
7	48	41	- 19	12	54
8	- 41	- 77	23	18	01
9	46	26	66	13	- 30
10	- 37	81	- 04	- 28	07
11	- 04	- 79	- 27	- 23	- 24
12	51	- 60	- 47	05	16
13	- 70	- 23	- 15	- 05	38
14	48	- 54	36	- 05	28
15	- 92	- 01	09	- 22	- 14
16	- 34	- 65	46	- 07	02
17	- 49	- 04	- 08	76	16

Table (10.20) CONSTRUCTS HAVING SIGNIFICANT LOADINGS IN KGRID 5

Component No.	Construct No.	Construct	Component Loading
1	15	Outside this Company/In this Company	.92
	6	Within my control/Not within my control	.81
	1	Interests me/Does not interest me	.70
	13	Have been involved with/Have not been involved with	.70
	3	Capable of future solution/Insoluble at any time	.69
	17	Requires decision making/Does not require decision making	.49
	4	Worth doing (Good)/Waste of time (Bad)	.42
	8	Fairly easy to do/Fairly difficult to do	.41
	10	Personal/Impersonal	.37
	16	Can be delegated/Cannot be delegated	.34
	5	Computer will assist with this problem/ Computer cannot assist with this problem	- .35
	9	Illogical/Logical	- .46
	14	Associated with computer/Not associated with Computer	- .48
	7	Long term/Short term	- .48
	12	Past/Present	- .51
2	10	Personal/Impersonal	.81
	7	Long term/Short term	.41
	1	Interests me/Does not interest me	.33
	4	Worth doing (Good)/Waste of time (Bad)	- .33
	5	Computer will assist with this problem/ Computer cannot assist with this problem	- .46
	14	Associated with computer/Not associated with computer	- .54
	12	Past/Present	- .60
	16	Can be delegated/Cannot be delegated	- .65
	2	Not specific to my own personality/ Specific to my own personality	- .67
	8	Fairly easy to do/Fairly difficult to do	- .77
	11	Can be written down/Cannot be written down	- .79
3	4	Worth doing (Good)/Waste of time (Bad)	.77
	12	Past/Present	.47
	14	Associated with Computer/Not associated with Computer	- .36
	2	Not specific to my own personality/ Specific to my own personality	- .37
	3	Capable of future solution/Insoluble at any time	- .43
	16	Can be delegated/Cannot be delegated	- .46
	9	Illogical/Logical	- .66

Table (10.20) Cont'd

Component No.	Construct No.	Construct	Component Loading
4	17	Requires decision making/Does not require decision making	.76
	5	Computer will assist with this problem/ Computer cannot assist with this problem	.51
	2	Not specific to my own personality/ Specific to my own personality	- .47
5	7	Long term/Short term	.54
	13	Have been involved with/Have not been involved with	.38
	5	Computer will assist with this problem/ Computer cannot assist with this problem	- .38

the group, further insight into the grid structure may be achieved. Of course, the factor need not be named in the above way for an examination of its construct loadings to give a feel for the way the subject used the constructs. Indeed an appropriate word or phrase may not exist to describe the particular use of the constructs revealed by the factor analysis. However when the factor can be named it serves the practical purpose of giving a reference label that is more explicit than the terms factor 1, factor 2 etc.

As can be seen from Table (10.19), five components explain approximately 80% of the variance in the grid. The grid was well differentiated in the sense that seventeen constructs were elicited. With regard to the scoring of the constructs, the subject appears to have used the wrong convention with respect to the scores on construct 15, "Outside this Company/In this Company". Reference to the grid shows that the subject has given a positive score to elements 19 & 20 thus indicating that side entry A ("Outside this Company") applies to the elements. Examination of the elements shows this not to be true. This mistake seems to have been a temporary lapse in the subject's scoring as inspection of the other scores on the elements does not reveal any similar inconsistencies. The effect of the anomalous scoring would be to rotate the poles of the construct. Thus the entry for variable 15 in component 1 of Table (10.20) should read "In this Company/Outside this Company" and not "Outside this Company/In this Company" as shown.

Fifteen constructs are loaded on component 1 and it can be seen from Table (10.20) that some of those with positive loadings relate to self ("have been involved with", "interest me") while the value judgement "worth doing" also has a positive loading and is thus associated with this pole of the component. On the basis of this argument, constructs loaded on the negative end of the construct might be labelled "not-self".

However such a label would be inadequate, since constructs relating to the subject's job are also included in the component ("within my control", "in this company"). Thus some composite label such as "Self-Job/Not self - Not job' is suggested to describe component 1. The construct pole "job" is not associated with "long term" or "past", that is to say it is regarded as short term and related to the present. Ominously the construct pole "computer will assist" is not associated with "job" either.

The second component has been labelled "Subjective - Objective". At the "subjective" pole is loaded "Long-term". This was not associated with the job in component 1, but is now associated with "interests me". This is consistent with the fact that it was this subject who asked that forecasting work be carried out to assist with the monitoring of stock.

However these construct loadings contrast with that for "Worth doing/ Not worth doing", although the loading is relatively small, and this may indicate some ambivalence to the value of the long term approach. The constructs with negative loadings on component 2 in Table (10.20) relate to fairly simple, specific things and hence the label "objective". The computer is seen to be of assistance in objective applications despite the reservations indicated in component 1 on the use of computers in the context of the subject's own job.

Component 3 has been termed "Important/Not important" since at one pole it collects the construct pole "Worth doing" and at the other "can be delegated". "Past" is associated with the "Important" pole in this construct. In fact the construct "Past/Present" appears with significant loadings in the first three components with three connotations. In component 1 it is not concerned with the job and in component 2 it is seen as objective, that is presumably, as a matter of record rather than of speculation. In component 3 its use is not clear unless its importance, or worth, relatesto past personal achievements since elements relating to the subject's previous work appear in the grid and are scored on the construct "Past/Present". Again the subject's attitude to computers is not particularly favourable as the pole "associated with computers" is loaded on the "not important" pole of the component.

Component 4 associates "requires decision making" and "computer will assist" and contrasts these with "not specific to my personality". Since the latter two construct poles were associated in component 2 on the "Objective" pole, there is the suggestion that the subject perceives the computer to have a role outside the fairly specific application of computers to simple decision making of the automatic kind. This possibility would merit probing with the subject in a subsequent interview.

Finally, component 5 perhaps indicates that long-term work with which the subject has been involved, could not be assisted by the computer. This possibility also requires probing. If this view were found to be true in an application such as that described in Part 1, the road to successful implementation would certainly have to incorporate a significant amount education (overtly or otherwise), which would not be necessary in the case of a totally committed manager.

In summary, the subject appears to perceive his job as being associated with the present, as opposed to planning for the future or analysing the past. He sees that some of his responsibilities can be discharged by delegation while others cannot. On the whole, the job is not regarded as being associated with the computer. (This is despite the fact that the responses in the grid relate mainly to the subject's experience in the fitting's stock control area which had an extensive computer-based stock control system). Although the subject found a personal interest in long term aspects of his work he appeared to doubt the value of such aspects. The computer was seen as being useful for those applications that could be simply defined, but was not identified as being of use for long-term work in the subject's experience. Nevertheless it seemed possible that the subject did feel there was scope for the use of computers outside the more simple and tightly defined applications with which he was familiar.

10.3.2 Test KGRID 7

With one exception described below, this test was administered in exactly the same way as KGRID 5. The test proceeded smoothly through all stages, although there was no extended conversation with the subject at the end of the sessions as had been the case in KGRID 5. There was no particular reason for this except, perhaps, that a working relationship had been established over a number of weeks with the subject of KGRID 7 before the test was carried out and he was more familiar, therefore, with the research than the first manager had been at the time of his test.

The change in procedure in KGRID 7 was that an element was supplied when the questions in fig. (10.4) failed to elicit one. Ultimately elements nos 4, 7, 9, 10, 12, 21 & 22 were supplied, the others being obtained in the normal way.

Before discussing the principal components analysis of the completed grid, attention is drawn to the fact that four constructs were elicited referring to information. Constructs 2 and 17 appeared to be very similar but the subject scored them differently. Construct 2 "Eliciting information/Demanding information" was obtained from a triad sort involving element nos. 2, 5 & 13 in which elements 5 & 13 were perceived to be similar and opposite to element 2. However the scores on elements 2, 5 & 13 were 2, 1, 2 respectively, thus indicating that elements 2 & 13 were perceived as similar. A similar result is to be seen in Construct 8 "Giving information/Receiving information" where element nos. 24 & 25 were sorted as being similar and different to element no. 5 although the scores indicate that element nos. 5 & 24 are perceived as being similar. The

COPY KGRID7

test no. 7R date

notes:-

29.11.71 Element elicitation
1.12.71 Construct elicitation
Scored following week

1.Finding and preparing relevant data for computer	2.Interpreting computer output	3.Use of computers for stock control	4.Meeting production budgets	5.Eliciting information from people	6.Obtaining delivery promises	7.Allocation of small production volume products	8.Allocating domestic orders	9.Determining adequate stock levels for products	10.Allocating non-budgeted items to factories	11.Normal Distribution	12.Deciding acceptable product mix within capacity restraints	13.Getting improved delivery dates	14.Producing stock status information	15.Obtaining sales data	16.Placing production orders	17.Operating production machine by computer	18.Using method study	19.Use of computers in Ford Motor Co. stock control system	20.Use of computers for evaluation of design criteria in, say, electric motors	21.Monitoring stock movements of low demand products	22.Resolving stock committ-ments against ctual stock in placing production orders	23.Use of computers for game playing (chess, naughts & crosses)	24.Presenting factual data in written form	25.Explaining the problems of data interpretation	side entry A	side entry B
+1	+2	+2	+1	-1	-1	+1	+1	+1	+1	-1	+1	-1	+2	+1	+1	+1	-1	+1	+2	+1	+1	0	+1	-1	1.Deals with facts	Deals with people
+1	+1	0	0	+2	+1	0	0	+1	0	0	+1	+1	-1	+1	0	0	+1	0	0	0	0	0	0	0	2.Eliciting information	Demanding information
0	0	+2	+1	0	0	+1	+2	+2	+1	+1	+1	-2	+2	+2	+2	+2	-1	+2	+2	+2	+2	-2	+1	-1	3.Computer application	Not a computer application
0	0	0	-2	0	-1	-1	-1	+2	-1	0	-2	-1	0	0	-1	0	0	0	0	+1	+2	0	0	0	4.Stock restraint	Production restraint
-1	+1	+2	+2	0	-1	+1	+2	+1	+1	0	-1	-1	0	+1	+1	0	+1	0	0	+1	+1	0	0	0	5.Decided by Company Policy	Decided by exigence
+1	+2	+2	+1	+1	+2	-1	+1	+2	-1	-2	-1	+1	+1	+1	+2	-2	-2	-2	-2	+1	+1	-2	+1	+1	6.Intimately concerns me	Does not intimately concern me
+2	0	0	-2	+1	0	+1	+1	+2	-1	0	-1	0	-1	0	0	0	0	0	0	0	0	0	+2	-1	7.Prepared and presented by me	Presented to me
-1	-2	-2	0	+2	-2	0	0	+1	0	0	0	-1	+1	-1	0	0	0	0	0	0	0	0	+2	+1	8.Giving information	Receiving information
+1	+2	+1	+1	+2	+1	-1	+1	+1	-1	0	+1	+1	+1	+1	+2	-1	0	+1	+1	-1	+1	-2	+1	-1	9.Important	Not important
-1	+1	0	+1	0	+1	+1	+1	+2	+2	0	+2	-1	-2	-1	+1	0	0	0	0	+1	+1	0	0	+1	10.Making a decision	Not making a decision
+1	+2	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	-1	-2	+1	-1	+1	+2	-1	+1	-2	-1	+1	11.Fairly complicated to understand	Fairly simple to understand
0	0	+1	0	0	0	0	0	0	0	0	0	0	+1	0	+1	+2	0	+2	+2	+1	+1	-2	0	0	12.Valuable application of computers	Not a valuable application of computers
+2	-2	-2	0	+2	+1	0	0	+1	0	-1	+1	+1	-1	+1	0	0	+1	0	0	0	0	0	-2	0	13.Research for information	Presentation of information
0	0	0	+2	0	0	-2	+2	-1	-1	0	+1	-1	0	0	0	0	0	0	0	-1	-1	0	0	0	14.Predetermined variable	Variable fluctuates
+1	+2	+2	+1	+2	+2	0	0	+2	+1	+1	0	+2	+2	0	0	+2	+2	+2	-2	+1	0	0	+1	0	15.Of practical use	Of theoretical use
-1	-2	-2	+1	+1	+1	+2	+1	-2	+2	-1	0	+1	-2	+1	+2	0	0	0	0	+2	-2	0	+1	0	16.Deals with small part of product range	Deals with whole of product range
+2	+1	+1	+1	+1	+1	+1	+2	+2	+1	+2	+1	-1	+1	+1	0	+2	-1	+2	+2	+1	+1	0	-2	-1	17.Numeric	Literal
-2	0	0	+1	+2	+1	0	0	0	0	0	-1	+2	0	+1	0	0	+1	0	0	0	0	0	0	0	18.Obtaining reluctantly given information	Straight forward information gathering
-1	0	-1	+2	0	+2	+2	+2	-1	+2	0	+2	+2	0	-2	+2	0	+2	+2	0	0	+1	0	0	0	19.Production function	Sales function
+2	-1	0	-1	+1	0	-1	-1	+2	+1	+1	-1	0	0	0	0	-1	+1	-1	-1	0	-1	0	0	0	20.Setting up standards	Utilising standards

FIG (10.10)

Table (10.21) PRODUCT MOMENT CORRELATION COEFFICIENTS OF CONSTRUCTS IN KGRID 7 ($X10^{-2}$)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1																				
2	-43																			
3	68	-40																		
4	09	-01	21																	
5	39	-28	43	10																
6	14	21	10	12	22															
7	00	33	05	43	-11	18														
8	-14	-14	09	19	-05	-17	18													
9	21	38	25	-06	01	55	18	-11												
10	02	09	13	-09	30	07	-09	03	-12											
11	17	08	13	-05	-11	02	-10	-18	25	10										
12	43	-32	67	19	06	-10	-10	-01	24	-13	32									
13	-46	62	-26	-07	-38	-06	15	08	03	01	-04	-17								
14	06	-02	07	-47	15	05	-29	-01	31	-01	05	-07	-04							
15	-20	27	-16	12	-07	24	09	-12	16	-18	12	02	-03	-10						
16	-26	04	-09	-52	02	-11	-07	15	-24	21	-40	-11	28	-03	-26					
17	38	-03	57	13	11	-15	06	-23	11	10	51	37	15	12	-04	-26				
18	-50	34	-29	-11	10	13	-19	07	16	-15	-07	-11	17	-04	30	31	-33			
19	-23	-09	-16	-58	00	-21	-34	-02	-01	39	05	01	12	13	-01	47	-16	20		
20	-36	38	-24	32	-17	15	42	17	-02	-12	-18	-34	38	-26	34	-13	-05	-04	-37	

Table (10.22) PRINCIPAL COMPONENTS ANALYSIS OF KGRID 7

Component No	1	2	3	4	5	6
Eigenvalues >1*	3.87	3.06	2.23	1.77	1.57	1.33
% of Total Variance	19.37	34.65	45.80	54.63	62.47	69.12
Construct No.	Component Loadings (X10^{-2})					
	(1)	(2)	(3)	(4)	(5)	(6)
1	- 86	04	01	11	- 09	14
2	59	38	- 50	- 14	- 27	06
3	- 82	06	- 01	02	- 37	- 28
4	- 20	73	40	08	- 04	- 19
5	- 44	- 15	02	55	- 33	05
6	- 03	36	- 43	61	- 21	13
7	11	65	16	- 01	- 38	01
8	13	01	43	05	- 27	- 39
9	- 16	29	- 74	23	- 14	- 12
10	- 07	- 31	- 05	- 04	- 61	24
11	- 34	11	- 50	- 45	19	- 03
12	- 65	02	- 11	- 20	06	- 61
13	55	14	- 20	- 50	- 39	- 11
14	- 13	- 34	- 44	09	02	32
15	21	38	- 33	19	34	- 24
16	38	- 59	10	04	- 45	- 26
17	- 55	20	- 24	- 57	- 20	04
18	48	- 13	- 32	35	09	- 54
19	19	- 72	- 25	- 11	- 11	- 18
20	45	64	16	- 03	- 16	11

*Eigenvalue corresponding to component no. 7 (not extracted) is also greater than unity having a value of 1.24.

Table (10.23) CONSTRUCTS HAVING SIGNIFICANT LOADINGS IN KGRID 7

Component No.	Construct No.	Construct	Component Loading
1	1	Deals with facts/Deals with People	.86
	3	Computer application/Not a computer application	.82
	12	Valuable application of computers/Not a valuable application of computers	.64
	17	Numeric/Literal	.55
	5	Decided by Company policy/Decided by exigence	.44
	11	Fairly complicated to understand/ Fairly simple to understand	.34
	16	Deals with a small part of product range/Deals with whole of product range	- .38
	20	Setting up standards/Utilising standards	- .45
	18	Obtaining reluctantly given information/ Straightforward information gathering	- .48
	13	Research for information/Presentation of information	- .55
	2	Eliciting information/Demanding information	- .59
2	4	Stock restraint/Production restraint	.73
	7	Prepared and presented by me/Presented to me	.65
	20	Setting up standards/Utilising standards	.64
	2	Eliciting information/Demanding information	.38
	15	Of practical use/of theoretical use	.38
	6	Intimately concerns me/Does not intimately concern me	.36
	10	Making a decision/Not making a decision	- .31
	14	Pre-determined variable/Variable fluctuates	- .34
	16	Deals with a small part of product range/Deals with whole of product range	- .59
	19	Production function/Sales function	- .72
3	9	Important/Not important	.74
	11	Fairly complicated to understand/Fairly simple to understand	.50
	2	Eliciting information/Demanding information	.50
	14	Pre-determined variable/Variable fluctuates	.44
	6	Intimately concerns me/Does not intimately concern me	.43
	15	Of practical use/of theoretical use	.33
	18	Obtaining reluctantly given information/ Straightforward information gathering	.32
	4	Stock restraint/Production restraint	- .40
	8	Giving information/Receiving information	- .43

Table (10.23) Cont'd

Component No.	Construct No.	Construct	Component Loading
4	6	Intimately concerns me/Does not intimately concern me	.61
	5	Decided by Company policy/Decided by exigence	.55
	18	Obtaining reluctantly given information/ Straightforward information gathering	.35
	11	Fairly complicated to understand/Fairly simple to understand	- .45
	13	Research for information/Presentation of information	- .50
	17	Numeric/Literal	- .57
5	10	Making a decision/Not making a decision	.61
	16	Deals with a small part of product range/Deals with whole of product range	.45
	13	Research for information/Presentation of information	.39
	7	Prepared and presented by me/Presented to me	.38
	3	Computer application/Not a computer application	.37
	5	Decided by Company policy/Decided by exigence	.33
	15	Of practical use/Of theoretical use	- .34
6	12	Valuable application of computers/Not a valuable application of computers	.61
	18	Obtaining reluctantly given information/ Straightforward information gathering	.54
	8	Giving information/Receiving information	.39
	14	Pre-determined variable/Variable fluctuates	- .32

fact that the subject has emphasised information flow so much would certainly merit further investigation and clarification, particularly in view of his uncertain scoring of two of the constructs relating to information.

The results of principal component analysis on KGRID 7 is presented in Table (10.22) where it can be seen that seven components are significant on the basis of Kaiser's criterion. In fact only six components were extracted from the analysis for discussion the seventh component only accounting for less than 7 per cent of the total variance. The grid is quite different from the preceding tests by virtue of the fact that the variance is distributed over so many independent dimensions. No attempt is made to name all the dimensions.

Component 1 is a fairly general one associating fairly specific construct poles ("deals with facts", "numeric") at one of its poles and construct poles involving activities that are less precisely defined (setting up standards, research for information) at the other. The construct poles "Computer application" and "Valuable application of computers" are associated with the specific end of the dimension.

Components 2, 3 and 4 appear to be dimensions of the subject's job. The construct "Intimately concerns me/Does not intimately concern me" loads in various amounts on each of the three components. The subject's interest in information flow is again evident from the fact that constructs relating to information are loaded on each of components 2, 3 & 4.

Examination of the construct poles gathered at each end of component 2 suggest that a "production-sales" dimension has been defined. The subject's own job, of course, is primarily associated with the production end of the scale. On the basis of this suggestion the subject does not associate decision making with his job in this dimension.

In component 3, importance is associated with "fairly complicated" and "practical" things.

An interpretation of component 4 might be that some aspects of the job are defined by policy and do not require deep understanding ("fairly complicated") or difficult to obtain information.

Component 5 is concerned with decision making. Presumably "deals with a small part of product range" loads on this component to the extent that it too involves decision making which is associated with research for information and is affected by Company policy. "Computer application" loads on to this dimension but not, it is noted, "Valuable application of computers". A distinction is made between "decision making" and "practical" in component 5 which corresponds to a similar contrast in component 2.

Finally it is noted that component 6 associates "valuable application of computers" with information flow constructs.

The last test of the series, KGRID 7, is the most difficult to interpret. In summary, the subject appeared to have a number of dimensions along which he construed his job and in particular he seemed to be much concerned with information flow. On the whole he appeared to see his decision making as being constrained by company policy and difficult to obtain information, so making it of limited practical use. Obviously such a diagnosis is very tentative, but it points the way for further investigation probably carried out by unstructured interview. One possibility is to go through the analysis with the subject and to ask for his comments, although the reservations expressed in 10.2 should be borne in mind if it is intended to ask the subject to carry out further grid tests.

10.4 DISCUSSION

While component analysis gives considerable insight into grids KGRID 5 and KGRID 7, factor analysis is still preferred for the reasons given in 10.1. In the present case factor analysis would certainly have been carried out had the test been used for system design purposes.

However the relative rates of progress of the research in the two areas of application meant that the study described in Part 1 had passed the stage where the system design philosophy could be modified by the test results. Thus the findings reported in 10.1 - 10.3 only serve to exemplify the kind of results obtained from the test and the way in which they may be interpreted. For these reasons the principal components analysis suffices.

Chapters 7-9 inclusive have concentrated on the development and theory of the test without stressing its power to extract information efficiently. Thus the test can be carried out in 1½ - 2½ hours of interface time plus analysis time. Thus must be contrasted with the alternative method of deriving similar information in a semi-structured or even fully open interview which would take a similar amount of time to carry out. However the techniques for data reduction based upon summarising notes or content analysis are not nearly as powerful as those of factor analysis applied to a repertory grid. Furthermore the repertory grid is a completely neutral investigatory tool, while the problems of conducting a completely neutral interview are well known. For example, leading questions can bias the interviewee's responses and the possibility that the interviewee has misinterpreted the question is very real. By asking the subject to merely sort elements that he himself has provided, the repertory grid test avoids these problems.

However it is not suggested that grid tests should be used on all occasions in preference to interviews. The grid test is to be preferred when there is some possibility that the subject may,deliberately or otherwise, not give his true attitudes toward the subject in question. Obviously there will be cases when there is no reason to expect that other than the absolute truth will be given and in these cases the more natural approach of an interview may suffice.

The tests described were initially intended for use in a study of the attitudes of the C.A.D. manager both before and after his use of the allocation system. If the test proved to be sufficiently sensitive, it would provide valuable information concerning the attitudes that governed the acceptance of new practices. However changes in C.A.D. personnel associated with the research made this scheme infeasible.

It was decided instead to attempt to use the model iteratively with which ever manager was concerned with the project at the time. Initially the test form developed above would be used and would be supplemented by further tests and/or interviews to clarify points arising. The results from this stage would provide a basis for a system design to meet the user expectation. As the design proceeded to more detailed stages, correspondingly more detailed elements concerning various aspects of the new system would be introduced. This procedure would provide feedback to the analyst of any mis-understanding on his part or lack of comprehension on the part of the ultimate user. Of course, if the managers associated with the research continued to change, it was unlikely that future managers' expectations would also be met by the system if it had proceeded beyond the basic design stage based upon criteria developed from the expectations of earlier incumbents of the job. Nevertheless it was expected that new managers could, with some background information about the system, be expected to participate with subsequent grid tests concerned with more detailed design. In the event however, it was not possible in the time available to proceed beyond the development of a test upon which to base the initial system design.

In the field of psychotherapy Kelly (1955) favoured the elicitation of elements, otherwise problems of interpretation could occur. For example, in an investigation of interpersonal relationships a child might be asked to provide an example of the person he respected most, to which he might respond "my teacher". If "my teacher" were given as an element to a second child, for it "my teacher" might correspond to "the person I dislike most", with obvious problems in the interpretation of the associated grid. However, as discussed in 9.3, the constructs and elements of concern in the present study are of lower order than those associated with people, and thus there is less room for mis-interpretation than in the former case if elements are given and not elicited. The great advantage of giving elements is that grids can be compared more easily and the subject's attention can be focussed on the area of interest. This latter approach would be essential if grids were to be used at a more detailed level of design.

11. CONCLUSIONS

In any research having two parallel themes it is likely that one will advance more quickly than the other. It is also to be expected that, as in the present case, the aspect to develop most rapidly will generally be the one in which the supporting theory is more established.

Thus on the one hand it was possible to implement a fully documented computer-based management information system within the period of the research, despite a variety of practical problems attendant upon the exercise which, although they significantly affected the research in various ways, were in all other respects irrelevant to it. (Some of these problems are discussed in the Epilogue). Additionally the first detailed examination of the pattern of demand for the Company's individual domestic tube products was undertaken and a number of forecasting models for one of the high sales volume products were critically discussed. A preliminary examination of the Company stock levels indicated that significant savings could be potentially achieved if buffer stocks were related to the variability of demand.

The management information system was test run on thirteen weeks real data. On the basis of this and in consultation with the C.A.D. manager a number of modifications were made. The system then became the responsibility of C.A.D. (The most recent history of the system is described in the Epilogue).

On the other hand, work on the general study of the problems of implementation was slow to get under way because of the virtual absence of any previous work in this field. By the time a general system of studying the problems of implementation had been devised, implementation of the specific management information system was well advanced and development of the general method of investigation never overtook the practical work during the period of the research.

However the results obtained from development of the general method using C.A.D. managers as subjects, showed the repertory grid approach to be an efficient method of eliciting information for use in system design that, upon analysis, can reveal the subject's attitudes and expectations of his job and its associated tasks. It is suggested that this is a more systematic and inherently more consistent method of obtaining such information than the only alternative of unstructured interview, although the latter approach could be used as a useful adjunct to the basic grid method in checking and expanding upon points revealed in the grid analysis.

The results reported in this study by no means exhaust the potential uses of the repertory grid technique in system design. In particular, further research should explore the iterative use of the grid test to increasingly detailed levels of design as discussed in Chapter 10.

EPILOGUE

EPILOGUE

In this final passage a number of points are made that are more relevant to the I.H.D. Scheme under whose provisions the research was carried out than to the research itself. Since the research reported in Parts 1 and 2 of the thesis was one of the first to be registered on the scheme which itself is still relatively new, it seems appropriate to record these observations in the hope that they will be of value to the administrators of the scheme and future students alike. Finally a brief mention is made of the more recent history of the PSALM system.

(1) Selection of Project

A draft brochure received shortly before registering under the I.H.D. Scheme stated that projects would " ... involve applied research work in one of the sponsoring industrial organisations on a problem of considerable importance to that organisation". A subsequent brochure states "The project should be of practical importance to your company ... ".

However the requirement that the research be of importance to the company is, in many cases - including the case reported here, in conflict with the normal time span of 2-3 years implicit in work carried out for M.Sc. and Ph.D. awards. This is so because for many companies, a problem of perceived importance requires a solution in a much shorter time-scale than 2-3 years. Usually a period measured in months rather than in years is the time allotted to important problems, and manpower and money resources are injected until a solution can be achieved in that time. This point is well understood by management consultants who generally operate on a short intensive project basis.

A problem that is suitable for study by one person for 2-3 years is therefore unlikely to be of importance to many a company. Obviously important projects having longer timescales can and do exist in industry, but they are the exception rather than the rule and are very important indeed to have an "important" classification for such long a long period. The choice appears to be between carrying out research in short, intensive, possibly only loosely connected studies on problems important to a sponsoring company or alternatively carrying out longer term research on matters of secondary importance. If, as in the present study, the research is associated with an important area of the company's operations, then the research plan is liable to be subjected to large amounts of incidental interference as time goes by and the company's objectives alter.

(2) Mis-match of Industrial/University Activity Cycle times

This point is closely related to the preceding one. The reason that many companies restrict their activities to projects on relatively short time bases is because their own environment can alter radically in a fairly short time. On the other hand the University environment is more stable and its cycle time is relatively extended. In consequence the latter is more orderly and predictable.

In the present case the sponsoring Company's organisation was radically changed during the period of the research. General economic recession caused approximately 400 redundancies, including some very senior staff; the price of copper changed radically; metrication took place and by the end of the period plans were almost finalised for a complete re-organisation of the Company's structure into product divisions.

Clearly the possibility of these kind of events taking place in all but the largest organisations must be anticipated in any University scheme that is to successfully carry out research in industry.

(3) Conflicting Industrial/University requirements

The requirements of university and industry can differ radically in the context of research for higher degrees. The University looks for originality of thought and application, while industry frequently requires a purely practical - even pragmatic - solution to its problems. Thus when work is carried out for a sponsoring organisation, the best solution to achieve the objectives of the sponsor need not coincide with university requirements for an original, non-trivial solution. This may lead to a conflict of interests for the researcher that is hard to resolve.

(4) Recent History of PSALM

The research formally ended in December 1971 and the PSALM system was taken over by C.A.D. in January 1972 and run with occasional assistance from the author until he left the organisation in May 1972. According to ex-colleagues its use was discontinued shortly afterwards because of stock shortages and production difficulties (see below). C.A.D. approached the O.R. and Statistics Section of Management Services Department in October of 1972 asking for assistance to re-instate the system on the Company computer. However existing commitments in the section meant that no help was available and the idea seems to have been forgotten for some time.

In a recent conversation with the C.A.D. manager (March, 1974), he said that PSALM had not been forgotten and its use was occasionally reviewed. The last occasion it had been reviewed was in November 1973, but the then impending disruption anticipated as a result of the industrial action of mineworkers and the subsequent three-day working which severely affected production, once again led to deferment of the decision to use PSALM. He still felt that PSALM had potential but felt events had never stabilized sufficiently to justify its use. He saw no benefit in an allocation system of any kind when the basic problem was that sufficient production could not be achieved.

As a final comment, it is difficult not to sympathise with the C.A.D. manager's outlook but in many ways it is, and always was, symptomatic of the whole problem. By concentrating on the emergencies of the present of which there were many, and not looking far enough into the future, the allocators failed to prevent the embryo problems of tomorrow growing into sturdy infants. It is the author's sincere opinion that the discipline the use of a system such as PSALM would impose on the situation, despite its lack of sophistication, would enable the vicious circle of tomorrow's problems becoming today's,to be broken. Once that had been achieved the benefits of more sophisticated forms of control would be recognised and could be rapidly achieved.

REFERENCES

REFERENCES

Bainbridge, L., Beishon, J., Hemming, J.H. and Splain, M. (1968), "A Study of real-time human decision-making using a plant simulator", Operational Research Quarterly Special Conference Issue, pp. 91-106.

Bannister, D. and Mair, J.M.M. (1968), "The Evaluation of Personal Constructs", London and New York, Academic Press.

Bare, G.E. (1966), "The Measurement of Attitudes toward Man-machine Systems", Human Factors, 8,1, p 71-79.

Baumal W.J. (1961), "Economic Theory and Operations Analysis", New Jersey, Prentice-Hall.

Bowen, H.M. (1967), "The 'imp' in the system". In Singleton, W.T., Easterby, R.S. and Whitfield, D. (Ed.), "The Human Operator in Complex Systems", London, Taylor and Francis.

Box, G.E.P. and Jenkins, G.M. (1962), "Some statistical Aspects of Adaptive Optimization and Control", Jnl. Royal Statistical Society, Series B, 24, Section 8, pp. 297-343.

Box, G.E.P. and Jenkins, G.M. (1968), "Some Recent Advances in Forecasting and Control", Applied Statistics, 17, Section 12, pp. 91-109.

Box, G.E.P. and Jenkins, G.M. (1970), "Time Series Analysis, Forecasting and Control", San Francisco, Holden-Day.

Beer, S. (1959), "Cybernetics and Management" London, E.U.P.

Beer, S. (1966), "Decision and Control," London and New York, John Wiley.

Beer, S. (1967), "Management Science: The Business Use of O.R.," Aldus Books.

Beer, S. (1972), "Brain of the Firm", Allen Lane, The Penguin Press.

Brown, R.G. (1962), "Smoothing, Forecasting and Prediction of Discrete Time Series", New Jersey, Prentice Hall.

Burt, C. and Banks, C. (1947), "A Factor Analysis of Body Measurements for British Adult Males", Ann. Eugenics, 13, pp. 238-256.

Coffey, J.L. (1961), "A Comparison of Vertical and Horizontal Arrangements of Alphanumeric Material", Human Factors, 3, 1, p 93-98.

Edwards, W. (1954), "The Theory of Decision Making", Psychol. Bulletin, 51, p. 380-417.

Edwards, W. (1961), "Behavioural Decision Theory", Ann. Rev. Psychol., 12, p. 473-98.

Edwards, W. and Tversky, A. (1967), "Decision Making", Penguin Books.

Eilon, S. (1962), "Elements of Production", New York, Macmillan.

Flowers, B.H. (1969), "Science, Industry and Government", Nature, 222, 3rd May, p. 421.

Gagne, R.M. (1963), "Human Functions in Systems" In "Psychological Principals in System Development", (Ed) Gagné, R.M., Chapter 3. Holt, Rinehart and Winston.

Good, I.J. (1962), "How Rational Should a Manager be", Management Science, 8, 4, p. 383-393.

Grace, G.L. (1970), Editor's Preface to Special Issue: Human Factors in Information Processing Systems, Human Factors, 12, 2, p. 161-164.

Grant, E.E. and Sackman, H. (1967), "An Exploratory Investigation of Programmer Performance Under on-line and off-line Conditions", I.E.E. Trans. of Human Factors in Electronics, 8, p. 59-102.

Gustafson D.H., Edwards, W., Phillips, L.H. and Slack, W.V. (1969) "Subjective Probabilities in Medical Diagnosis", I.E.E. Trans. Man-Machine Systems, 10, 3, p. 61-65.

Hanlon, J. (1973), "Chile leaps into Cybernetic Future", New Scientist, p. 363-364.

Harman, H.H. (1967), "Modern Factor Analysis", Chicago and London, University of Chicago Press.

Hitt, W.D. (1961), "Development of Design Criteria for Intelligence Display Formats", Human Factors, 3, 1, p. 86-92.

Holt, C.C. (1957), "Forecasting Trends and Seasonals by Exponentially Weighted Moving Averages", O.N.R. Memorandum, No. 52, Carnegie Institute of Technology.

Holzinger, K.J. and Harman, H.H. (1941), "Factor Analysis", Chicago, University of Chicago Press pp. 122-136.

Interdisciplinary Higher Degrees Scheme (1971) "Notes for Guidance", University of Aston, April.

Jordan, N. (1962), "Motivational Problems in Human Computer Operations", Human Factors, 4, 3, p. 171-175.

Jordan, N. (1963), "Allocation of functions between man and machines in automated systems". Journal of Applied Psychology, 47, 3, p. 161-165.

Kaiser, H.F. (1960), "The Application of Electronic Computers to Factor Analysis", Educational and Psychological Measurement, 20, pp. 141-151.

Keeneg, R.L. (1968), "Evaluating Multi-dimensional Situations using Quasi-seperable Utility Functions", I.E.E. Trans. of Man-machine Systems, 9, 1, pp. 25-28.

Kelly, G.A. (1955), "The Psychology of Personal Constructs", Volumes 1 & 2, New York, Norton, 1955.

Kendall, M.G. (1945), "On the Theoretical Specification of Sampling Properties of Autocorrelated Time Series", Journal Royal Statistical Society, 108, p. 93.

Kendall, M.G. (1948), "Rank Correlation Methods", London, Griffin.

Kendall, M.G. and Stuart, A. (1966a), "The Advanced Theory of Statistics, Vol. 3, London, Griffin.

Kendall, M.G. and Stuart, A. (1966b), "The Advanced Theory of Statistics", Vol. 1, 8.3-8.4, London, Griffin.

Knowles, W.B., Burger, W.J., Mitchell, M.B., Hanifan, D.T. and Wulfeck, J.W. (1969), "Models, Measures and Judgements in Systems Design", Human Factors 11, 2, p. 577-590.

Leondes, C.T. and Mendel, J.M. (1969), "Artificial Intelligence Control", in "Survey of Cybernetics", Chapter 11, London, Iliffe. (Ed. J. Rose).

Licklider, J.C.R. (1960), "Man-Computer Symbiosis", I.R.E. Trans. of Human Factors in Electronics, 1, 1, p. 4-11.

Longbottom, D.A. (1972), "Decision Making - A Review", Management Decision 10, Winter, pp. 225-242.

Moroney, M.J. (1956), "Facts from Figures, "Harmondsworth, Penguin, p. 312-314.

Mosteller, F. and Nogee, P. (1951), "An Experimental Measure of Utility", Journal of Political Economy, 59, no. 5, pp. 371-404.

Nickerson, R. (1969), "Man-Computer Interaction: A Challenge for Human Factors Research", Ergonomics, 12, 4, p. 500-517.

Oldfield, L. (1970), Internal Report, Yorkshire Imperial Metals Ltd., Leeds, England.

Parsons, H.M. (1970), "The Scope of Human Factors in Computer-Based Data Processing Systems", Human Factors, 12, 2, p. 165-175.

Peace, D.M.S. and Easterby, R.S. (1973), "The Evaluation of User Interaction with Computer-Based Management Information Systems", Human Factors, 15, 2, p. 163-177.

Pearson, E.S. and Hartley, H.O. (1954), "Biometrika Tables for Statisticians", Vol. 1., Cambridge University Press.

Phillips, L.D., Hays, W.L. and Edwards, W. (1966), "Conservatism in Complex Probabilistic Inference", I.E.E.E. Trans. of Human Factors in Electronics, 7, p. 7 et seq.

Rosen, S. (1973), Correspondence - Scientific American, 229, 4, p.6.

Sackman, H. (1970), "Experimental Analysis of Man-Computer Problem Solving", Human Factors, 12, 2, p. 187-201.

Schone, M.L. (1967), "Exponential Smoothing with an Adaptive Response Rate", Operations Research Quarterly, 18, 3, pp. 318-319.

Shackel, B. (1969), "Man Computer Interaction: The Contribution of the Human Sciences", Ergonomics, 12, 4, pp. 485-500.

Shutz, H.G. (1961a), "An Evaluation of Formats for Graphic Trends Display", Human Factors, 3, 1, p. 99-107.

Shutz, H.G. (1961b), "An Evaluation of Methods for Presentation of Graphic Multiple Trends", Human Factors, 3, 1, p. 108-119.

Slater, P. (1965), "The Principal Components of a Repertory Grid", London, Vincent Andrews & Co.

Slater, P. (1967), "Notes on Ingrid '67", Biometrics Unit, Maudsley Hospital, London.

Tintner, G and Rao, J.N.K. (1963), "On the Variate Difference Method", Australian Journal of Statistics, 5, p106.

Trigg, D.W. (1964), "Monitoring a Forecasting System", Operational Research Quarterly, 15, 3, pp. 271-274.

Trigg, D.W. and Leach, A.G. (1967), "Exponential Smoothing with an Adaptive Response Rate", Operational Research Quarterly, 18, 1, pp. 53-59.

Von Neumann, J. and Morganstern, O. (1947), "Theory of Games and Economic Behaviour (2nd Ed), Princeton, Princeton University Press.

Wagle, B. (1969), "The Use of Models for Environmental Forecasting and Corporate Planning", Operational Research Quarterly, 20, 3, p. 327-336.

Ware, C.T. (1964), "Individual and Situational Variables Affecting Human Performance, "Human Factors, 6, 6, pp. 673-674.

Williams, J.D. (1954), "The Compleate Strategyst", New York, McGraw Hill Book Company.

Winters, P.R. (1960), "Forecasting Sales by Exponentially Weighted Moving Averages, "Management Science, 6, pp. 324-342.

Wulfeck, J.W. and Zeitlin, L.R. (1963), "Human Capabilities and Limitations", in "Psychological Principals in System Development", (Ed. Gagné, R.M.), Chapter 4, Holt, Rinehart and Winston.

Yntema, D.B. and Torgerson, W.S. (1961), "Man-Computer Co-operation in Decisions Requiring Common Sense", I.R.E. Trans. on Human Factors in Electronics, 2, no. 1, p. 20-26.

Zobrist, A.L. and Carlson, F.R. (1973), "An advice-taking Chess Computer", Scientific American, June.

APPENDICES

APPENDIX 1

The purpose of the simulation model was to simulate an integral number of years' activity at weekly intervals. In this way the effect of various policies could be explored and limits of statistical confidence could be placed on the variables of interest.

Before starting the simulation proper, the weeks the four factories would shut down for plant maintenance and works' holidays had to be known in order to calculate "holiday stocks". It was a policy of the company that there should be in stock at the start of an annual factory shutdown an amount equivalent to the output that would have been expected had the factory remained open. This "holiday stock" was quite separate to stocks held to meet variations in demand.

To establish what the factories would have made had they been open, the demand in each week that a holiday occurred for one or more of the factories was found. This was calculated as one thirteenth of the forecast demand for the quarter in which the week occurred. This quantity was allocated by Monte Carlo simulation to the four factories in the manner described below. The total production from the factory on holiday, measured in tons, was divided by the number of weeks from the beginning of the year to the start of the holiday period. This gave the holiday increment that had to be made each week so that at the start of the holiday the total holiday stock for the given factory would have been produced. This was repeated for each factory.

The first step in the week by week simulation, was to generate actual demand values for each of the products by Monte Carlo methods. The method used was similar to that described below for simulating the product mix. When the individual product demand values had been generated, stock levels were reduced by a similar amount and tested to check whether they had fallen below the minimum level specified. In those cases where stock had fallen below the minimum level, plans were made to produce more of the items in short supply the following week at the appropriate factory.

The total stock at the start of the following week was then calculated by adding to the total stock at the start of the current week the production in the current week and subtracting the demand. Next the expected stock at the end of the year was calculated, and the production level for the following week(s) was adjusted so that all excess stock would be used by the end of the year. An adjustment to the production level was only necessary when forecast demand and actual demand did not coincide. Although this was likely to happen the amount of adjustment was expected to be small.

Having established the production level for the following week, production for the current week was broken down by simulation to give production output for each product. The amount generated by simulation in this way was not the total to be produced, but was the production level calculated the previous week (in an identical manner to that described above) plus the holiday increment also described previously, less the amount made to supplement stocks depleted in the past week. The latter quantity was not simulated at random across the factories but was added to the output of the appropriate factories when the bulk of production had been simulated, since its factory of manufacture was known from the allocation rules. Thus the rule was to make items specifically required at the appropriate factory, while the bulk of production arising from the forecast of demand was simulated at random across the factories.

In this method the week's product mix reflected the distributions on which the simulations were based. These distributions were constructed from the records of production allocation for 1968. The method of simulation is indicated in fig (1) which shows the cumulative probability of making an amount 'x'. The simulation was carried out by generating a random number in the range 0 - 99 and applying this to the ordinate scale in the usual way. Where the number fell between two points as in fig (1)

METHOD OF SIMULATION

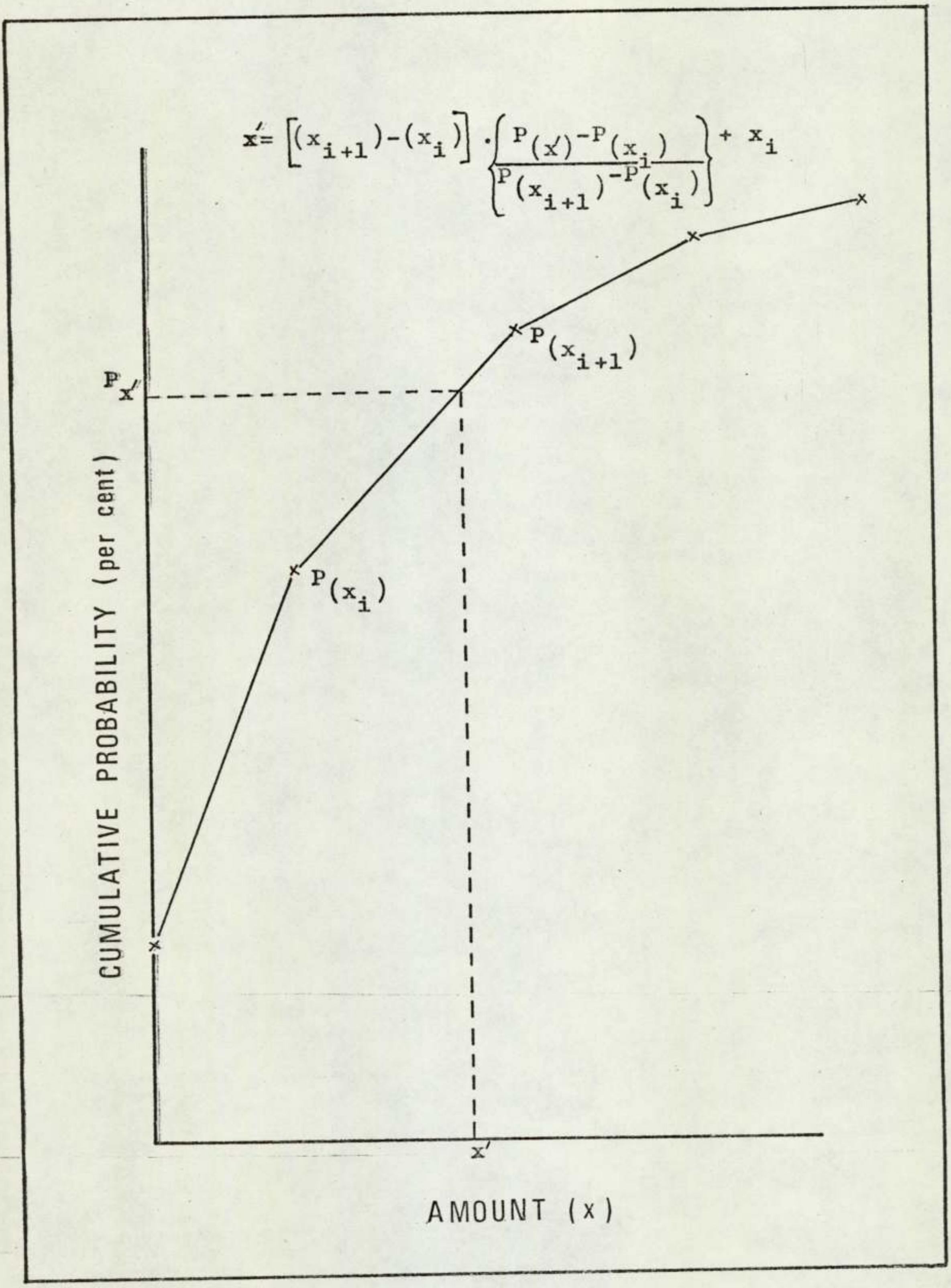

$x' = \left[(x_{i+1})-(x_i)\right] \cdot \left\{\frac{P_{(x')}-P_{(x_i)}}{P_{(x_{i+1})}-P_{(x_i)}}\right\} + x_i$
$P_{(x_{i+1})}$
$P_{x''}$
$P_{(x_i)}$
CUMULATIVE PROBABILITY (per cent)
x'
AMOUNT (x)

FIG (1)

the amount produced was calculated by interpolation. The amounts of the product types made in real life was approx. 400, 25 and 5 - 10 tons per week for Table 'A', B.S. 3931 and B.S. 1386 respectively. To reproduce these proportions the model relied on the production allocation distributions. It should be noted that simulating using these distributions, which were available for each factory, not only generated the amount of each product made but also indicated the factory of manufacture.

When production had been allocated in this way the extra production for stock replenishment was added where necessary and the total make of each product was then added to stock. A check was made to see whether any maximum stock levels had been exceeded and when this occurred the excess over the maximum level was accumulated. When all products had been checked in this way the accumulation of excess was re-allocated again by simulation, markers being automatically set to prevent more of those products already at maximum level being made. A new check of the levels was made and the procedure was repeated until all the required production had been made and stocked satisfactorily.

Finally a full breakdown of the production by product type and by site was output, followed by the new stock levels and records of actual demand. At the end of the year's simulation a graphical summary of total weekly production and stock could be output if required.

COMPUTER PROGRAM MANUALS

Contents:-

APPENDIX M1

Foreword: There is certainly more than one solution to the production smoothing and allocation problem. This manual describes one such solution, that implemented in the PSALM package. Although this approach is considered best for the present, future changes in policy, better information and so forth will certainly justify changes to, and ultimately the complete replacement of, the procedures described herein.

PSALM

(Production Smoothing and ALlocation Model)

User Manual

Author: D. M. S. Peace.
April, 1972.

CONTENTS

1. INTRODUCTION

This manual gives a general description of the capabilities of the PSALM package and specific details for its use. It is thus an instruction manual and the user's operating manual. In both capacities it is written for non-computer specialists and is in no way a technical specification, which will be produced separately.

The package is concerned with the problem of smoothing and allocation of production of construction copper tubes. In particular it considers the Table 'X', Table 'Z' and Table 'Y' product groups. As it is assumed throughout that the nature of the problem is understood, the latter is described here only in relation to the particular solution offered by PSALM.

There are four main sections to the manual:-

(i) Description of Procedures. The main program of the PSALM package is run at weekly intervals and performs the smoothing and allocation function by utilising six major procedures which are described in sections 2.1 - 2.6.

(ii) Commentary on Output. Salient features of output associated with the procedures described above are commented on. The output described in the commentary is presented in sub-sections that correspond to the procedures to which they relate. Thus the output described in 3.1 would be produced by the procedure described in 2.1.

(iii) Details of Input. The make up of weekly input is given. The method of coding the input data is described and the function of the identification codes is outlined.

(iv) Changing the Input File. PSALM uses information kept on a magnetic tape called the input file. It is explained in (i) above that changing the contents of this file is an integral part of using the package. The method of doing this using the PSALM File Editor is described.

2. DESCRIPTION OF PROCEDURES

This section describes in principle how the various operations performed by PSALM are carried out. Its purpose is to enable the user to understand how his output was derived, and thus to enable him to assess in a particular decision making situation, how much weight he will give to the information contained therein.

Time Scale

The time scale PSALM works within is shown in Fig. 2.1

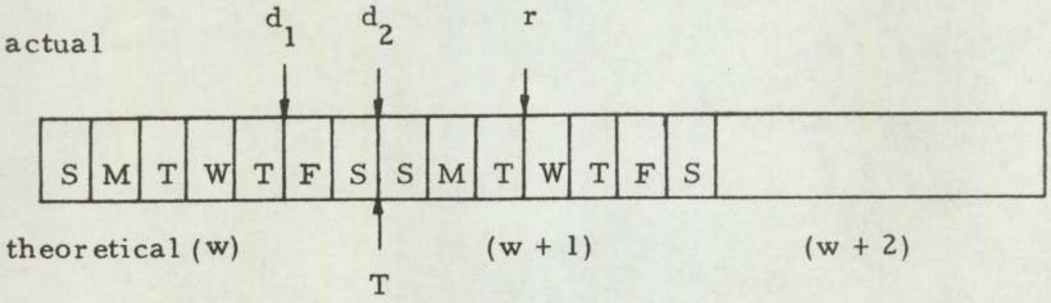


Fig. (2.1)

Despatch and stock information is collected during week (w) between days (d_1) and (d_2) at depots and factories and is sent to Leeds where it is coded and punched for PSALM. The PSALM main program is run at time (r). This manual adopts the convention that the PSALM main program is run at time (T) so producing an allocation for week (w + 2). Demand and stock data output thus relates to week (w).

2.1 DATA INPUT PROCEDURE (with Stock and Production Summary Listing)

This procedure is essentially concerned with reading the weekly punched card input, checking it for certain errors and printing out the information in tabular form (3.1). Additionally the current stock levels are checked against minimum and maximum levels and where these are broken, the actions described below are taken.

2.1.1 Card Input

The program requires a fixed number of cards as input containing the same categories of information in the same order each week. In order that the program can check the input is in the correct order each card is punched with identification characters and a sequence number, for example, * PLY 2. These characters are also intended to be meaningful to the user in order that he can readily identify the values on a given card. Thus * PLY 2 would indicate that the values punched in the card referred to production at Leeds site of Table 'Y' and that the card was the second in the sequence of cards having the identification characters * PLY. (The asterisk '*' serves a technical purpose that will not be described, but its presence is essential). Complete details of input punching codes and requirements are given in section 4.

If the program finds a card which is out of sequence or contains the wrong identification characters, it will halt with a message indicating the characters read and the characters it expected. The user must then check the deck of cards which have probably been inadvertently shuffled. A similar message is output if too few entries have been made in a card. In each case the type of error is indicated to avoid confusion.

2.1.2 Stock Level Check and Re-ordering.

If card input has been error free the current stock levels (which have been input as data - section 4) are checked against maximum and minimum levels which are kept on the input file (for details of the input file see section 5). A table of maximum and minimum stock levels used in this procedure is given in Appendix 1.

If current stock of a given item is less than the minimum stock level, the difference between the minimum and current stock level is calculated and compared against that on order in the coming week. (Since there is a fortnight lead time, the production make for the coming week has been in existence since the previous week). There may of course be no order currently raised for the particular item, or it may be insufficient to cover the deficit. In either case the quantity required to bring current stock back to the minimum level at the end of the coming fortnight, assuming no demand in that period, is calculated.

The minimum re-order amount is 10 tons and thus items with stock deficits of less than this amount will be lifted above the minimum stock level at the end of the lead time if no demand occurs. If, on the other hand, the current deficit is met by orders raised on the coming week, no further order will be raised.

With regard to maximum stock levels, items with stocks in excess of these are marked for later consideration (see section 2.4).

2.2 STOCK PROJECTION PROCEDURE

The output from this procedure is a graph of past and projected total Company stock levels. This section describes how this is achieved by production smoothing (2.2.1), and how the projection may be changed by the user to avoid unacceptable short term stock fluctuations (2.2.2) and to give load balancing (2.2.3).

2.2.1 The Basic Stock Projection

The graph contains a plot of three variables against week number,

(i) Demand (measured as despatch)

(ii) Weekly Production

(iii) Total Company Stock

The above quantities are plotted for a period extending from the "base" week to the "horizon" week, a period defined by the user. Thus a "base" week might be the first week of a quarter and the "horizon" week might be the last week of the quarter. In general the "horizon" week is the farthest convenient point in the future about which forecasts of demand can be made. Initially the whole "horizon" period is considered. The total weekly demand expected in the period is summed to give "total future demand". In order to produce a smooth production level throughout that period, the total future demand is divided by the number of weeks in the period thus giving a smooth weekly production level. By assuming the forecasts of total weekly demand are accurate, and knowing the smoothed weekly production level, it is possible to project the stock levels into the future. This is done by adding to the stock the difference of weekly demand and smoothed weekly production, for each week of the "horizon" period. These values can then be plotted against the week number to which they apply.

As time goes by the current week will become intermediate between the "base" and "horizon" weeks. For the weeks between the "base" week and the current week, it is no longer appropriate to show projected figures as they have been replaced by actual ones. Thus in the output from this procedure actual values of weekly demand, weekly production and weekly stock are plotted up to and including the current week, and forecasts and projections are shown beyond.

Although the above describes the essential features of this procedure, in fact there are three further considerations. Firstly because of the lead time mentioned in (2. 1. 1), it is not possible to smooth production in the manner described above for the coming week - week (w + 1) - only for the weeks after it. This is because production schedules for the coming week have been prepared on the basis of smoothed production calculated the previous week - week (w - 1) - and in general the latter will not be the same as that currently calculated.

The reason for this arises because each week actual demand and production seldom coincide with the values forecast and requested respectively. This in turn means that the stock does not attain its projected level. However the basis of production smoothing is that production in the period considered - "base" week to "horizon" week - shall be equal to demand in the period (with important qualifications discussed in 2. 2. 2). Since present stock in combination with future production must meet future outstanding demands, deviations from projections of stock will require adjustments to future production levelsif demand is to be matched by production.

The second consideration is that of buffer stock. The errors inherent in forecasts of demand require that safety stocks be kept for contingency purposes. With this in mind the rule used for production smoothing may be stated as:- Adjust production levels with regard to future demand and current stock so that at the end of the period a suitable buffer stock will remain.

The actual formula used is:-

$$\text{Smoothed Production} = \frac{(\text{Total Future Demand}^{*} - (\text{Stock projected for end of coming week} - \text{Buffer Stock}))}{(\text{Horizon wk} - (\text{Current week} + 1))}$$

* This is the total demand expected in remainder of period (i. e. up to and including the horizon week) LESS demand in the coming week which is accounted for in "stock projected for end of coming week".

Thirdly, the smoothed production level as calculated above is only permitted to vary within certain limits related to the total weekly production budget. Although the actual values might be re-set from time to time typical limits would be to permit the smoothed production level to range between 85-125% of the total weekly budget. If the smoothed production level as calculated was outside this range, it would be re-set to the nearest of the two limits. In this case the objective of holding only buffer stocks at the end of the period would not be achieved in the projection.

2.2.2 Adjusting the Stock Projection.

In the long term, production must approximately equal demand or the Company will either run completely out of stock or accumulate huge surpluses. In the short term however it is often perfectly acceptable to carry larger stocks than is otherwise desirable when the alternative is to dismiss part of a skilled work force. This of course would assume that in the long term, demand is expected to be adequate to sustain existing manning levels. On the other hand when the long term demand forecast is low and production is smoothed to a correspondingly low level, an isolated period of increased short term demand could deplete stocks to the extent of leaving them dangerously low for a short time.

Thus it is of great importance that, in the long term, production matches demand. It is of equal importance to see the short term implications in terms of stock and to take timely action to avoid potential crises. It was explained in the preceeding section how PSALM gives a projection of stock levels that are consequent on adoption of a smoothing principle aimed at matching production to long term demand. This projection will show any short term variations to be expected on the basis of information currently available. The next step is to take preventive action.

PSALM permits this by allowing the user to modify the smoothed production in any future week, with the exception of that coming where the schedules are already fixed. The method of doing this is best understood by considering the example below which shows a typical situation requiring some alteration of the smoothed production level.

Example

Table 2.2.2 shows a situation after smoothing production with no adjustments. In the period considered and with the pattern of demand forecast, stocks will fall below the buffer stock level (900 tonnes) at a rate of 20 tonnes per week starting at week 9 and not rising above this level until week 16. It is assumed in this example that long term production and demand are perfectly matched. It can be seen that a short term stock crisis will occur if some action is not taken.

Table 2.2.2

Week No.	Smoothed Production (tonnes/week)	Forecast Demand (tonnes/week)	Stock * (tonnes)	
9	420	440	900	
10	420	440	880	
11	420	440	860	
12	420	440	840	
13	420	440	820	
14	420	393	847	
15	420	393	874	* see overleaf
16	420	393	901	

* Buffer Stock level = 900 tonnes

In the above, the greatest deficit expected is 80 tonnes in week 13, which could be avoided by making, say, an extra 8 tonnes per week for a period of 10 weeks prior to week 13 where the minimum occurs. This would have to be balanced by reducing the make in subsequent weeks in order that the long term balance would be maintained. In the above case this could be done by reducing the smoothed production level by 8 tonnes per week to 412 tonnes per week on say the fourteenth week. In this type of situation the nett adjustment should be zero.

Adjustments are made to smoothed weekly production by adding or subtracting the required amounts to the smoothed production calculated for the required week(s). The final production projection for a particular week is thus the smoothed production level, which is the same for all remaining weeks in the period, plus the adjustment for that week. The adjustments can take any value and are added to the backing file using the PSALM File Editor in a separate operation to the weekly run. On the file are stored adjustments for each week of the horizon period. If the smoothed production level for a given week is acceptable, the adjustment will be zero. Once adjustments have been made they will apply each week until changed by the user.

As explained in(2.2.1)the smoothed production level will tend to vary each week and this will cause the effect of the adjustments to be correspondingly changed. As the adjustment facility is to be regarded as a fairly coarse form of control, minor changes in production level should not affect the overall situation, but major changes in level developing over a period of time will obviously require the adjustments to be re-set. Adjustments on the backing file relating to the coming, current or past weeks do not affect the output.

2.2.3 Load Balancing

It is a feature of the products monitored by PSALM that their volume of production is to some extent controlled by the level of activity of the company in respect of its other products. As a matter of policy it may be decided to either generally over produce or under produce Table 'X', Table 'Y', and Table 'Z' in order to either occupy the existing workforce or to release machines for manufacture of different products, respectively. This situation can be controlled by using the adjustment facility described in (2.2.2). Here the nett adjustment will not be zero as in the example of (2.2.2) since the load balancing policy automatically results in stocks that would deviate from the simple smoothing situation described in (2.2.1).

2.3 ALLOCATION PROCEDURE

The basis for allocation is the weekly production budget published in advance of each quarter. This provides a weekly production budget ($B_{p,f}$) for each product (p) at each manufacturing factory (f), where f = 1, 2, 3, 4. Thus the total weekly budget (W) is the sum of all the $B_{p,f}$ which in this procedure determine the weekly product mix. However the weekly production volume depends on the smoothed production (S) calculated for that week. Thus the "smoothed" weekly budget ($B'_{p,f}$) is obtained from $B_{p,f}$ by proportion, thus,

$$B'_{p,f} = B_{p,f} \cdot \frac{S}{W} \quad \text{--------} \quad (1)$$

since,

$$(B'_{1,1} + B'_{1,2} \text{ ---- } B'_{1,4} + B'_{2,1} + B'_{2,2} \text{ ---- } B'_{p,f}) =$$

$$(B_{1,1} + B_{1,2} \text{ ---- } B_{1,4} + B_{2,1} + B_{2,2} \text{ ---- } B_{p,f}) \cdot \frac{S}{W}$$

and $(B_{1,1} + B_{1,2} + \text{ ---- } B_{p,f}) = W$ by definition;

$B' = (B_{1,1} + B_{1,2} + \text{ ---- } B_{p,f}) = S$, the smoothed weekly production for the given week.

The "smoothed" budgets $B'_{p,f}$ form the basis of the allocation, but they may be modified in two ways.

Firstly if a re-order of amount (R) has been raised to replenish stocks as described in (2.1.2), a check is made to see if the sum of the smoothed budgets B'_p will cover the re-order amount,

where $B'_p = B'_{p,1} + B'_{p,2} + B'_{p,3} + B'_{p,4}$

If B'_p is greater than or equal to R, the re-order amount, no further action is taken. If not, B'_p is further adjusted in the proportion (R/B'_p). Thus for product (p)

$$B''_{p,f} = B'_{p,f} \cdot \frac{R}{B'_p} \quad \text{for } f = 1, 2, 3, 4.$$

where $B''_{p,f}$ is the final production request for product (p). It can thus be seen to be basically the budgeted amount adjusted for changes in smoothed production level and possibly further adjusted for re-order quantities to replenish stock below the minimum level.

The second modification arises if a re-order quantity has been raised for an item for which no production is budgeted ($B_{p,1}$; $B_{p,2}$; $B_{p,3}$; $B_{p,4} = 0$). Under these circumstances the whole of the re-order quantity is raised at the factory of first preference for that product. A table of the factories of first preference for the manufacture of each product covered by PSALM is given in Appendix 2.

On the other hand, if the check on stock levels described in (2.1.2) revealed stocks in excess of the maximum level, any production of the items concerned would be set to zero.

At this point the allocation request is complete providing that in total (now including re-order amounts) it does not exceed the limits placed on the smoothed production level (see section 2.2.1). If this has happened a final proportional adjustment is made analogous to that shown in equation (1) of this section to bring the total allocation to the nearest of the two limits on the smoothed production level.

2.4 FORECAST MONITORING PROCEDURE

The allocation procedure described above relies heavily on the quality of the production budget to produce meaningful results. The budgets in turn depend upon the accuracy of forecasts of demand. If as time progresses the latter should prove to be inaccurate, it may prove necessary to change the production budget. Unfortunately until a large discrepancy develops it is not easy to decide by simple comparison of actual and forecast values, when a forecast has become completely erroneous. This is because even a good forecast will deviate slightly from the actual and thus some errors will always exist, the problem is to how to decide when this has become so large that the forecast must be regarded as being wrong. As demand invariably contains random variation, decisions to abandon the current forecast are best made on a statistical basis. A number of techniques exist to do this, the one used in this procedure is known as a "Cusum Technique". The principle of this technique is described in (2.4. 1) and it is recommended that this is read to aid in interpreting the results. However as described in (3. 4. 1) it is possible to follow the recommendations of the program without further consideration of the method by which they were achieved.

2.4.1 Principle of the Cusum Technique

The principle of the technique depends on the fact that if forecasts of demand are "accurate", the minor errors that occur, if added on a weekly basis, will form a cumulative sum whose value is never far from zero, since a positive error one week will be offset by a negative value in a subsequent week. If, however, the forecast begins to deviate in a systematic way from the actual, the cumulative sum (cusum) will grow increasingly in a positive or negative sense until it exceeds some specified value. It is then deemed to have gone out of control and the forecasts from which it was derived can be said to be inaccurate. The limit on the controlling size of a cusum is determined by the inherent variability of demand and the amount by which the forecast can be allowed to deviate from the actual before it is unacceptable.

This may best be demonstrated by considering an example. Typically the demand for one of the products covered by PSALM is 120 tons/week with a standard deviation of 40 tons. This means that demand each week is regularly observed to range between 80 and 160 tons and often over a greater range. If the mean demand were to change to 100 tons/week with the same standard deviation,much of the new pattern of demand would fall in the same range as the old, that is between 80 and 160 tons/week, and it would be some time before visual inspection of the results enabled a fall in mean demand of 20 tons/week to be perceived. The cusum method of monitoring would be much more sensitive in detecting such changes in mean demand because it would naturally take into account the 40 ton standard deviation in setting its control limits.

As with all statistical control techniques, there is a small known probability that the cusum technique will indicate that the forecast is in error where in fact it is not, and a correspondingly slight chance that the forecast will be shown to be acceptable when in fact it is not. In practice things are arranged so that this does not seriously affect the utility of the scheme. In the present case, the forecast of total demand would be wrongly shown to be in error once every two years when it is set to detect a genuine error of 20% between forecast and actual demand. The probability of obtaining an erroneous result of the kind just described depends on the size of genuine deviation of actual demand from that forecast it is desired to detect. In general the greater the variability and the smaller the genuine deviation that is to be detected, the more likely is such an error.

If the above considerations are bornein mind they should help in the week to week use of the program. The controlling limits for the cusum analysis are not the responsibility of the user. Their calculation and insertion on to the input file at the start of each quarter is done as part of the technical support required by the package. A cusum analysis can be carried out for each individual product where a forecast exists, and for the total forecast of demand. Further information on cusum techniques may be found in ref. 1. The cusum procedure implemented in PSALM follows the scheme outlined on page 60 of that reference.

2.5 PRODUCTION MONITORING PROCEDURE

This and the final procedure (2. 6) only produce print outs on the last week of a quarter.

Each week the production realised at each factory is compared with the production budget (i. e. $B_{p,f}$ in the notation of section 2. 3) and the production request (i. e. $B'_{p,f}$ or $B''_{p,f}$ as appropriate). Re-orders to replenish stocks of products not appearing in the production budget ($B'_{p} = 0$) are ignored in this analysis. The differences between actual production and production budget, and between actual production and requested production are plotted as two series against week number.

When the production budget is consistent with the smoothing policy described in (2.2) the production requests will not not markedly deviate from the budget, and the plots of the two series will be essentially similar. If because of the smoothing function, the two do differ appreciably it will be for the user to decide which under the circumstances is the most valid for his purposes.

2.6 DEMAND STATISTICS PROCEDURE

At the end of each quarter, this procedure produces a listing of the mean demand, variance and standard deviation of demand for each product covered by the package. It should be noted that the variance of total demand is calculated separately and does not assume independence between the demands for individual products. In general, therefore, the sum of the variances of demand for individual products will not be equal to the variance of total demand.

3. COMMENTARY ON OUTPUT FROM PSALM MAIN PROGRAM.

This section should be read in conjunction with a copy of the weekly printout produced by the PSALM Main Program, which ideally should be for the end of a quarter (week Nos. 13, 26, 39 or 52) in order that the full output is available.

Since the major procedures have already been described (Section 2) most of the output will require no further explanation. The purpose of this section is to cover those points arising out of the presentation rather than the related procedure. As indicated in Section 1 the sub-sections of this and the previous section correspond, although not all sub-sections require comment and are therefore empty.

One general point is made before proceeding to the sub-sections.

Page Headings

At the head of each page reading from left to right, appear the week number, the time the program was run on the computer, the program title, the date of the run and the page number.

The time framework the main program works within has been described in 2.2 and using the terminology defined there, the week number on the page headings is week 'w'.

The purpose of the time and date is to uniquely define the particular run, as more than one run may be obtained on one day when the program is used to evaluate the effect of different patterns of forward loading.

3.1 DATA INPUT PROCEDURE - COMMENTARY

A single asterisk at the side of closing stock levels indicates that stock is below the minimum value shown to the left of the entry. Similarly two asterisks indicate that stock is in excess of the maximum value shown to the right of the entry.

Information pertaining to Table 'Y' relates to material in the soft condition.

3.2 STOCK PROJECTION PROCEDURE - COMMENTARY

The output from this procedure is graphical in form. The graph is set out at right angles to the print so that its length (i. e. the number of weeks covered) is not restricted. It is best studied by turning the print-out on its side, so making the week number (abscissae) axis horizontal. The units of the vertical (ordinates) axis in this position are tonnes. Both axes (abscissae and ordinates) may require multiplication by scaling factors of 10; 100 or more as indicated by the labels. Thus,

(ORDINATE VALUES X 10EXP 2)

means all vertical scale values must be multiplied by the factor 10^2 or 100. (N. B. 10EXP 0 or 10^0 means that the scaling factor is unity - the scale of the graph is as marked on the axes. When the exponent is negative, for example 10EXP-2, the scale values must be <u>divided</u> by 10^2).

When the range of values to be plotted is large, the scale must become compressed in order that the graph can be contained on the paper. Under these circumstances a single scale division can represent perhaps 10 or 100 tonnes. In order that the exact values of variables plotted are known, these are printed out before the graph, under appropriate headings. Each value is preceded by its sequence number and this pair of numbers is enclosed in parentheses. A typical entry would be,

(4, 423. 6)

the sequence number being 4 and the value being 423. 6. The sequence numbers are merely the week numbers in a different guise, and are obtained by renumbering the week numbers from unity in increasing order. Thus in the 4th quarter of a year the week numbers of interest would be,

40, 41, 42, ------ 52

and renumbering these from unity would give,

1, 2, 3, ------ 13

the sequence numbers.

To relate the listing of actual values to the plotted values it will be seen that the first value plotted (lowest week number) has sequence number one, the second has sequence number two and so on.

If the future nett adjustments are not zero, a warning will be printed to this effect after the projections. A load balancing situation will have been created as described in section (2.2.3.)

3.3 ALLOCATION PROCEDURE-COMMENTARY.

3.4 FORECAST MONITORING PROCEDURE-COMMENTARY

3.4.1 Output from Cusum Procedure

The status of the forecast is given as "in control", "under review" and "out of control". In the first case no action need be taken. In the second again no action need be taken but there is evidence that the forecast MAY be going out of control. The user may desire to investigate the bases of the forecast and whether they might have substantially changed, if it is important that the very first indications of the forecast going out of control are detected. If this procedure is adopted it is stressed that on many occasions the investigation will be fruitless and subsequently the forecast will revert to the "in control" status. It is, therefore, a procedure only to be adopted when early warning is of paramount importance and so justifies a large proportion of extra effort.

In the case of the "out of control" status being indicated, the forecast has been found to be inaccurate and should be abandoned. (The comments in section 2.4.1. with regard to the statistical basis of this recommendation should be noted). A plus sign in this columm indicates that actual demand is in excess of the forecast and a minus sign indicates that actual demand is less than forecast.

3.5 PRODUCTION MONITORING PROCEDURE - COMMENTARY

Two graphs are plotted for each manufacturing factory. One is of the differences between production budget and actual production, the other of the differences between the allocation requests and the actual production.

If the production budgets are realistic the allocation requests should be similar to the budgets and the factories might expect not to deviate greatly from zero on either standard. Certainly the deviations from the budget should not be systematic unless they have been in the direction of the allocation which itself has differed markedly from the budget.

On the other hand, if the allocation requests are not greatly different from the budgets, the factories should be able to meet them. Systematic failure to do so makes attempts to correct minor deviations from the budgets quite abortive.

3.6 DEMAND STATISTICS PROCEDURE - COMMENTARY

4. PSALM INPUT REQUIREMENTS

This section gives detailed information concerning how data must be prepared for the PSALM main program. The layout of numeric data is considered first (4. 1), and then the order in which these data are required for the various categories (eg. demand, production at each manufacturing site, etc) of input (4.2). Finally the card sequencing system is described and discussed (4. 3).

The above is summarised in the "Input Check List" (4.4).

4.1 LAYOUT OF NUMERIC DATA

Data for input to the PSALM main program are first written on special coding sheets which are then used to prepare punched cards. Each line on the coding sheet corresponds to a single card and each division of the line a single column of the card. Thus there are 80 divisions of a line corresponding to the 80 columns of a computer card.

PSALM will accept as input integer numbers (eg. 26, 343) or decimal fractions (ie. 26. 3, 343. 8) which obviates the need to write quantities such as 27 in the form 27. 0.

It is usual to define "number fields" on punched cards within which the computer expects to find certain numbers punched. For example card columns 1-8 of a given card might be expected by the computer to contain a code number, and inadvertently entering the code in columns 2-9 would give rise to errors which might go undetected. To avoid this type of error PSALM does not use fixed field definitions as described above. Instead it merely requires a certain number of fields to be present on each card. In this scheme, a field is defined by a series of numbers (eg. 2314 or 261.78) the field being terminated by the first blank column encountered.

The input program requires seven fields on each card with the possible exception of the last where less than seven fields are permitted if the total number of data being input is not a multiple of seven. Thus where 15 entries are to be input, the first two cards contain seven entries and the last only one. From the foregoing it will be seen that the seven fields could be punched in a given card separated by only one blank column and it would be acceptable to the program. However because the fields vary considerably in length from one entry to the next such a procedure produces a coding sheet that is difficult to check and to punch. Both these factors can give rise to errors.

For these reasons the PSALM coding sheets are organised so that each field begins in one of columns 1, 11, 21 ------ 61. This results in an orderly arrangement of data that are "left justified". These are easy to visually check, punch and verify.

A final point concerning number fields is that each product group covered by PSALM contains 15 products, so requiring 15 number fields. It is stressed that an entry must be made for each product even when it is zero, failure to do so will result in a data error as described below (4. 3) when the program is run. Number fields corresponding to products in a given product group are in increasing size order of products. Thus the first number field corresponds to 6mm size, the fifth for example, to 15mm and the fifteenth field to 159mm. This is summarised in the table below:-

Table 4. 1

Product Size (mm)	Field No.	Card No.
6	1	1
8	2	1
10	3	1
12	4	1
15	5	1
18	6	1
22	7	1
28	8	2
35	9	2
42	10	2
54	11	2
76. 1	12	2
108	13	2
133	14	2
159	15	3

4.2 ORDER OF INPUT BY CATEGORY

The PSALM main program requires a fixed amount of input each week, which may be broken into categories. These categories are listed in Table 4.2.1.

Table 4.2.1.

Order of Input	Category
1	Week Number.
2	Demand in above week.
3	Production in above week at Leeds Works.
4	" " " Kirkby Works.
5	" " " Barrhead Works.
6	" " " Allen Everitt Works.
7	Stock at end of above week.

With the exception of the week number, each category is further subdivided into product groups. Within a category the order of product groups is always as given in Table 4.2.2.

Table 4.2.2.

Order of input within categories	Sub-Category (Product Group)
1	Table 'X'
2	Table 'Z'
3	Table 'Y'

Within a product group data is set out as described in Section 4.1.

4.3 CARD IDENTIFICATION

In the process of coding, punching, verifying and ultimately feeding the data cards into the computer, a number of errors can arise that may or may not go undetected by the computer. These errors may be summarised as follows:-

(i) Errors in the values coded.

(ii) Errors in the number of fields per card.

(iii) Errors in the order of card input.

Although nothing can be done in the present case to detect errors of type (i), PSALM contains input checking procedures that are capable of detecting certain errors of types (ii) and (iii). To do this each card is punched with identification characters in columns 75-80. These enable the program to detect,

(a) too few numeric fields in a particular card - type (ii) errors.

(b) cards in the wrong order for input - type (iii) errors.

Although the program does not check explicitly for too many numeric fields per card, this situation implies too few elsewhere, a condition which will be detected.

In more detail, the identification characters are comprised of two parts, a mnemonic and a sequence number. The mnemonic code relates to the sub-category and the sequence number maintains the input order within the sub-category.

4.4. INPUT DATA CHECK LIST

Table 4.4 summarises the contents of sections 4.1 - 4.3.

Table 4.4. Input Data Check List.

Card No.	No. of Numeric Fields	Contents	Product Type	Size Range (mm)	Identification Code	Sequence No.
1	1	Week No.	-	-	* WKN	1
2	7	Demand	Table 'X'	6-22	* DMX	1
3	7	"	"	28-133	* DMX	2
4	1	"	"	159	* DMX	3
5	7	"	Table 'Z'	6-22	* DMZ	1
6	7	"	"	28-133	* DMZ	2
7	1	"	"	159	* DMZ	3
8	7	"	Table 'Y'	6-22	* DMY	1
9	7	"	"	28-133	* DMY	2
10	1	"	"	159	* DMY	3
11	7	Prdn. Leeds	Table 'X'	6-22	* PLX	1
12	7	"	"	28-133	* PLX	2
13	1	"	"	159	* PLX	3
14	7	"	Table 'Z'	6-22	* PLZ	1
15	7	"	"	28-133	* PLZ	2
16	1	"	"	159	* PLZ	3
17	7	"	Table 'Y'	6-22	* PLY	1
18	7	"	"	28-133	* PLY	2
19	1	"	"	159	* PLY	3
20	7	Prdn. Kirkby	Table 'X'	6-22	* PKX	1
21	7	"	"	28-133	* PKX	2

Card No.	No. of Numeric Fields	Contents	Product Type	Size Range (mm)	Identification Code	Sequence No.
22	1	Prdn.Kirkby	Table 'X'	159	* PKX	3
23	7	"	Table 'Z'	6-22	* PKZ	1
24	7	"	"	28-133	* PKZ	2
25	1	"	"	159	* PKZ	3
26	7	"	Table 'Y'	6-22	* PKY	1
27	7	"	"	28-133	* PKY	2
28	1	"	"	159	* PKY	3
29	7	Prdn. B'head	Table 'X'	6-22	* PBX	1
30	7	"	"	28-133	* PBX	2
31	1	"	"	159	* PBX	3
32	7	"	Table 'Z'	6-22	*PBZ	1
33	7	"	"	28-133	* PBZ	2
34	1	"	"	159	* PBZ	3
35	7	"	Table 'Y'	6-22	* PBY	1
36	7	"	"	28-133	* PBY	2
37	1	"	"	159	* PBY	3
38	7	Prdn. A. E.	Table 'X'	6-22	* PAX	1
39	7	"	"	28-133	* PAX	2
40	1	"	"	159	* PAX	3
41	7	"	Table 'Z'	6-22	* PAZ	1
42	7	"	"	28-133	* PAZ	2
43	1	"	"	159	* PAZ	3
44	7	"	Table 'Y'	6-22	* PAY	1

Card No.	No. of Numeric Fields.	Contents	Product Type	Size Range (mm)	Identification Code	Sequence No.
45	7	Prdn. A.E.	Table 'Y'	28-133	* PAY	2
46	1	"	"	159	* PAY	3
47	7	Co. Stock	Table 'X'	6-22	* STX	1
48	7	"	"	28-133	* STX	2
49	1	"	"	159	* STX	3
50	7	"	Table 'Z'	6-22	* STZ	1
51	7	"	"	28-133	* STZ	2
52	1	"	"	159	* STZ	3
53	7	"	Table 'Y'	6-22	* STY	1
54	7	"	"	28-133	* STY	2
55	1	"	"	159	* STY	3

5. CHANGING THE INPUT FILE

The preceding sections have described how the PSALM main program works. It was indicated that the basic smoothing operation carried out by the program could be modified by the user. (See the example of section 2.2). The following sections describe how this can be done in practice. Section 5.1 summarises the kinds of control the user can exercise, namely those of volume of production (5.1.1) and product mix (5.1.2). Section 5.2. is concerned with describing how the controls of 5.1 may be put into effect using the PSALM File Editor. Details of the program directives and the data required are given. Section 5.3 summarises the input requirements for a complete run of the PSALM File Editor.

5.1 USER CONTROLS

The user can exert two kinds of control on the actions of the PSALM main program. Company Product Mix is controlled by the weekly production budget and the total volume of production is controlled by production adjustments.

5.1.1 Level of Production

As described in 2.2, the basic level of production depends on the smoothing function, between limits that are related to the total weekly production budget. However, once this level has been determined it may be adjusted for future weeks to any value the user desires.

If the sum of future adjustments is not zero, the user is load balancing. That is, he is planning to produce more (or less) than is required to meet the smoothing requirements.

5.1.2 Product Mix

Consideration of 2.3 will show that once the level of production is fixed, the amount of individual products made depends on the basic weekly production budget of items. In fact the absolute value of the budget for an individual product is immaterial in this scheme, only the proportion this represents of the whole budget is important.

To digress a little, the weekly production budget could be entered on the input file in percentage terms. For example, it is current practice to budget for each product in terms of the number of tons to be produced each week at a given factory. When all such budgets are added together the total is the weekly production budget. PSALM uses these values to produce allocations. However it would produce exactly the same allocations if all the production budgets were expressed as percentages of the total weekly production budget, rather than in tonnage terms.

In practice it would be usual for the product mix to be slightly different to that of the product budget by virtue of replenishment orders raised when current stock levels fall below the limits set. Since replenishment would normally be expected to account for only a small fraction of the weekly production allocation, its effect will generally be negligible but might become important in times of unusually low stock. In the latter case the problem could be avoided by lowering the minimum stock levels to a point where replenishment orders could not be generated.

5.2 THE PSALM FILE EDITOR

Each week it is necessary to preserve information for the next run of the PSALM program. This information is recorded on a magnetic tape called the input file. For the reasons described in section 2 and above, it is occasionally necessary to change the contents of the input file before running PSALM again.

The user supplies directives which enable the File Editor to insert amendments on to the file. Section 5.2.1 describes directives that enable the level of production to be changed and section 5.2.2 describes the directives that enables changes in product mix to be effected.

5.2.1 Changing the Level of Production.

When changes are made to the basic smoothed production level calculated by PSALM using the adjustment facility in 2.2, the directive used is,

```
* ADJUSTMENTS s n
```

where s = week number where adjustments are to start

n = number of adjustments

Both s and n are integer numbers (no decimal point) and must be separated from each other by at least one blank card column. The directive must be left justified to column 1 of the card.

The adjustments themselves must follow on subsequent cards, and follow the rules given in 5.2.3 on data layout.

If adjustments are not consecutive, a further directive must be used. For example, if amendments are to be made to weeks 8, 9, 10, 11, 12, 16, 17, 18, this must be done using two directives since the weeks 12 and 16 are not consecutive. This is shown below.

There is no limit to the number of separate * ADJUSTMENTS directives used.

Example

```
* ADJUSTMENTS   8  5
    2  2.3  - 2.3  - 2   - 2
* ADJUSTMENTS   16  3
   9  10  - 20.
```

Although they would nct generally be regarded as a means of control, the forecasts of total demand are used for smoothing and thus affect the level of production. If more recent and thus, presumably, more accurate forecasts come available, they must replace the previous forecasts on the input file. This is done by using the directive:-

* FORECASTS s n

In all respects its use is the same as with the * ADJUSTMENTS directive.

When the directives * ADJUSTMENTS and * FORECASTS are used, the amendments are listed by the File Editor. If a complete listing of either the amended forecasts or adjustments,from the base week to the horizon week,are required the directives are,

* LIST ADJUSTMENTS

* LIST FORECASTS

5.2.2 Changing the Product Mix

The product mix is changed using the directive:-

* BUDGETS

The directive must be left justified to column one of the card. It must be followed by cards containing a complete input of production budgets for each product at each factory. Where products are not to be made, zero must be entered. The order of input is as described under entries 3-6 of Table 4.2.1. of section 4.2.

The data layout required is identical to that used for production at the various factories (see section 4. for details). The identification characters are similar to those used for production except that the letter 'P' in those codes becomes 'B' in the present case, giving typically * BLX, * BKZ etc.

5.3 SUMMARY OF DATA REQUIREMENTS FOR PSALM FILE EDITOR.

Data following the * ADJUSTMENTS and * FORECASTS directives may be positive (or negative in the case of the adjustments), integer or decimal fractions, or a mixture of decimal fractions and integer. Each number field must be separated from its neighbour by at least one blank card column but otherwise its position on the card is immaterial. However a number must not be allowed to run from the end of one card to the start of another; all number fields must terminate on the card on which they begin.

A PSALM File Editor run is commenced by use of the directive:-

* WEEK w

where w = week number of the last run of the PSALM main program and is an integer number.

The directive is left justified to card column one.

The directives * ADJUSTMENTS and * FORECASTS may be input in any order relative to one another, but each must be followed by its related data.

All * ADJUSTMENTS, * FORECASTS, * LIST ADJUSTMENTS, and * LIST FORECASTS, where they are used, must be input before the * BUDGETS directive, if the latter is to be used.

A File Editor run is terminated by the directive:-

* END

The directive is left justified to card column one.

Table 5. 3. 1

Check List of Input to PSALM File Editor

Card No.	Contents	Parameters
1	* WEEK	w = week no. of last run of PSALM main program

2 (i)	* ADJUSTMENTSbsbn	s = week no. where adjustments start.
		n = number of adjustments
		b = at least one blank card column.

2 (ii), (iii)		
3 (i)	* FORECASTSbsbn	as for 2 (i)
3 (ii), (iii)		
4	* LIST ADJUSTMENTS	none
5	* LIST FORECASTS	none
6	* BUDGETS	none
7	* END	

Notes

(a) All directives must begin in card column 1.

(b) All parameters are integer and must be separated by at least one blank card column.

(c) The order of the directives * ADJUSTMENTS, * FORECASTS, * LIST FORECASTS and * LIST ADJUSTMENTS is immaterial save that they must follow the * WEEK directive and precede the * BUDGETS directive when present.

(d) With the exception of the * WEEK and * END directives, any or all of the remaining directives may be omitted in a given run. (Omitting all the directives would merely copy the input file which could be done more efficiently by other means).

It should be noted that the action of running the File Editor program is to produce a new input file containing the required amendments while leaving the old input file unaffected. Thus any mistakes subsequently found in the new file can be corrected by either (i) running the File Editor on this file, or (ii) reverting to the original file and creating an amended input file from this, after correcting the cause of error in the original amendment run.

As the above procedure can give rise to a number of virtually identical tapes, it is of paramount importance to keep some record of the identity of each and how they differ one from another. This can be done by keeping note of the tape serial number (T. S. N.) which is unique to a particular tape together with some indication of the contents. Serial numbers of the tapes used in a particular run of the computer are listed on the computer log, a copy of which is sent with the print-out to the user. Further information on this aspect is available from the Computer Department.

6. REFERENCE

1. R.H. Woodward, P.L. Goldsmith; "Cumulative Sum Techniques,"

ICI Monograph No. 3, Oliver & Boyd, 1964, p. 60.

ADDENDUM

The following gives details of a special form of the *FORECASTS directive described in 5. 2. 1.

Changing the "Base Week" and Horizon Period"

Section 2. 1. 1. defines the "base week" and "horizon period". The value of these variables on the input file may be changed using the PSALM File Editor. The directive used is,

```
*FORECASTS bsbn   where s = n = Ø and
                        b = at least one blank
                            card column.
```

This directive must be followed by a card containing the week number of the new base week and the number of weeks in the horizon period, in that order. These numbers must be integer (ie contain no decimal point) and must be separated by at least one blank card column. Other information pertaining to the *FORECASTS directive in Section 5 applies here.

If only one of the two variables "base week" and "horizon period" are changed, BOTH values must be re-input.

APPENDIX 1.

STOCK LEVELS USED IN

PSALM MAIN PROGRAM

Size	Minimum Levels			Maximum Levels		
	(tonnes)			(tonnes)		
(mm)	Table X	Table Z	Table Y	Table X	Table Z	Table Y
6	-	-	-	-	-	-
8	5	2	2	7	2	2
10	-	-	-	-	-	-
12	5	3	2	8	3	2
15	300	30	60	1000	60	100
18	-	-	-	-	-	-
22	300	30	50	900	60	80
28	150	15	40	350	30	60
35	100	10	10	200	20	15
42	80	10	10	160	20	15
54	60	10	10	110	20	15
76	30	10	-	50	15	-
108	30	10	-	50	15	-
133	5	3	-	8	3	-
159	5	3	-	8	3	-

APPENDIX 2.

FACTORIES OF FIRST PREFERENCE

FOR

MANUFACTURE OF CONSTRUCTION COPPER TUBES

Size	Table X	Table Z	Table Y
6	-	-	-
8	L	L	K
10	-	-	-
12	L	L	K
15	K	K	K
18	-	-	-
22	B	K	K
28	K	K	K
35	K	K	K
42	K	K	K
54	K	K	K
76	K	K	K
108	L	L	L
133	L	L	-
159	L	L	-

Note:		
L	=	Leeds
K	=	Kirkby

psalm coding sheet

1 of 3

Notes:-

(i) Details of individual products are entered in ascending order of size (ie. 6mm., 8mm., etc.)

(ii) There are 7 entries per card within a product group (eg. Table 'X') with the exception of the last which contains only one.

(iii) Individual entries are started after each major half line.

Identification Codes*

WKN	Week No.
DM-	Demand (despatches)
P--	Production
ST-	Stock
-L-	Leeds
-K-	Kirkby
-B-	Barrhead
-A-	Allen Everitt
--X	Table 'X'
--Z	Table 'Z'
--Y	Table 'Y'

*"-" indicates a missing code letter.

FULL DETAILS OF INPUT REQUIREMENTS ARE GIVEN IN THE USER MANUAL.

Example:-

0	3.8	0	0	0.1	2	4.9	*STX 2

week no:-

1	11	21	31	41	51	61	75 80
							*WKN 1
							*DMX 1
							*DMX 2
							*DMX 3
							*DMZ 1
							*DMZ 2
							*DMZ 3
							*DMY 1
							*DMY 2

psalm coding sheet

2 of 3

1	11	21	31	41	51	61		75	80
								* D M Y	3
								* P L X	1
								* P L X	2
								* P L X	3
								* P L Z	1
								* P L Z	2
								* P L Z	3
								* P L Y	1
								* P L Y	2
								* P L Y	3
								* P K X	1
								* P K X	2
								* P K X	3
								* P K Z	1
								* P K Z	2
								* P K Z	3
								* P K Y	1
								* P K Y	2
								* P K Y	3
								* P B X	1
								* P B X	2
								* P B X	3
								* P B Z	1

psalm coding sheet

3 of 3

```
1         11        21        31        41        51        61            75   80
                                                                          *PBZ 2
                                                                          *PBZ 3
                                                                          *PBY 1
                                                                          *PBY 2
                                                                          *PBY 3
                                                                          *PAX 1
                                                                          *PAX 2
                                                                          *PAX 3
                                                                          *PAZ 1
                                                                          *PAZ 2
                                                                          *PAZ 3
                                                                          *PAY 1
                                                                          *PAY 2
                                                                          *PAY 3
                                                                          *STX 1
                                                                          *STX 2
                                                                          *STX 3
                                                                          *STZ 1
                                                                          *STZ 2
                                                                          *STZ 3
                                                                          *STY 1
                                                                          *STY 2
                                                                          *STY 3
```

APPENDIX M2

PSALM

(Production Smoothing and ALlocation Model)

Technical Manual

Author: D. M. S. Peace.
April, 1972.

CONTENTS

1. INTRODUCTION

This manual gives technical information about the PSALM package. Its purpose is to enable a competent FORTRAN programmerto make adjustments as requirements change with time, and to correct errors should they appear. (Descriptions of procedures carried out by the PSALM Main Program and PSALM File Editor (User Version) are given in the PSALM User Manual). Since a policy of continuous development has been adopted, the considerable effort involved in producing flowcharts could not be justified and instead the programs are extensively annotated using comment cards.

Sections 2-4 give technical information relating to the three major programs in the package, namely,

(i) SLM1 - PSALM Main Program

(ii) SLM2 - PSALM File Editor.

(iii) SLM3 - PSALM File Editor (User Version).

Common to all the above programs is a tape file called SLM1DATAFILE. The file organisation is described in Section 5 and the file contents and their locations are listed in Appendix 2.

A number of subroutines are used by some or all of the three major programs and are concerned with activities such as file handling and input/output. Specifications for these "Service Routines" are given in Section 6.

Some of the output from SLM1 is graphical in form. The graphs and associated data listings are produced by a variant of the GOODS program, which is described in the GOODS manual.[1] Details of this in-program version of GOODS are given in section 7.

2. SLM1: PSALM MAIN PROGRAM

2.1 Hardware Requirements.

The hardware configuration and the way it is used is outlined below:-

(a) 21,760 words of core storage (1900 series processor). The program is overlayed as described in section (2.2). The binary dump of the program occupies 254 one block buckets on E.D.S. file PROGRAM FOR8. (C.S.N. 000403).

A source version is stored on cards and on file PROGRAM FOR4 in subfiles SLM1 and GOODS OVERLAY. These subfiles were created using the I.C.L. file editor XMED. To re-compile, the program description must be supplied on cards and the above subfiles then accessed by use of READ FROM statements.

(b) 2 tape decks - the input file is created on the previous run, During a given run, a scratch tape is picked up, re-named SLM1DATAFILE and given a generation number equal to that of the current week. Information to be retained is copied to the new file which thus replaces the current input file. The latter is retained for security purposes. Each tape is given a retention period of 15 days and may be released after that time.

(c) 1 E. D. S. - This store (C. S. N. 000403) holds the overlay program and provides the scratch area for an unformatted backing file.

(d) 1 card reader - used for data input.

(e) 1 line printer

(f) 1 Console typewriter - used for error messages.

2.2 Overlay Organisation.

Table 2.2 summarises the overlay arrangements. The greatest depths of overlay is 6, allowing for calls back to the permanent unit from the longest sequence of calls.

Table 2.2 Overlay Organisation.

Area	Unit	Subroutine Name	Subroutine Size (words)	Unit Size (words)	Area Size (words)
1	1	STOCK SUMMARY	1602	1602	1602
1	2	STOCK PROJECTION	678	1543	
		PRODUCTION ALLOCATION	865		
1	3	FORECAST MONITOR	522	1491	
		CUSUM	327		
		PRODUCTION MONITOR	471		
		DEMAND STATISTICS	171		
2	1	GOODS1	975	975	1225
2	2	GOODS2	1208	1208	
2	3	GOODS3	1225	1225	
2	4	INPUT	223	330	
		SEQ CHECK	107		
2	5	FREAD	89	295	
		FWRITE	56		
		FCOPY	150		
2	6	CONCARD	192	192	
3	1	SCALE	660	660	660
3	2	SELFORMAT	186	317	
		DYNFORMAT	131		
3	3	EXPONENT	285	285	
4	1	IVRTRANK	113	113	210
4	2	CHANJSCL	210	210	

2.3 Service Subroutines Called

INPUT
FREAD
FWRITE
COPY

Details of Service Routines are given in Section 6.

2.4 Error Checks.

These are described in Section 4 of the PSALM User Manual.

An error trap of the type described in XFAT manual, Chapter 7 is used. The subroutine used is CARD CHECK. This checks for EXECUTION ERROR 0 which occurs if non-numeric characters are found in a numeric field. If the erroneous character is an asterisk '*' the program assumes that a numeric entry has been missed off a card and an error message is output to the line printer to this effect. This assumption can be made because free format is used for input which causes the identification characters to be scanned if the correct number of numeric fields is not present. The first identification character in every case is an asterisk. If any other non-numeric character is found a message is printed indicating the character found and the identifier of the card containing it.

Any other error will cause the standard trace to be output.

The program is compiled in Trace 2 mode.

3. SLM2: PSALM FILE EDITOR

This program can be used to amend any item of data on the input file. It is not described in the PSALM User Manual as its use requires some experience of computer systems and in particular detailed knowledge of the file organisation used in the program. Details of the latter are given in section 5. This section describes the directives used to control the actions of the program. Hardware requirements are listed, service routines used are given and the error checks made are itemised.

3.1 Directives for File Editor (SLM2)

Including the Terminator (****) there are four directives for this program. All directives begin in column 1 of the card on which they are contained.

3.1.1 *WEEK - This directive is always the first of the run. Its form is,

*WEEKbn where 'b' indicates at least one blank card column and 'n' is the file generation number (week no) of the tape to be amended (integer).

3.1.2 *COPY - This directive causes the service routine FCOPY to be called - see section (6.3)-its form is,

*COPYbn where 'b' indicates at least one blank card column and 'n' is the record no. of the record to be copied from the input file to the output file (integer).

3.1.3 *UPDATE - This directive calls the service routine UPDATE, which may be entered in a variety of modes as described in (6.4). Some of the arguments of UPDATE are supplied as parameters on the directive card, the remainder are inserted by the program. The directive has the form,

*UPDATEbrbabmbxbf

where b = at least one blank card column (integer).

r = record number to be accessed. (integer).

a = amendment type (integer)

m = mode of entry. (integer)

x = constant amount to be added (real)

f = format requirement. (integer)

The significanceof the parameters r,a,m,x and f is explained in (6.4), where the data requirements for UPDATE are also given.

3.1.4 Run Terminator - A run of SLM2 is terminated by the directive,

****

which is the last card of the input.

3.2 Hardware Requirements.

The following hardware configuration is used,

(a) 8,704 words of core storage (1900 series processor).

(b) 2 tape decks - the input tape SLM1DATAFILE will generally have been produced by an earlier run of SLM1. It may however have been produced by runs of SLM3 or SLM2. The amended version is output with the same name and file generation number as the input file. SLM2 can be used to create a new tape when, of course, there will be no input file and all the data will be input from cards.

(c) 1 E. D. S. - this store (C. S. N. 000403) holds the binary version of SLM2 on file PROGRAM FOR8. A source version is stored on cards and on file PROGRAM FOR4 subfile SLM2 which was created using the I. C. L. file editor XMED.

(d) 1 card reader - used for input of directives and data.

(e) 1 Line printer.

(f) 1 console typewriter - used for error messages.

3.3 Service Subroutines Called

UPDATE
FCOPY

3.4 Error Checks

An error trap of the type described in XFAT manual, Chapter 7, is used. The subroutine used is PARAMETER ERROR. Its operation is best understood by considering the manner in which directives are input and identified.

When a directive is expected by the program the whole of the card is read into array TEMP in 'A' format. TEMP (1) is then examined to identify the directive. However TEMP has been assigned a channel number using DEFBUF, which means that once the directive type has been identified an appropriate format can be selected to re-read the contents of TEMP, skipping the alphameric field using an X descriptor and inputting the appropriate parameters using free format. If a parameter is missing, the free format causes TEMP to be read yet again and at this point the non-numeric '*' of the directive is read, so generating EXECUTION ERROR 0. The error trap passes control to subroutine PARAMETER ERROR which checks the execution error number. If the number is zero the non-numeric character is examined and, if found to be an asterisk, it is assumed that one or more parameters are missing from the directive card. An appropriate error message is printed on the lineprinter, and the program halts displaying the message,

ABANDON RUN - PARAMETER CARD ERROR

on the console log.

If the non-numeric character is not an asterisk an error message is again output to the lineprinter indicating the character found. The program again terminates with the above console log message.

Execution error numbers in excess of zero cause standard FORTRAN trace diagnostics to be output, which requires the program to be compiled in Trace 2 mode.

4. SLM3: PSALM FILE EDITOR (USER VERSION)

Details of the method of using this version of the File Editor are given in section 5 of the PSALM User Manual. This section outlines the hardware requirements of SLM3, the service routines used and error diagnostics given.

4.1 Hardware Requirements.

(a) 10,432 words core storage (1900 series processor)

(b) 2 tape decks - the remarks of 3.2 note (b) apply here with the exception that tapes cannot be created.

(c) 1 E.D.S - this store (C.S.N. 000403) holds the binary version of SLM3 on file PROGRAM FOR8. A source version is stored on cards and on file PROGRAM FOR4 subfile SLM3 which was created using I.C.L File Editor XMED.

(d) 1 card reader - used to input directives and data.

(e) 1 lineprinter.

(f) 1 console typewriter - used for error messages.

4.2 Service Subroutines Called.

INPUT
FREAD
FWRITE
FCOPY

4.3 Error Checks.

Two error trap routines are used in SLM3.

That described in (3.4) is used to check directives and additionally directives are checked for their context. Thus a valid directive, for example *FORECASTS, could not appear after the *BUDGETS directive since it would be out of context (see PSALM User Manual, Section 5.3).

When the * BUDGETS directive is identified, the checking procedure described in section (4.3) of the PSALM User Manual is instated.

5. FILE ORGANISATION.

Each run of SLM1 produces a tape with name SLM1DATAFILE and generation number equal to the week number of the data in the run. The file is unformatted with 1024 word block size. The FORTRAN system requires 4 words for file organisation leaving 1020 words per record available for the PSALM system. The principle adopted for file organisation and the lay-out of data with a record is described in section (5.1). Section (5.2) describes how COMMON blocks and the use of EQUIVALENCE in conjunction with the input file make information available to the program. A library of the contents and locations of data on the input file is given in Appendix 2.

5.1 File and Record Structure

5.1.1. Principle of the File Structure

As described in the User Manual, SLM1 is comprised of six major subroutines. Each of these requires information which may be classified into two groups. Thus some information is required by all or nearly all the major subroutines and some is required by only one such subroutine. This was recognised at the time when the file organisation was being specified and accordingly it was decided that each major subroutine would be assigned one or more records, and that each record would be split into "permanent" information and "temporary" information. Permanent information would be required by most or all the major subroutines called subsequently, and thus would remain permanently in core once it had been read from the input tape. On the other hand, existing temporary information would be overwritten by that of the next major subroutine called. In the event it has been found that all the permanent information is required in the first major subroutine. Thus input to subsequent major subroutines is all of the temporary type.

5.1.2 The Record Structure

The firstfour words in a PSALM record are the variables,

NWORDS, NTEMP, NPERM, NRECD,

and these are followed by the array,

IBUFF (1016)

which together fill the 1020 words available in the record. IBUFF contains all the information in the record, although generally speaking only (NWORDS-4) of 1016 words available are used. Thus NWORDS is the total number of words used in the record including itself and the other three variables above. NPERM is the number of words of permanent information held on the file and NTEMP is the number of words of temporary information. (The last NPERM words in IBUFF are permanent information).

NRECD is the number of the record which currently is a multiple of ten to enable extra records to be inserted if necessary without disrupting the system. Thus record No. 41 could be inserted between record Nos 40 and 50 without affecting the service routines that access the file.

5.2 Input/Output of File Information.

Records are read from the input tape into COMMON block /BUFFER/ which has the follow structure,

COMMON/BUFFER/NWORDS, NTEMP, NPERM, NRECD, IBUFF (1016).

If NPERM is greater than zero, the permanent information in IBUFF is copied to a second COMMON area/PERM/ which is available to all major subroutines. Copying is done using the I. C. L. routine FMOVE after using EQUIVALENCE to share the area of core used by the permanent information in/PERM/ with a dummy array IPERM. This is done in subroutine STOCK SUMMARY. In this routine a second record is required to complete the transfer of permanent information and to effect the transfer of the temporary information. EQUIVALENCE statements are used to locate the information in IBUFF with the variables in the program.

It is essential to copy the temporary information to a new file on leaving a major subroutine because reading the next record overwrites any existing information. However a record may contain permanent information which is preserved in/PERM/ and this may be changed after the record has been copied to the new file, which is necessary to preserve the temporary part of it. Notably this occurs with the production allocations (PMAKE) of subroutine PRODUCTION ALLOCATION which have the same location in COMMON block/PERM/ as RQPROD the requested production in subroutine STOCK SUMMARY. which is the first major subroutine called in SLM1. This enables the allocations of one week to be considered as the production requested in the following week's run. In order to resolve the problem of records where the permanent contents may be /changed after the record has been copied to preserve the temporary part, such records are copied to a scratch backing file on disc. When all changes have been made to permanent variables the records are read back from disc file, the appropriate permanent values inserted and the final record is output to magnetic tape to create an updated input file for the next run of SLM1.

Finally, the COMMON block/PERM/ is larger than required to hold permanent information input from file. This is because information is generated in the program that has to be available to later subroutines. Variables input from file are in numbered continuation lines of the program, and variables generated in the program are on continuation lines marked with the letters A, B, C etc.

6. SERVICE SUBROUTINES

Programs SLM1, SLM2 and SLM3 use a number of service subroutines for data input and file handling purposes. This section describes these service routines, their function, arguments and error messages, if any. Where other subroutines are called, these are listed. The subroutines are regarded as "free-standing" for this purpose. In fact this standpoint is quite valid, for example INPUT could be incorporated into any program without change, with the single reservation that COMMON areas would have to be compatible.

6.1 FREAD

Calling statement: CALL FREAD (p, r)

where p = input peripheral channel no. (integer)

r = record no. (integer)

Function: To read PSALM record number 'r' from peripheral channel number 'p'.

Subroutines called: none

Common areas: /BUFFER/

Error displays: Console display DISP:-FD. An attempt has been made to access a record number less than that currently in/BUFFER/. (The input file cannot be backspaced.)

6.2 FWRITE

Calling Statement: CALL FWRITE (p)

where: p = output peripheral channel no. (integer)

Function: To write the contents of COMMON area /BUFFER/ to peripheral channel number 'p'.

Subroutines called: none

Common areas: /BUFFER/

Error displays: none.

6.3 FCOPY

Calling statements: CALL FCOPY (p_1, p_2, r,e)

where p_1 = input peripheral channel no. (Integer)

p_2 = output peripheral channel no. (Integer)

r = record no to be copied (Integer)

e = endfile marker. (Integer)

Function: To copy PSALM record 'r' from input peripheral 'p', to output peripheral 'p_2'. Optionally the output file may be closed by setting e=1 (gives ENDFILE p_2), otherwise e=0.

Subroutines called: none

Common areas: /BUFFER/

Error displays: Console display DISP:-FY

An attempt has been made to access a record number less than that currently in /BUFFER/. (The input file cannot be back-spaced).

6.4 UPDATE

Calling statement: CALL UPDATE (p_1, p_2, p_3, r,a,m,x,f)

where p_1 = file input channel no. (Integer)

p_2 = file output channel no. (Integer)

p_3 = amendment input channel no. (integer)

r = record no. to be assessed. (integer)

a = amendment type (integer)

m = mode of entry (integer)

x = constant amount to be added (real)

f = format requirement. (integer).

Function:

To optionally input (from peripheral channel number p_1) or create PSALM record number 'r' in/BUFFER/. Amendments are input on peripheral channel number p_3 and optionally the amended record 'r' may be output to peripheral channel number p_2. The form the amendments take is described in section (6.4.1).

Arguments a, m, x and f control the above options in the following way:-

a = 0 Replace the current contents of record 'r' with the amendments

= 1 Add the amendments to the current contents of record 'r'.

= 2 Add the constant amount 'x' to the current contents of record 'r'.

m = 0 Read record 'r' from channel 'p_1' and write the amended record to channel 'p_2'.

= 1 Make amendments to the current contents of /BUFFER/ and write the amended record to channel 'p_2'.

= 2 Input record 'r' from channel 'p_1' and leave the amended record in /BUFFER/.

= 3 Make amendments to the record currently in /BUFFER/ and leave the amended record there.

x = The constant amount to be added to the contents of/BUFFER/ at the location specified in the data (see section 6.4.1). This argument is real but will be ignored if a $\neq$ 2.

f = 0 Numeric data to be input as described in section (6.4.1).

f = 1 Data is to be input according to the format supplied in the data - see section (6.4.1).

Subroutines called: I. C. L. routine COPY8

Common areas:/BUFFER/

Error Displays: Lineprinter display:-

Error - A record has been requested in the above directive having a number less than that last accessed. Records can only be accessed in ascending order of record number.

Console display:-

ABANDON RUN - PARAMETER CARD ERROR

6.4.1 Data Requirements for UPDATE

Each *UPDATE directive card is followed by cards containing information concerning (a) where the amendments are to be made and (b) the amendments themselves.

Definition.

An amendment to a PSALM record is defined as a change made to data stored at consecutive addresses in the record. If the data to be amended are not consecutive in store, a separate amendment has to be made.

The number of amendments as defined above associated with a given directive is punched (I0) in the card following the directive.

Each series of amendments is preceded by an amendment definition card punched,

nb_1sb_2REAL or,) punching should commence in
) card column 1.
nb_1sb_2INT   )

where n = number of entries in the amendment

s = start address of amendment

b_1 = at least one blank card column.

b_2 = ONE blank card column

If, as a result of changing the organisation of a record or creating a new record, the variables NWORDS, NTEMP, NPERM and NRECD are to be changed, or inserted for the first time, the amendment definition card must be punched with n=s=0. This must be followed by the four (even if some have not changed) new values punched on one card (4I0). A new directive must then follow.

If argument 'a' is set to 2 in the call to UPDATE, the constant amount 'x' (defined in the argument definitions) will be added to the locations specified in the amendment definition card and no other data will be required.

If alphameric information is to be input, argument 'f' of the *UPDATE directive must be set f=1, and a format specification input on a card to describe the layout of the alphameric information. This format card must follow the amendment definition card and must be present in each amendment associated with an *UPDATE directive having argument f=1.

The above is summarised in Table 6.4.1.

Table 6.4.1 Check List-Assembly of data for Subroutine UPDATE

Card No.	Contents.
1	No. of amendments (d) - format (I0)
2	Amendment Definition Card - nb_1sb_2REAL or nb_1sb_2INT where n = no. of entries in amendment. b_1 = at least one blank card column s = start address of amendment b_2 = ONE blank card column.
2a	present only when argument 'f' of subroutine UPDATE is set f=1.

Table 6.4.1 Cont'd

Card No.	Contents.
	Format is punched on one card (card columns 1-40) as described in I. C. L. FORTRAN MANUAL.
3 etc	Amendment entries in (508F0. 0) format if amendment definition is, $nb_1 sb_2$REAL and in (101610) format if amendment definition is, $nb_1 sb_2$INT If a format specification has been input (f=1) then the amendments will be read in the manner specified. Sequence 2-2a-3 is present 'd' times in a given call to UPDATE.

6.5 INPUT

Calling statement: CALL INPUT (u, z, k, p_1, p_2, p_3)

where u = dummy array (real)

z = size of dummy array (integer)

k = four character identification code in which the first character is always an asterisk '*'.

p_1 = card reader input channel no. (integer)

p_2 = channel defined /ARRAY for core transfers (Integer).

p_3 = Line printer channel no. (integer)

Function: To read data punched on cards (or possibly from other peripherals) and check that they are in the correct sequence and have the correct number of entries per card. Currently seven entries per card are expected on all except possibly the last, depending on the size of 'z'.

Further information on the checking procedures are given in the PSALM User Manual.

Subroutines Called: SEQ CHECK

Common areas: /ERROR/

Error displays: See section (6.6) SEQ CHECK.

6.6 SEQ CHECK

Calling statement: CALL SEQ CHECK (i, q, k, t, p_3)

where i = identification characters read from card in INPUT.

q = sequence number read from card in INPUT.

k = identification characters expected by program.

t = sequence count i. e. sequence number expected by program.

p_3 = line printer channel no.

Function:

To compare the identification codes of a given deck of cards against a pre-determined sequence contained in the calling program. Error messages are given when a match is not obtained.

Subroutines called:

PAGE This gives a page throw and page header.

CARD CHECK - Error Trap.

Common areas:/ERROR/

Error displays: Line printer:- ERROR - Program expects the identification characters '------' but has found '-----'. The data cards are out of sequence or have been given the wrong sequence numbers.

6.7 CARD CHECK

Calling Statement: Called by FORTRAN Trace 2 error routines.

Function: To report two types of error,

(i) that too few entries are present on a card. This is deduced when the asterisk at the start of the identification characters is read.

(ii) that non-numeric characters have been found in card columns 1-74. If an asterisk has been punched in these columns, the program will wrongly report an error of type (i).

Subroutines called: none

Common areas:/ERROR/

Error displays: Lineprinter: A non-numeric character other than a decimal point has been found in columns 1-74 of the card identified as '------'.

OR,

The card identified as '........' appears to contain too few entries.

7. GOODS: In Program Version

The graph plotting routines used by PSALM are essentially those of the "free-standing" GOODS program and are described in the GOODS Manual.[1] As the latter explains, to utilise some of the options available with GOODS requires card input in addition to the data series co-ordinates to be plotted (which were also usually on cards). In the "in program" version, it would be unacceptable to have to input control cards to produce the graphical output, and thus certain modifications have been made. However these modifications have been designed to minimise the differences between the "free standing" and "in program" versions of GOODS, and thus the main reference remains the GOODS Manual. This section describes differences between the two versions and explains how the GOODS facilities may be obtained by the calling program.

7.1 Information Transfers.

Information from the calling program is copied into COMMON areas associated with the "in-program" version of GOODS, which is called GOODS OVERLAY. There are 3 COMMON areas,

/GRAPH/, /CONTROL/and/EXTRA/.

/GRAPH/ is essentially the same common block as is used in GOODS, /CONTROL/ is used in conjunction with subroutine CONCARD as described below and/EXTRA/ is used to permit the three main subroutines of GOODS OVERLAY, (GOODS1, GOODS2 and GOODS3) to be overlayed.

A run of GOODS is initiated by a control card. The latter is simulated in GOODS OVERLAY by subroutine CONCARD whose arguments are the same in number and type as the contents of a GOODS control card with the exception of LORD and LABS which must be entered as TEXT variables (e.g. 1HI, 1H-). The arguments are then copied by CONCARD into COMMON block/CONTROL/.

The data series to be plotted are copied into array DATAR. DATAR (1,J,I) contains abscissa values and DATAR (2,J,I) contains ordinate values. J is the number of the data series (maximum value of J is 5) and I is the co-ordinate number, stored in increasing order of abscissae.

Variables that would normally be input on cards such as NOHORMK, NOVRTMK etc must be assigned their values in the calling program, and these are available to GOODS OVERLAY through COMMON.

When more than one call to GOODS OVERLAY is made in the course of a program, it is necessary to zeroise the above COMMON blocks before assigning new values. This prevents values from a previous call corrupting the new call. It is conveniently done by using EQUIVALENCEd arrays TEMP1, TEMP2 and TEMP3 in conjunction with FMOVE.

The IFMAT facility (ADDENDUM 2.14-GOODS Manual). has been enhanced so that if floating point listing is not specified (IFMAT=0) the listing will be dynamically formatted to accommodate any size of number printed to one decimal place.

There is no title facility in GOODS OVERLAY, instead axis descriptors must be copied from the calling program into arrays 'X AXIS' and 'Y AXIS'. Similarly the data descriptors used in data listing and the 'Symbol Key' must be copied into array DES (I, J) where I is the series number and J takes values 1 to 5 so enabling 40 characters to be read.

8. REFERENCE

1. PEACE D.M.S; 'GOODS: Graphical Output of Data Series'

Nov 1970.

APPENDIX 1

CALCULATION OF ADDRESSES

Most of the information relating to the products Table X, Table Z and Table Y are stored in arrays dimensioned (16,3). The program was originally written for imperial size products where there were 16 products in each product range. Now that the program is concerned only with the metric range, only the first 15 storage locations are used in each case. Thus Table X data starts at (1,1) Table Z at (1,2) and Table Y at (1,3) in a given array. This order of product groups (X, Z, Y) is strictly observed throughout.

When the products are related to particular factories as, for example, in the weekly production figures, the array is dimensioned (16,3,4) where,

(1,1,1) is the start of Leeds data.

(1,1,2) is the start of Kirkby data.

(1,1,3) is the start of B'head data.

(1,1,4) is the start of Allen Everitt data.

Within a given factory group the products maintain the order above (i.e. X, Z, Y).

Appendix 2 lists the start addresses of arrays in each of the records on the input file. However when data pertaining to a particular product, say 54mm Table Z, is to be accessed, a certain amount of calculation is required to obtain the address. To minimise this problem Table 1 lists constants to be added to the start address of a given array to obtain the address of a particular product.

Table 1.

Location Constants for Use with Start Addresses in Accessing Product Data.

No.	Product Size (mm)	Table 'X'	Product Type Table 'Z' Location Constants	Table 'Y'
1	6	0	32	64
2	8	2	34	66
3	10	4	36	68
4	12	6	38	70
5	15	8	40	72
6	18	10	42	74
7	22	12	44	76
8	28	14	46	78
9	35	16	48	80
10	42	18	50	82
11	54	20	52	84
12	76	22	54	86
13	108	24	56	88
14	133	26	58	90
15	159	28	60	92
16	Spare	30	62	94

The above constants should be added to the start addresses given in Appendix 2 to give a product address.

Notes:-

(i) The above apply only for REAL arrays. They must be divided by two for integer arrays.

(ii) When products are defined by factory, add the following amounts to the above table of constants Leeds 0; Kirkby 96; Barrhead 192; Allen Everitt 288. (For integer arrays these should be divided by two).

(iii) The arrays used to store forecasts etc for output in graphical form have unit dimension of (104). An address 'a' is calculated in two stages

(a) calculate the position 'p' in the stored series using the expression,

p = w-b+1 where w=week no. to be accessed

b=base week

(b) calculate the absolute address 'a' using the expression,

a = 2(p-1) + s where s=start address.

This procedure is not necessary when using SLM3, but is required for SLM2.

Example.

To access product 54mm Table 'Z' in a real array with start address 205, add the constant 52 from Table 1 to give the address 257. If the product were made at Kirkby its address would be 257+96=353.

APPENDIX 2

INPUT FILE CONTENTS

Record called by subroutine: STOCK SUMMARY Record No:10

Array	Dimension	Type	Size (words)	IBUFF Start Address	End Address
ICODE	(16,3)	I	48	1	48
PTYPE	(3)	R	6	49	54
IPNAME	(5,16,3)	I	240	55	294
IFACT	(4)	I	4	295	298
ISITE	(3,4)	I	12	299	310
CSTOC	(16,3)	R	96	311	406
MAXLVL	(16,3)	I	48	407	454
RQPROD	(16,3,4)	R	384	455	838
DEM SUM	(16,3)	R	96	839	934

NWORDS = 938; NTEMP=0; NPERM=934; NRECD=10.

Note:- IPERM is dimensioned (1032) and is equivalenced to ICODE(1) to enable FMOVE to be used to copy NPERM words from /BUFFER/. (IPERM is built up from record 10 and 20).

Record called by subroutine: STOCK SUMMARY Record No.: 20.

Array	Dimension	Type	Size (words)	IBUFF Start Address	End Address
MINVL	(16,3)	I	48	1	48
DEM SQ	(16,3)	R	96	49	144
WKBUDG	-	R	2	145	146

NWORDS= 150; NTEMP=48; NPERM=98; NWORDS = 150.

Record called by subroutine· STOCK PROJECTION Record No.30

Array	Dimension	Type	Size (words)	IBUFF Start Address	End Address.
FHOR	(104)	R	208	1	208
SHOR	(104)	R	208	209	416
PHOR	(104)	R	208	417	624
PADJ	(104)	R	208	625	832
BUFF STOCK	-	R	2	833	834
IBASE WK	-	I	1	835	-
IHOR WK	-	I	1	836	-

NWORDS = 840; NTEMP=836; NPERM=0; NRECD=30.

Record called by subroutine: PRODUCTION ALLOCATION Record No40.

Array	Dimension	Type	Size (words)	IBUFF Start Address	End Address
PBUDG	(16,3,4)	R	384	1	384

NWORDS = 388; NTEMP=384; NPERM=0; NRECD = 40.

Record called by subroutine: FORECAST MONITOR Record No:50

Array	Dimension	Type	Size	IBUFF	
			(words)	Start Address	End Address
PFCST	(16,3)	R	96	1	96
URV	(")	R	96	97	192
ALRV	(")	R	96	193	288
UDI	(")	R	96	289	384
ALDI	(")	R	96	385	480
ISTATUS	(2,16,3)	I	96	481	576
USCORE	(16,3)	R	96	577	672
ALSCORE	(")	R	96	673	768
TURV	-	R	2	769	770
TALRV	-	R	2	771	772
TUDI	-	R	2	773	774
TALDI	- -	R	2	775	776
ITSTATUS	(2)	I	2	777	778
TUSCORE	-	R	2	779	780
TLSCORE	-	R	2	781	782
CUM FCST	(16,3)	R	96	783	878
CUM DIFF	(")	R	96	879	974
TCUM FCST	-	R	2	975	976
TCUM DIFF	-	R	2	977	978
TCUM DEM	-	R	2	979	980

NWORDS = 984; NTEMP=980; NPERM=0; NRECD=50.

Record called by subroutine: STOCK MONITOR Record No: 60

Array	Dimension	Type	Size (words)	IBUFF Start Address	IBUFF End Address
PDEV	(13,4)	R	104	1	104
BDEV	(13.4)	R	104	105	208
PBUDGET	(4)	R	8	209	216
CURRENT PREQ	(4)	R	8	217	220

NWORDS= 228; NTEMP=224; NPERM=0; NRECD=60.

APPENDIX M3

CONTENTS

1. GENERAL DESCRIPTION.

2. FACILITIES AVAILABLE.

3. DATA ASSEMBLY

1. GENERAL DESCRIPTION

1.1 INTRODUCTION.

The graphical presentation of data is a valuable and widely used method of recording and understanding information. Unfortunately the preparation of graphs is tedious and time consuming, which can be a strong disincentive to avoid using this potentially revealing approach. GOODS is a program designed to minimise the time required to produce graphs which have most of the features of those manually prepared. Of course some computer installations have "GRAPH-PLOTTER" programs which can produce graphs of excellent quality when used in conjunction with off-line graphplotting equipment. However in many cases GOODS has the following advantages over GRAPHPLOTTER routines.

An obvious advantage of GOODS is that it provides a computer graph plotting facility at installations where the GRAPHPLOTTER hardware is not available. Unlike the GRAPHPLOTTER routines, GOODS requires no knowledge of a programming language, although its scope may be increased if statements in a standard format are inserted into the source program. GOODS will readily accept as input, output from other programs punched, say, on paper tape. Finally output is printed on the line-printer which, being an on-line activity, produces the results more quickly than with the off-line graphplotting equipment.

However the GRAPHPLOTTER does give superior accuracy since it is able to plot continuous lines, whereas GOODS only plots points at discrete intervals. The GRAPHPLOTTER therefore is usually necessary for computer preparation of calibration and other forms of reference curve. For many other forms of data, the "reliability" of the points to be plotted does not justify the accuracy available with the GRAPHPLOTTER and GOODS may be preferred.

1.2 MODES OF USE.

For the purposes of explanation, three modes of use may be described, although in practice it is expected that (1.2.1) will be the one most commonly used.

1.2.1 Co-ordinate Mode.

In this mode up to five curves may be plotted on the same axes using different symbols. In addition up to 15 vertical and 15 horizontal lines may be drawn on the graph in positions specified by the user. The purpose of these lines, if required, is either to divide the graph into separate areas or to provide a grid to facilitate reading the scale at positions remote from the axes.

Cont'd.......

The scales on the axes may be determined in two ways.

(i) If the co-ordinates (x_1, y_1), (x_2, y_2), ---(x_n, y_n) are input without other instructions, the program will select a scale consistent with the minimum and maximum values of x and y.

(ii) The user may input the maximum and minimum values of x and y to be encountered and so "fix" the scale. The values input may be dummies, that is they need not correspond to the actual maxima and minima. This permits a series of graphs with differing ranges to be drawn on comparable scales.

Title information is presented on cards which is reproduced on the line-printer together with a printout of the values of all the co-ordinates plotted. This output may be suppressed if not desired.

The characters used for plotting co-ordinates are provided by the program, but the user's set may be used alternatively.

Finally more than one graph can be constructed in a given run.

1.2.2. Function Mode.

Once again up to five functions may be plotted. However all must be functions of a common variable (x) which is input in the form (x_1, O), $x_2, O)$, --- (x_n, O).

1.2.3. Combined Mode.

Up to ten curves may be plotted on the same axes, although this is not recommended because of the loss of clarity. These curves may be comprised of up to five series of co-ordinates and up to five functions, the latter based on a series x_1, x_2, ---- x_n. This series may be obtained from one of the co-ordinate series of the form (x_1, y_1), (x_2, y_2), ---- (x_n, y_n) or may be input as an additional series of the type described in (1.2.2).

1.3 INPUT DATA ORGANISATION AND USE.

The following terms are introduced to aid the exposition.

Definitions.

(1) Data Series - previously called a co-ordinate series it takes the form (x_1, y_1), (x_2, y_2), ---(x_n, y_n) and represents the "n" points to be plotted to form one curve.

Cont'd.......

Note - the values x_1, x_2, --- x_n must be in ascending order of magnitude.

(ii) Data Set - this is the collection of all data series required to plot ONE graph, together with any title information that may be required.

(iii) Data Block - this is the collection of all data sets required for a run. Thus a data block is composed of all the data cards for a given run.

At run time all the data series are punched on cards together with title information. (The user merely writes this information on coding sheets in the order it is required. The cards punched from these sheets are then in the correct order to be submitted for processing i. e. they form the data block) Additionally a single control card is inserted before every data set which selects the facilities to be used in the graph. The first control card in the data block also indicates the number of graphs to be constructed (or equivalently the number of data sets present in the data block requiring processing).

2. FACILITIES AVAILABLE.

This section describes in detail the facilities available and is included mainly for reference purposes. The method of calling the facilities by using a control card is described in detail in section 3, which also gives check lists-Tables (3.1) and (3.2)-for assembling data in the correct order. However to understand this section it must be known that GOODS requires one control card for each graph plotted, to indicate the facilities to be used. Any additional date required as a result of calling the various facilities is required next, and finally the data on which the curves are based is given. The exact way this is done is not considered further in this section.

For ease of reference, code names, which in fact are program control variables, have been given to the various facilities. These are the first "word" in the titles below, and the values given to them on the control card preceding each set of data cards determines how the program will operate and what supplementary data, if any, will be required.

It is recommended that when reading the following sections for the first time, reference is made to the sample coding sheets and corresponding output at the end of the manual.

Cont'd......

2.1 (NG) NUMBER OF GRAPHS.

Up to ninety nine graphs can be constructed in one run (i.e. NG can take values 1-99)

2.2 (NO) NUMBER OF DATA SERIES IN GRAPH.

Up to five data series can be plotted on one graph. (i.e. NO can take values 1-5).

2.3 (NF) NUMBER OF FUNCTIONS IN GRAPH.

Up to five functions can be plotted on one graph. (i.e. NF can take values 1-5).

See also (2.4) and (2.5) below.

Since it is impossible to anticipate which functions might be required the user is required to supply these himself, which of course requires some programming knowledge. In most cases this should not need to be extensive as many common functions may be expressed as "statement functions" on one card. More complex functions may require three or four cards be inserted. Details of functions and statement functions may be found the I. C. L. FORTRAN Manual pp. 30 and 50.

When the functions have been inserted into the program they are referenced by GOODS using the standard names FUNC1, FUNC2, --- FUNC5. The numbers 1-5 at the end of each standard name are part of the name but also serve to indicate which function series they generate. In practice each function will be written, say, FUNC1 (j_1, j_2, --- j_n) where j_1, j_2, ---j_n represent the arguments - see FORTRAN manual.

In the computer program, preceded by the comments,

```
C
C      Calculate Min. and Max. values of functions.
C
```

are five cards (i.e. lines) punched,

```
CbbbbbFUNC (I, 1) = FUNC1 (J1, J2, ---JN) etc.
```

These standard cards are made operative by replacing them with a duplicate except that no "C" is punched in column 1. If NF functions are to be plotted then the first NF cards are made operative. The "C"s should be re-punched before a new run is commenced.

Cont'd.........

Summary

If functions are to be plotted three things must be done,

(i) NF must be set to the appropriate value - see also notes (2.4) and (2.5).

(ii) A card (or cards) must be added to the program to describe the function(s) to be calculated and plotted (See I. C. L. FORTRAN Manual).

(iii) The standard cards in the program must be made operative by removing the "C" punched in column one.

2.4 (IPDS) INPUT DATA SERIES SELECTION FOR FUNCTIONS.

All functions in GOODS share a common variable x. Thus if y = f(x) a series of x values will generate a corresponding number of y values, and these are output as (x, y) co-ordinates.

For example, a linear regression analysis might produce the relationship.

Equation (1) $y^* = m.x + c$ (ie $y^* = f(x)$)

If this function were used in GOODS, Equation (1) could be plotted by inputting a suitable series of values of 'x' in the form (x_1, 0), (x_2, 0) etc. Alternatively, in the above example, it might be desired to plot the data on which the regression analysis was based i. e. (x_1, y_1) (x_2, y_2) etc. In this case, instead of inputting a separate series on which to calculate the function values, the values of 'x' in the original data could be used for this purpose. Thus for every value of x plotted on the graph there would be two ordinate values - plotted using different symbols, see (2.7) - corresponding to the original observation (y) and the regression equation value (y^*).

IPDS is used to indicate which data series is to provide the basis for the function values. It can take values in the range 1 - (NO+1). If IPDS = NO+1 then an additional series (x_1, 0), (x_2, 0) etc. must be input for the purpose of evaluating the functions. (IPDS must not exceed the value IPDS = 5, however.)

If no functions are to be plotted (NF=0), IPDS is not used and must be set to zero.

Cont'd........

2.5 IOF: PLOTTING FUNCTIONS ONLY.

When all the series to be plotted are expressed as functions, IOF must be set to one. At all other times IOF = O.

When IOF = 1, there can only be one input data series and NO = IPDS = 1. The data series may be of the form $(x_1, 0)$, $(x_2, 0)$ etc or (x_1, y_1), (x_2, y_2) etc.

2.6 ISETSCL: USER SELECTION OF SCALE.

The user may select the scale he wishes to use by setting ISETSCL = 1. If ISETSCL = 0, the program will select an appropriate scale.

The scales available on either axis are 0-100, 0-50, 50-100, and 10n - (10n +20) where n = 0, 1 --- 8.

In addition scaling factors are calculated and are output in the form (ABSCISSA VALUES x 10 EXP n) (ORDINATE VALUES x 10EXP n) where n = exponent to which power 10 is to be raised.

If ISETSCL = 1 the user is additionally required to input minimum and maximum values of'x' and 'y' in order that the program may select the scales. These values need not correspond to the actual maxima and minima, and in this way a number of graphs may be constructed on comparable axes, where otherwise the scales would be quite different. However the values input must span the actual range of values or the program will fail.

2.7 ISELPT: USER SELECTION OF CHARACTERS FOR CO-ORDINATE PLOTS.

The program uses the following characters in plotting co-ordinates of the various data series and functions.

Table 1.

Data Series Number.	Character.	Function Number.	Character.
1	*	1	A
2	+	2	B
3	.	3	C
4	#	4	D
5	X	5	E

Cont'd.......

If ISELPT is set equal to 1 the user may input his own characters. It should be noted that a full set of symbols (10 characters) need not be submitted, merely sufficient for the number of curves to be plotted.

2.8 LORD AND LABS: CONSTRUCTION OF HORIZONTAL AND VERTICAL LINES

A maximum of fifteen horizontal and a similar number of vertical lines can be drawn using any character in the 1900 series set except zero. This facility is selected by setting LORD (or LABS) equal to the desired character. Additionally the user must indicate the number of lines to be drawn and their positions. These latter values are connected by the program to the scale of the graph. They must also be punched in ascending order of magnitude and fall within the range of the data. When negative data is present, LABS="-" and LORD = "I". This is automatic and overrides the user's input when this is present.

2.9 ISUPRZ:SUPPRESSION OF DATA LISTING.

Before plotting a graph the program will normally (ISUPRZ=O) print out the data series input and any function series that have been calculated. Title information previously input by the user describing these series collectively is also output.

If ISUPRZ=1, this output will be suppressed although title information relating to the data series and functions is still needed in the data - see (2.12).

2.10 LOGX, LOGY: LOGARITHM (base 10) TRANSFORMATION OF INPUT DATA.

The logarithms (base 10) of the input data may be plotted by setting LOGX and/or LOGY = 1. LOGX = 1 will transform the abscissae to their logarithms and LOGY = 1 will convert the ordinates, otherwise LOGX = LOGY = 0 *

2.11 LNX, LNY: LOGARITHM (base "e") TRANSFORMATION OF INPUT DATA.

As (2.10) but transformation is to natural logarithms (base"e"). LNX = 1, gives transformation of abscissae and LNY = 1 gives transformation of ordinates. Otherwise, LNX = LNY = 0.*

* NOTE: (i) LOGX and LNX should not be set to one at the same time. Similarly for LOGY and LNY.

(ii) If ISUPRZ = 0, the data listing will be of the transformed values and not the input values.

(iii) All input data must be positive.

Cont'd.......

2.12. TITLE INFORMATION.

2.12.1 Major Headings.

Title information is punched on cards, one card containing one title line (79 characters). Typically one card would contain title words, the next underlining, the next a blank card (strictly onlycols. 2-80 are blank) to give a line feed and so on. At run time each card is scanned and its contents printed, until the complete title has been written.

The process of scanning and printing continues until a card is scanned in which column one is either blank or punched with a zero. All preceding cards must contain "1" punched in column one as this signifies another card has to be read. If no title is to be written a single blank card will suffice. (In effect this causes one line of blanks to be printed and since the control punching in col. 1 is blank the program proceeds).

Three sets of title information can always be output, although these can be suppressed in effect using a blank card as described above, or in the case of (2) below, by setting ISUPRZ=1.

(1) A title page is output at the start of a run. The program prints the date on this sheet in addition to user's title information.

(2) A heading describing the graph is output on a new page and this is followed by a listing of the data series and functions (2.12.2.).

(3) A further title may be printed before the graph is drawn. Presumably this would normally be suppressed but it may be used to print the graph title or describe the axis variables if ISUPRZ = 1 and the title described in (2) above has been suppressed.

2.12.2 Descriptions of Data Series and Functions.

A description of each data series is input in order that on output this can precede the listing.

If functions are to be calculated then descriptions of these are also input in order that they too have a title on the output listing.

Cont'd.......

The above descriptions are limited to 40 characters and must be included even if the data series listings etc. are to be suppressed (ISUPRZ=1) since they are also used in the symbol key which is printed on the graph.

2.13. GROUPING OF CHARACTERS.

In operation the program converts input data into a scaled integer set which corresponds to the actual input values. Thus under some circumstances the co-ordinates (90.8, 36.1), (91.1, 40.3), (91.4, 44.1), would be converted to the integer set (91, 36), 91, 40), (91,44) so causing three points to be plotted against a common abscissa. In this case, in addition to the points sharing a common abscissa, a further character "M" is printed in a position corresponding to the mean of these points.*

Note:- If "M" is printed on the ordinate axis and no other points appear, in fact other points are present but have been over printed on the characters making up the axis. Since the axis is printed using asterisks, this can only occur when the plotting character is also an asterisk, and the "missing" points can be identified from the data listing.

If only a single asterisk is printed on either axis, "M" will not appear but instead the co-ordinates are printed on the same line but off the graph using the format - (x,y) = *. Normally (ISELPT=0) the asterisk is used for plotting data series No. 1 - see (2.7). The program will not output the co-ordinate values as described above when ISELPT=1, and this should be remembered when selecting alternative characters to those supplied by the program, or usihg the same characters but in a different order.

* Unpredictable behaviour will occur if grouped abscissae are used for function evaluation.

3. DATA ASSEMBLY.

This section gives details of coding formats necessary for the preparation of data on punch cards. It is recommended that when reading the following sections for the first time, reference is made to the sample coding sheets and corresponding output at the end of the manual. The first card(s) in every data block contains title information (see 2.12.1) used to construct the title sheet which precedes the data listings and graphical output. Since there is only one title sheet this information is required only once.

Each graph is produced from the contents of a data set and it will be seen - for definitions see (1.3) - that a data block comprises of one set of the title cards described above and a number of data sets. The assembly of a data set is described below.

Cont'd.......

3.1 FORMATS.

Data for goods is punched in two formats:-

(i) Right justified integer format.

(ii) Free format.

3.1.1. Right Justified Integer Format (I)

This data must have no decimal point. In cases where more columns are available than are required to represent the number, the number must be right justified. That is, the least significant digit should be in the right-most column of the space available. For example, if the integer thirty nine were to be punched in this format in columns 1-4 of a card, then the first four columns would read bb39, where 'b' indicates a blank column. In abbreviated form "columns 1-4 (I)" would indicate that a variable was to be punched in right justified integer format in columns 1-4.

3.1.2 Free format - (F)

In this format each number must be separated from the preceding number by at least one blank column, but otherwise the position of numbers on the cards is irrelevant. A decimal point is optional for integer values.

Usually 10 numbers are read per card although this is not always so as will be seen in later sections. If the total number of co-ordinates (see also 3.1.3) is notamultiple of ten the last card will contain less than ten entries. This will not cause errors.

If the numbers are exceptionally long e.g. 82356221 or 0.0000000367, ten numbers will not fit on one card. In this case suppose seven numbers could be fitted on one card without overflow, terminating, say, in column 76. The remaining three numbers could be written on the next card and then a new card started. Once again, spacing between numbers is not important providing at least one blank column separateseach number from its neighbour. Normally however all ten numbers will fit on one card. The abbreviation for free format is (F).

Cont'd.........

3.1.3. Data Input

The data to be plotted is input as (x,y) co-ordinates. Thus five co-ordinates are written on each card except possibly the last (see 3.1.2). The co-ordinates (1.0, 1.25),(2.0, 2.5) (3.0, 3.75) could be written 1.0b1. 25b2.0b 2.5b 3.0b 3.75 where 'b' represents a blank column. The spacing is immaterial but the order is essential. The 'x' values must be in ascending order of magnitude.

3.2 CARD(S) FOR TITLE SHEET.

This card(s) must be the FIRST in every data block, and should contain the informationwhich is to appear on the title sheet. There is no limit to the number of cards that can be used (see 2.12.1) but at least ONE card must always be present.

3.3 THE CONTROL CARD

This card must be first in every data set.

The facilities described in chapt. 2 are obtained by punching the appropriate columns of the control card. The possible values that can be punched are indicated in Table (3.1). Facilities are identified by the codes names given in section 2. The relevant sub-sections of section 2 are listed in the 'Description' column.

3.4 TRAILER CARDS.

The parameters punched on the control card instruct the program to use the facilities selected by the user. Some of these facilities require extra information which must be provided in the correct sequence. Where the extra information is contained on more than one card, these latter cards must be in the correct sub-sequence.

Table 3.2 contains the sequence and sub-sequence numbers. The table is entered at sequence No. 1. Depending on the value of the control variable ISUPRZ the line corresponding to sub-sequence 1(i) or 1(ii) is followed. A brief description of the contents of the trailer card is given in the column "Card Contents". The following two columns are self-explanatory and the final column indicates where the table should next be entered. In some cases the sub-sequence just completed has to be repeated, in others a transfer to the next sequence is indicated.

An example of the coding and corresponding output is attached to the report cover.

3.5 FURTHER GRAPHS.

If more than one graph is to be plotted at run time, the procedure starting at (3.3) should be repeated for each further graph to be drawn.

Cont'd.........

TABLE 3.1 CONTROL CARD CONTENTS

Card Column No.	Facility	Possible Values (Inclusive)	Description	Notes.
1,2	NG	1 - 99	No. of graphs - (2.1)	Only required on first control card. (May be blank otherwise). Format (I)- see (3.1.1)
3	NO	1 - 5	No. of data series (2.2)	
4	NF	1 - 5	No. of functions - (2.3)	
5	IPDS	(a) NF 0 1 - (NO+1) but see"Notes"column. (b) NF= 0 IPDS=0	Data series used in function evaluation (2.4)	IPDS must not exceed value IPDS=5

6	IOF	0 or 1	Only functions to be plotted - (2.5)	When IOF=1 only functions are plotted.
				The following are essential,
				NO = 1
				IPDS. = 1

7	ISETSCL	0 or 1	User selection of Scale-(2.6)	ISETSCL=1 indicates user will select scale.

TABLE 3.1 CONT'D

Card Column No.	Facility	Possible Values (Inclusive)	Description.	Notes.
8	ISELPT	0 or 1	User input of characters for plotting - (2.7)	ISELPT=1 indicates user will input his own characters.
9	LORD)	Any character in I. C. L 1900 Series printing set.	Printing of Horiz. (LORD) and Vert. (LABS)lines - (2.8)	If LORD and/or LABS punched zero it is assumed corresponding facility is not required.
10.	LABS)			
11.	ISUPRZ	0 or 1	Suppression of data listing - (2.9)	ISUPRZ = 1 indicates suppression.
12.	LOGX	0 or 1	Transformation of abscissae to logarithms (base 10) - (2.10)	LOGX=1 gives transformation.
13.	LOGY	0 or 1	Transformation of ordinates to logarithms(base 10) - (2.10)	LOGY=1 gives transformation.
14.	LNX	0 or 1	Transformation of abscissae to natural logarithms (base"e") - (2.11)	LNX=1 gives transformation
15.	LNY	0 or 1	Transformation of ordinates to natural logarithms(base"e") - (2.11)	LNY=1 gives transformation.

TABLE 3.2 CONSTRUCTION OF DATA SET

Notes:- (i) A control card must precede every data set constructed as described below.
(ii) REMEMBER the title cards for the title sheet - see (2.12.1) and (3.2)

Sequence No.	Control Variable	Sub-sequence No.	Card Contents.	Maximum no of entries per card.	Columns Available	Subsequent Action.
1	ISUPRZ=0	(i)	Graph title	See(2.12.1)	2 - 80	(a)Col. 1. Punched 1. Go to 1 (i) (b) Col. 1 blank or punched 0 Go to 2
	ISUPRZ=1	(ii)	No card required.			Go to 2
2	ISETSCL=0	(i)	No card required.			Go to 3
	ISETSCL=1	(ii)	(a) min 'x' value (b) max 'x' value (c) min 'y' value (d) max 'y' value	4 entries in order a, b, c, d. All four values are essential.	(F)-see (3.1.2)	Go to 3
3	ISELPT=0	(i)	no card required.			Go to 4
	ISELPT=1	(ii)	characters to be used for plotting-(2.7)	10 characters	1-5(data series) 6-10 (functions)	Go to 4

TABLE 3.2 CONTD.

Sequence No.	Control Variable	Sub-sequence No.	Card Contents.	Maximum no of entries per card.	Columns Available	Subsequent Action.
4.	LORD=0 (ie zero)	(i)	No card required			Go to 5
	LORD≠ 0	(ii)	number of horiz. lines to be plotted.	1	1-2(I)+-see (3.1.1)	Go to 4 (iii)
		(iii)	Position of horiz. lines.	10	(F)	Go to 5
5.	LABS=0 (ie zero)	(i)	No card required			Go to 6
	LABS≠ 0	(ii)	No. of vertical lines to be plotted.	1	1-2(I)+	Go to 5 (iii)
		(iii)	Position of vertical lines.	10	(F)	Go to 6
6.	IPDS≤NO	(i)	Data description *	40characters	1-40	Go to 6 (ii)
		(ii)	No of co-ordinates in data series	1	1-3(I)+	Go to 6 (iii)
		(iii)	Data co-ordinates	10 (see3.1.3)	(F)	Repeat starting at 6(i) a further(NO-1)times, then go to 7.

+ In this case the (I) format requirements may be relaxed and the integer number may be written anywhere on the card. Obviously numbers written in (I) format will still be acceptable.

* See overleaf.

TABLE 3.2 CONTD

Sequence No.	Control Variable	Sub-Sequence No.	Card Contents	Maximum no of entries per card	Columns Available	Subsequent Action
	IPDS=NO+1**	(iv)				As for 6(i)-6(iii) initially but repeat a further (NO) times, then go to 7.
7	NF=0	(i)	no card required.			Go to 8
	NF > 0	(ii)	Function description	40 characters	1 - 40	Repeat 7 (ii) a further (NF-1) times, then go to 8.
8	None	(i)	Graph title - see 2.12.1 (at least ONE card is required)		2 - 80	Finally, go to 9
9	NG=1	(i)	no card required			END
	NG>1	(ii)	Ditto			Repeat starting at 1 a further (NG-1) times then, END. (Remember-a control card must precede every data series)

* If IOF=1, then IPDS=NO=1 and 6(i) should be a blank card.

** Data description - 6(i) - for data series NO+1 is not needed but a blank card is essential.

AUTHOR'S NOTE

The GOODS program has been tested on a large number of graphs of different sorts and has worked satisfactorily. It will be appreciated however that there are far too many combinations of facilities for each to be tested individually. If errors do occur under certain conditions it would be very helpful if details could be sent to me together with a specimen output showing the fault. This will enable a correction to be made.

A copy of the program in its present form has been appended together with an example of output.

A further copy is kept on file both on the University of Aston computer and on the YIM computer.

At Aston the program in binary form can be obtained by use of the STOREDPROG macro. The instructions are,

```
Job Card.
STOREDPROG b F249GDS, B500
****
DOC DATA
Data as described in section 3.
****
```

All the above cards are punched starting in column one. Further information on the STOREDPROG macro may be obtained from the computer centre.

At YIM the program is stored on E. D. S. No. 000403 in file PROGRAMbFOR4 sub file GOODS in source form using ICL utility program # XMED. It may be accessed using the statements,

```
SHORT LIST
PROGRAM (B500)
INPUT 1 = CRO
OUTPUT 3 = LP7
TRACE 2
END
READ FROM (ED, PROGRAMbFOR4. GOODS)
```

It should be compiled using # XFAT compiler.

ADDENDUM

Four new facilities have been added to the GOODS program. They may be obtained by use of the control card as described below. (The paragraph numbering system in section 2 is extended to the cases below).

2. 14 IFMAT: FLOATING POINT LISTING OF DATA SERIES (AND FUNCTIONS)

This facility is obtained by punching the digit "1" in column 16 of the control card.

Normally the listing (see also 2. 12) takes the form (nnn. n) where "n" is a decimal digit. Thus the number 1234. 56 on input would be listed as 1234. 6 although its actual value would be used in any calculation. The largest positive number that can be output in this way is 999999. 9 and the smallest negative number -99999. 9. By setting IFMAT=1, the listing would represent 1234. 56 as 0. 12346E04, that is 0. 12346 x 10^4. (In each case rounding up occurs when the first truncated digit is 5 or more).

Thus the floating point facility would normally be used to represent numbers outside the range that can be accommodated in the normal format, or when greater accuracy of representation is required. For example 0. 000051 on input would be listed as 0. 0 normally, but as 0. 51000E-04 in floating point format.

2. 15 ISEP. INPUT OF CO-ORDINATES AS SEPARATE SERIES

This facility is obtained by punching the digit "1" in column 17 of the control card.

Normally data is input as (x, y) co-ordinates punched on the same card (see 3. 1. 3). For some purposes this is rather inconvenient, for example when a number of time series are to be plotted against the same base period. Similarly in the preparation of "scattergrams" when a number of variables are plotted in turn one against the other, the normal method of input would require an excessive amount of punching in order that each combination of co-ordinates was available.

If the ISEP facility is used data is assembled in the usual way as described in Table 3. 2 up to and including sequence No. 6(ii), which is the card containing the number of co-ordinates.

The abscissae are then punched in free format (F) -(See 3. 1. 2) with 10 entries to the card.

Cont'd. . . .

The corresponding ordinates are then punched in exactly the same way starting a new card for the first entry of the ordinate series.

This procedure including the card containing the number of co-ordinates is repeated a further (No-1) times as described in Table 3.2. at sequence No. 6.

2.16. ILINE: VARIABLE LENGTH GRAPHS

This facility is obtained by punching the digit "1" in column 18 of the control card, AND requires the insertion of an extra card in this data set as described below.

The basic GOODS program is designed to plot points on a grid composed of 100 x 100 graduations, and it will scale any input data to fit on these axes. However, the graudations on the x-axis coincide with the lines on the lineprinter output paper and thus the length of the graph is only limited by the length of paper on the printer. This may be made use of by setting ILINE=1, which then permits the user to vary the length of the 'x' axis between 1 and a maximum of 1000 graduations (lines), or approximately 15 pages of output as compared with the standard graph which occupies approximately two pages.

If the ILINE facility is used the user must input the following information:-

(i) Minimum abscissa value
(ii) Maximum abscissa value
(iii) Number of lines of output required (in the range 1-1000)

The significance of (i) and (ii) is in the scaling operation carried out by the program, which treats the data as though it were drawing a normal 100 graduation graph. The user thus sets the maximum value of abscissa as the value to appear against the 100th graduation on the graph. The program then scales the abscissae with respect to the minimum and maximum values input, even though some of the values exceed the stated maximum and selects the appropriate scale (see 2.6). The user must then ensure that a sufficient number of lines will be output to completely represent the data series which otherwise will be truncated at the number of lines specified. If more lines are specified than are required to represent the data, the effect will be to merely extend the 'x' axis.

Input Requirements

The information in (i)-(iii) above should be punched on one card, each number being separated from the others by at least one space. Otherwise position on the card is unimportant. The values

Cont'd....

of (i) and (ii) may be integer or fractional (i. e. contain a decimal point) but (iii) must be integer.

The card itself should be positioned in the data set after any card required at sequence No. 5, Table 3.2, and before starting sequence No. 6.

Example

It is desired to plot the weekly demand for a product over the period of one year. The week Nos. will be plotted as abscissae.

Since there are 52 weeks in a year, the program will choose a scale of 0-100 (See 2. 6) if the graph is plotted in the normal way. This would mean that only half the axis would be used. To avoid this, set ILINE to one using the control card and input the following values,

0 50 107

on a card positioned as described above. This indicates that the 100th graduation will be scaled as value 50 and that two graduations are equivalent to one abscissa unit. Since 52 units are to be plotted 105*graduations are needed and two more have been added for visual balance, making the third entry 107.

NB The ILINE facility has not relaxed the restriction that a data series cannot contain more than 100 points. (As this restriction exists merely to limit the amount of computer core store regularly required it could be relaxed in particular cases).

* Note that 101 points are required to span the range 0-100.

2.17. AUTOMATIC CENTRING OF TITLE INFORMATION

This facility is not obtained using the control card.

The method of preparing title information described in (2.12.1.) may still be used but with the restriction that only cols. 2-77 may be used for the title. The function of col. 1 in each title card remains unchanged (2.12.1.)

This new facility saves the user having to centre his information on the title card. Instead the title is started in col. 2, and the number of the last column punched with title information is punched in col. 79 and 80. (If this number is less than 10 it must be punched in col. 80). The information will be centred on output.

If cols. 79 and 80 are left blank the program will assume that the title has been centred by the user as described in (2.12.1.).

COMPUTER PROGRAMS

Contents:-

Note: All the programs in this section are written in I.C.L. 1900 series FORTRAN - see I.C.L. Manual ref. no. TL1167, "FORTRAN", March, 1969, and I.C.L. Manual ref. no. 4149, "FORTRAN: 32K Disc Compiler", March, 1969.

APPENDIX P1

Program Description: PSALM main program (SLM1) - see Appendices M1 & M2 for details of operation and organisation.

```
PROGRAM(SLM1)
OVERLAY (1,1) STOCK SUMMARY
OVERLAY(1,2) STOCK PROJECTION,PRODUCTION ALLOCATION
OVERLAY(1,3) FORECAST MONITOR,CUSUM,PRODUCTION MONITOR
OVERLAY(1,3) DEMAND STATISTICS
OVERLAY(2,1) GOODS1
OVERLAY(2,2) GOODS2
OVERLAY(2,3) GOODS3
OVERLAY(2,4) INPUT,SEQ CHECK
OVERLAY(2,5) FREAD,FWRITE,FCOPY
OVERLAY(2,6) CONCARD
OVERLAY(3,1) SCALE
OVERLAY(3,2) SELFORMAT,DYN FORMAT
OVERLAY(3,3) EXPONENT
OVERLAY(4,1) IVRTRANK
OVERLAY(4,2) CHANJSCL
INPUT 1=CR1
OUTPUT 3=LP1
USE 2=/ARRAY
INPUT 4=MT1/UNFORMATTED(UNKNOWNASYET)/1024
CREATE 5=MT2/UNFORMATTED(UNKNOWNASYET)/1024
USE 6=ED1/UNFORMATTED
COMPRESS INTEGER AND LOGICAL
TRACE 2
END
```

```
      MASTER ALLOCATION
      INTEGER CR1,DBF
      INTEGER ED1
      COMMON/INPUT OUTPUT/CR1,LP1,DBF,MT1,MT2,ED1
      COMMON/APAGE/NO WEEK,NO PAGE,ADATE,ATIME
      CR1=1
      DBF=2
      LP1=3
      MT1=4
      MT2=5
      ED1=6
      CALL STOCK SUMMARY
      CALL STOCK PROJECTION
      CALL PRODUCTION ALLOCATION
      CALL FORECAST MONITOR
      CALL PRODUCTION MONITOR
      IF(NO WEEK-NO WEEK/13*13.EQ.0) CALL DEMAND STATISTICS
      ENDFILE 5
      STOP
      END
```

```
             SUBROUTINE STOCK SUMMARY
             INTEGER CR1,DBF
             INTEGER ED1
             DIMENSION IPERM(1032),INDEM(3),INPROD(3,4),MINLVL(16,3),SPROD(16,3
            1,4),INDIC(16,3),PSTOC TOT(3),PROD TOT(3),            PROD(16,3)
            2,INSTOC(3)
             COMMON/INPUT OUTPUT/CR1,LP1,DBF,MT1,MT2,ED1
             COMMON/PERM/ICODE(16,3),PTYPE(3),IPNAME(5,16,3),IFACT(4),ISITE
            1                 (3,4),CSTOC(16,3),MAXLVL(16,3),ROPROD(16,3,4),
            2                 DEM SUM(16,3),DEM SQ(16,3),WKBUDG
            A                 ,PDEM(16,3),PREQ(4),RE ORD(16,3),IPSTOP(16,3),
            B                 TOT STOCK,TOT PROD,SPROD TOT(3,4),SITE TOT(4),
            C                 TOT DEM,SMPROD
             COMMON /APAGE/NO WEEK,NO PAGE,ADATE,ATIME
             COMMON/BUFFER/NWORDS,NTEMP,NPERM,NRECD,IBUFF(1016)
             COMMON/ERROR/ICOPY,ISEQ NO,INPUT ERROR
             DATA INDEM(1)/'*DMX*DMZ*DMY'/,INPROD(1)/'*PLX*PLZ*PLY*PKX*PKZ*PKY*
            1PBX*PBZ*PBY*PAX*PAZ*PAY'/,IBLANK/'    '/,IAST1/'*   '/,IAST2/'**
            2'/,INSTOC(1)/'*STX*STZ*STY'/
             EQUIVALENCE(IBUFF(1),MINLVL(1)),(IBUFF(49),SPROD(1)),(IBUFF(433),
            1INDIC(1)),(IBUFF(481),PSTOC TOT(1)),(IBUFF(487),PROD TOT(1)),
            2                    (IBUFF(589),PROD(1)),(IPERM(1),ICODE(1))
             EXTERNAL CARD CHECK
      C
      C          INITIALISATION STATEMENTS
      C
             CALL TIME(ATIME)
             CALL DATE(ADATE)
             NO PAGE=1
             NRECD=0
             INPUT ERROR=0
             CALL FTRAP(CARD CHECK)
             ISEQ CNT=1
             READ(CR1,10) NO WEEK,IDENT,ISEQ NO
      10     FORMAT(I0,T75,A4,I2)
             CALL SEQ CHECK(IDENT,ISEQ NO,4H*WKN,ISEQ CNT,LP1)
      C
      C          SPECIFY MAG. TAPE INPUT AND OUTPUT FILES.
      C
             LAST WK=NO WEEK-1
             IF(LAST WK.EQ.0) LAST WK=52
             CALL FILE(MT1,12HSLM1DATAFILE,LAST WK,15)
      CALL FILE(MT2,12HSLM1DATAFILE,NO WEEK,15)
C
C         READ INPUT FROM FILE AND COPY INTO /PERM/
C
      CALL FREAD(MT1,10)
        DO 101 I=1,NPERM
        IPERM(I)=IBUFF(I)
101     CONTINUE
      IPERM NO=NPERM
      NWORD1=NWORDS
      NTEMP1=NTEMP
      CALL FREAD(MT1,20)
        DO 1011 I=1,NPERM
        IPERM(IPERM NO+I)=IBUFF(NTEMP+I)
1011    CONTINUE
C
C         IF CURRENT WEEK IS START OF A NEW QUARTER, ZEROISE 'DEM SUM'
C         AND 'DEM SQ'.
C
      IF(NO WEEK-NO WEEK/13*13.NE.1) GO TO 102
      DEM SUM(1,1),DEM SQ(1,1)=0.0
      CALL FMOVE(DEM SUM(1,1),DEM SUM(2,1),47)
      CALL FMOVE(DEM SQ(1,1),DEM SQ(2,1),47)
102   CONTINUE
C
C         READ DEMAND AND PRODUCTION FIGURES FOR PREVIOUS WEEK.
C         INPUT ORDER:
C
C                             1) DEMAND
C
C                                  TABLE'A'
C                                  Y.T.W.
C                                  B.S.1386
C
C                             2) PRODUCTION
C
C                                  LEEDS          )  INPUT ORDER OF PRODUCTS
C                                  KIRKBY         )  WITHIN FACTORY ORDERING
C                                  BARRHEAD       )  IS AS FOR DEMAND IN (1)
C                                  ALAN EVERITT   )  ABOVE.
C
          DO 11 I=1,3
        PDEM(16,I)=0.0
        CALL INPUT(PDEM(1,I),15,INDEM(I),CR1,DBF,LP1)
11      CONTINUE
      CONTINUE
        DO 12 I=1,4
        DO 12 J=1,3
        SPROD(16,J,I)=0.0
        CALL INPUT(SPROD(1,J,I),15,INPROD(J,I),CR1,DBF,LP1)
12      CONTINUE
C
C         READ STOCK AT END OF PREVIOUS WEEK.
C
        DO 121 I=1,3
        CSTOC(16,I)=0.0
        CALL INPUT(CSTOC(1,I),15,INSTOC(I),CR1,DBF,LP1)
121     CONTINUE
      IF(INPUT ERROR.EQ.0) GO TO 123
      WRITE(LP1,122)
122   FORMAT(1H0,'THE ABOVE CARD(S) SHOULD BE CHECKED FOR MORE THAN ONE
     1ERROR.'/1H0,'PROGRAM ABANDONED IN ORDER THAT AMENDMENTS CAN BE MAD
     2E.')
      STOP 'DATA ERROR - ABANDON RUN'
123   CALL FRESET
C
C         CALCULATE TOTAL PRODUCTION OF EACH PRODUCT 'PROD'.
C
        DO 13 I=1,3
        DO 13 J=1,16
        PROD(J,I)=0.0
          DO 13 K=1,4
          PROD(J,I)=PROD(J,I)+SPROD(J,I,K)
13      CONTINUE
C
C         CALCULATE TOTAL DEMAND 'TOT DEM'.
C
      TOT DEM=0.0
        DO 14 I=1,3
        DO 14 J=1,16
        TOT DEM=TOT DEM+PDEM(J,I)
        IPSTOP(J,I)=0
        INDIC(J,I)=IBLANK
14      CONTINUE
```

```
C
C           CHECK STOCK LEVELS, CALCULATE RE-ORDER QUANTITIES, SET STOCK
C           LEVEL INDICATORS (*) AND (**) FOR PRINT OUT,AND SET PRODUCTION
C           MARKERS TO SUPPRESS PRODUCTION OF OVER STOCKED PRODUCTS.
C
          DO 15 I=1,3
          DO 15 J=1,16
          IF(CSTOC (J,I).GE.MINLVL(J,I))GO TO 141
          INDIC(J,I)=IAST1
          CRQPROD=RQPROD(J,I,1)+RQPROD(J,I,2)+RQPROD(J,I,3)+RQPROD(J,I,4)
          IF(CRQPROD+CSTOC(J,I).GE.MINLVL(J,I)) GO TO 15
          RE ORD(J,I)=MINLVL(J,I)-CRQPROD-CSTOC(J,I)
          IF(RE ORD(J,I).LT.10.0) RE ORD(J,I)=10.0
          GO TO 15
  141     RE ORD(J,I)=0.0
          IF(CSTOC (J,I).LE.MAXLVL(J,I)) GO TO 15
          IPSTOP(J,I)=1
          INDIC(J,I)=IAST2
  15      CONTINUE
C
C           CALCULATE FACTORY PRODUCTION TOTALS 'SPROD TOT'.
C
          DO 16 K=1,4
          DO 16 I=1,3
          SPROD TOT(I,K)=0.0
            DO 16 J=1,16
            SPROD TOT(I,K)=SPROD TOT(I,K)+SPROD(J,I,K)
  16      CONTINUE
C
C           CALCULATE TOTAL STOCK 'TOT STOCK', TOTAL PRODUCTION 'TOT PROD'
C           AND TOTALS BY PRODUCT TYPE 'PSTOC TOT' AND 'PROD TOT'
C
          DO 17 I=1,3
          PSTOC TOT(I),PROD TOT(I)=0.0
            DO 17 J=1,16
            PSTOC TOT(I)=PSTOC TOT(I)+CSTOC(J,I)
            PROD TOT(I)=PROD TOT(I)+ PROD(J,I)
  17      CONTINUE
         TOT STOCK,TOT PROD=0.0
           DO 171 I=1,3
           TOT PROD=TOT PROD+PROD TOT(I)
           TOT STOCK=TOT STOCK+PSTOC TOT(I)
  171      CONTINUE
C
C            CALCULATE SITE PRODUCTION TOTALS.
C
           DO 172 I=1,4
           SITE TOT(I)=0.0
             DO 172 J=1,3
             DO 172 K=1,16
             SITE TOT(I)=SITE TOT(I)+SPROD(K,J,I)
  172      CONTINUE
C
C            UPDATE MONITORING VARIABLES 'DEM SUM' AND 'DEM SQ', AND
C            'PREQ'.
C
             DO 173 I=1,3
             DO 173 J=1,16
             DEM SUM(J,I)=DEM SUM(J,I)+PDEM(J,I)
             DEM SQ(J,I)=DEM SQ(J,I)+PDEM(J,I)*PDEM(J,I)
  173        CONTINUE
           CONTINUE
             DO 174 I=1,4
             PREQ(I)=0.0
             DO 174 J=1,3
             DO 174 K=1,16
             PREQ(I)=PREQ(I)+RQPROD(K,J,I)
  174        CONTINUE
C
C            WRITE TABLE 1.
C
          CALL PAGE(LP1)
          WRITE(LP1,18)
  18      FORMAT(///5X,'TABLE 1',T31,'CONSTRUCTION COPPER - STOCK AND',
         1' PRODUCTION SUMMARY BY PRODUCT'/5X,7(1H-),T31,61(1H-)///T12,'CODE
         2 NO.',T25,'PRODUCT NAME',T43,'DEMAND',T56,'PRODUCTION',T73,'MINI',
         3'MUM',T85,'CLOSING STOCK',T103,'MAXIMUM'/T42,'(TONNES)',T57,
         4'(TONNES)',T71,'STOCK LEVEL',T87,'(TONNES)',T101,'STOCK LEVEL'/
         5T72,'(TONNES)',T102,'(TONNES)'/)
          WRITE(LP1,19) ((ICODE(J,I),(IPNAME(L,J,I),L=1,5),PDEM(J,I),PROD
         1(J,I),MINLVL(J,I),CSTOC(J,I),INDIC(J,I),MAXLVL(J,I),J=1,16),
         2I=1,3)
  19      FORMAT(1H ,T13,I5,T22,4A4,A2,T43,F5.1,T58,F5.1,T75,I3,T88,
         1F5.1,1X,A2,T104,I4)
```

```
C
C          WRITE TABLE 2.
C
       CALL PAGE(LP1)
       WRITE(LP1,20) (PTYPE(I),PROD TOT(I),I=1,3),TOT PROD
20     FORMAT(11(/),  5X,'TABLE 2',T45,'PRODUCTION SUMMARY BY',
      1' PRODUCT TYPE'/5X,7(1H-),T45,34(1H-)///T42,'PRODUCT TYPE',
      2T65,'WEIGHT PRODUCED (TONNES)'/T42,12(1H-),T65,24(1H-)//
      33(T44,A8,T74,F6.1/),T44,8HTOTAL    ,T74,F6.1)
C
C          WRITE TABLE 3
C
       WRITE(LP1,21)  (PTYPE(I),PSTOC TOT(I),I=1,3),TOT STOCK
21     FORMAT(13(/),5X,'TABLE 3',T48,'STOCK SUMMARY BY PRODUCT TYPE'
      1/5X,7(1H-),T48,29(1H-)///T42,'PRODUCT TYPE',T71,'STOCK (TONNES)'
      2/T42,12(1H-),T71,14(1H-)//3(T44,A8,T75,F6.1/),T44,8HTOTAL    ,T75,
      3F6.1)
C
C          WRITE TABLE 4
C
         DO 22 K=1,4
         CALL PAGE(LP1)
         IF(K-1) 2111,0,2111
         WRITE(LP1,211)
211        FORMAT(  //5X,'TABLE 4',T42,'PRODUCTION SUMMARY BY ',
       1 'MANUFACTURING SITE'/5X,7(1H-),T42,40(1H-)//)
         GO TO 212
2111     WRITE(LP1,2112)
2112     FORMAT(  //5X,'TABLE 4 CONT''D'/5X,7(1H-)//)
212      WRITE(LP1,213) K,IFACT(K),(ISITE(L,K),L=1,3)
213      FORMAT(        5X,1H(,I1,') PRODUCTION AT FACTORY NO. ',
       1 I2,' - ',3A4/5X,32(1H-)///T37,'CODE NO.',T56,'PRODUCT ',
       2 'SIZE',T74,'WEIGHT PRODUCED (TONNES)'/T58,'AND TYPE'/)
         WRITE(LP1,214) ((ICODE(J,I),(IPNAME(L,J,I),L=1,5),SPROD(J,I,K),
       1 J=1,16),I=1,3)
214      FORMAT(37X,I5,T53,4A4,A2,T81,F5.1)
22       CONTINUE
C
C          WRITE TABLE 5
C
       CALL PAGE(LP1)
       WRITE(LP1,23)
```

```
23      FORMAT( 7(/),5X,'TABLE 5',T42,'PRODUCT TYPE TOTALS BY ',
       1'MANUFACTURING SITE'/5X,7(1H-),T42,41(1H-)///T27,'FACTORY',
       2' NO.',T46,'FACTORY SITE',T66,'PRODUCT TYPE',T82,'WEIGHT ',
       3'PRODUCED (TONNES)'//)
        WRITE(LP1,24) (IFACT(K),(ISITE(L,K),L=1,3),(PTYPE(I),SPROD TOT(I,K
       1                ),I=1,3),SITE TOT(K),K=1,4)
24      FORMAT((T31,I2,T46,3A4,3(T68,A8,T87,F6.1/),T68,'TOTAL    ',
       1T87,F6.1/))
C
C           WRITE RECD. 2 TO BACKING FILE TO PRESERVE ITS CONTENTS.
C
          DO 25 I=1,NPERM
          IBUFF(NTEMP+I)=IPERM(IPERM NO+I)
25        CONTINUE
        CALL FWRITE(ED1)
C
C           COPY 'IPERM' BACK INTO 'IBUFF' AND WRITE TO TEMPORARY BACKING
C           FILE.
C
        NWORDS=NWORD1
        NTEMP=NTEMP1
        NPERM=IPERM NO
        NRECD=10
          DO 26 I=1,IPERM NO
          IBUFF(I)=IPERM(I)
26        CONTINUE
        CALL FWRITE(ED1)
        RETURN
        END
```

```
      SUBROUTINE STOCK PROJECTION
      INTEGER PT,CR1,DBF
      INTEGER ED1
      DIMENSION      FHOR(104),SHOR(104),PHOR(104),PADJ(104),SDES(5,3)
      DIMENSION X LABEL(4),Y LABEL(4)
      DIMENSION TEMP1(3102),TEMP2(191),TEMP3(8)
      COMMON/INPUT OUTPUT/CR1,LP1,DBF,MT1,MT2,ED1
      COMMON/PERM/ICODE(16,3),PTYPE(3),IPNAME(5,16,3),IFACT(4),ISITE
     1                  (3,4),CSTOC(16,3),MAXLVL(16,3),RQPROD(16,3,4),
     2                  DEM SUM(16,3),DEM SQ(16,3),WKBUDG
     A                  ,PDEM(16,3),PREQ(4),RE ORD(16,3),IPSTOP(16,3),
     B                  TOT STOCK,TOT PROD,SPROD TOT(3,4),SITE TOT(4),
     C                  TOT DEM,SMPROD
      COMMON /APAGE/NO WEEK,NO PAGE,ADATE,ATIME
      COMMON/BUFFER/NWORDS,NTEMP,NPERM,NRECD,IBUFF(1016)
      COMMON/GRAPH/LINE(106),IDATA(3,100,5),NOHORMK,IHORMK(16),HORMK(15)
     1             ,NOVRTMK,IVRTMK(16),VRTMK(15),DATAR(3,100,5),FUNC(100
     2             ,5),IFUNC(100,5),NOENTRY(5)
      COMMON/EXTRA/II,NXMIN,NOLINES,IMD,IVCHE K,PT(10),DATAL(100),FDES(5
     1             ,5),DES(5,5),XMIN,XMAX,YMIN,YMAX,JMKR(5),NK(5),JA(5),
     2             JX(5),IFSET,X AXIS(4),Y AXIS(4),STD FORMAT(11)
      COMMON/CONTROL/NO,NF,IPDS,IOF,ISETSCL,ISELPT,LORD,LABS,ISUPRZ,LOGX
     1,LOGY,LNX,LNY,IFMAT,      ILINE,IXSCALE
      DATA SDES(1,1)/ 40HFORECAST DEMAND (TONNES)                  /,
     1     SDES(1,2)/ 40HSMOOTHED WEEKLY PRODUCTION (TONNES)       /,
     2     SDES(1,3)/ 40HPROJECTION OF TOTAL CO. STOCKS (TONNES) /
      DATA X LABEL(1)/'WEEK NO.                            '/,
     1     Y LABEL(1)/'TONNES (SEE SYMBOL KEY)             '/
      EQUIVALENCE  (IBUFF(1),FHOR(1)),(IBUFF(209),SHOR(1)),(IBUFF(417),
     1              PHOR(1)),(IBUFF(625),PADJ(1)),(IBUFF(835),BUFF STOCK
     2              ),(IBUFF(835),IBASE WK),(IBUFF(836),IHOR WK)
      EQUIVALENCE (LINE(2),TEMP1(1)),(II,TEMP2(1)),(NO,TEMP3(1))
C
C          ZEROISE GRAPH PLOTTER COMMON BLOCKS - 'GRAPH','EXTRA' AND
C                                                           'CONTROL'.
C
      LINE(1)=0
      TEMP1(1),TEMP2(1),TEMP3(1)=0.0
      CALL FMOVE(TEMP1(1),TEMP1(2),3101)
      CALL FMOVE(TEMP2(1),TEMP2(2),190)
      CALL FMOVE(TEMP3(1),TEMP3(2),7)
C
```

```
C         INPUT FROM FILE, HORIZ. PERIOD OF FORECASTS 'FHOR', HORIZ.
C         PERIOD OF STOCKS 'SHOR', HORIZ. PERIOD OF TOTAL PRODUCTION
C         'PHOR' AND PRODUCTION ADJUSTMENTS 'PADJ' FOR HORIZON PERIOD.
C         ALSO INPUT BUFFER STOCK 'BUFF STOCK' ,   BASE LINE WEEK,
C         'IBASE WK', HORIZON WEEK 'IHOR WK'.
C
      CALL FREAD(MT1,30)
C
C         CALCULATE  WEEK NO. RELATIVE TO BASE WEEK 'IWKREL' AND INSERT
C         ACTUAL VALUES FOR PRODUCTION  ,  DEMAND AND STOCK.
C
      IWK REL=NO WEEK-IBASE WK+1
      FHOR(IWK REL)=TOT DEM
      SHOR(IWK REL)=TOT STOCK
      PHOR(IWK REL)=TOT PROD
C
C         CALC. TOTAL 'ROPROD' AND SET PRODUCTION FOR COMING WEEK
C         'NEXT WK' TO THIS.
C
      TOT ROPROD=0.0
        DO 104 I=1,3
        DO 104 J=1,16
        DO 104 K=1,4
        TOT ROPROD=TOT ROPROD+ROPROD(J,I,K)
104     CONTINUE
106   NEXT WK=IWK REL+1
      PHOR(NEXT WK)=TOT ROPROD+PADJ(NEXT WK)
C
C         CALCULATE STOCK FOR 'NEXT WK'
C
      SHOR(NEXT WK)=SHOR(IWK REL)+PHOR(NEXT WK)-FHOR(NEXT WK)
C
C         CALCULATE TOTAL FORECAST DEMAND TO HORIZ., AND NET ADJUSTMENT.
C
      ANET ADJ,TOT FDEM=0.0
        DO 11 I=IWK REL+2,IHOR WK
        ANET ADJ=ANET ADJ+PADJ(I)
        TOT FDEM=TOT FDEM+FHOR(I)
11      CONTINUE
C
C         CALCULATE SMOOTHED PRODUCTION LEVEL
C
```

```
      SMPROD=(TOT FDEM-(SHOR(NEXT WK)-BUFF STOCK))/(IHOR WK-IWK REL-1)
      PRODN LIMIT=0.85*WKBUDG
      IF(SMPROD-PRODN LIMIT) 0,112,111
      SMPROD=PRODN LIMIT
      GO TO 112
111   PRODN LIMIT=1.25*WKBUDG
      IF(SMPROD.GT.PRODN LIMIT) SMPROD=PRODN LIMIT
112     DO 12 I=IWK REL+2,IHOR WK
        PHOR(I)=SMPROD+PADJ(I)
12      CONTINUE
C
C       PRODUCE STOCK PROJECTION
C
        DO 13 I=IWK REL+2,IHOR WK
        SHOR(I)=SHOR(I-1)+PHOR(I)-FHOR(I)
13      CONTINUE
C
C         COPY DATA SERIES INTO DATAR
C
        DO 14 I=1,IHOR WK
        DATAR(1,I,1)=I
        DATAR(2,I,1)=FHOR(I)
        DATAR(1,I,2)=I
        DATAR(2,I,2)=PHOR(I)
        DATAR(1,I,3)=I
        DATAR(2,I,3)=SHOR(I)
14      CONTINUE
C
C         SET REMAINING PARAMETERS FOR GOODS.
C
      CALL CONCARD(3,0,0,0,0,1,1HI,1H0,0,0,0,0,0,0,1,IBASE WK-1)
      NO ENTRY(1),NO ENTRY(2),NO ENTRY(3)=IHOR WK
      NOHORMK=1
      HORMK(1)=BUFF STOCK
      XMIN=0.0
      XMAX=50.0
      NO LINES=2*IHOR WK+6
C
C         COPY LINE DESCRIPTORS
      CALL FMOVE(SDES(1,1),DES(1,1),15)
C
C         COPY AXIS DESCRIPTORS INTO 'X AXIS' AND 'Y AXIS'.
```

```
C
       CALL FMOVE(X LABEL,X AXIS,4)
       CALL FMOVE(Y LABEL,Y AXIS,4)
       CALL PAGE(LP1)
       WRITE(LP1,15) NO WEEK
15     FORMAT(//// 41X,'COMPANY STOCK PROJECTION AT WEEK NO. ',I3/
      141X,40(1H-)/)
       CALL GOODS OVERLAY
       IF(ANET ADJ.NE.0.0) WRITE(LP1,16) ANET ADJ
16     FORMAT(////3X,116(1H-)//5X,'WARNING: THE NETT WEEKLY PRODUCTION AD
      1JUSTMENT SHOULD BE ZERO, BUT IT HAS BEEN FOUND TO BE ',F7.1,
      2' TONNES.'/14X,'THIS HAS PRODUCED A LOAD FILLING SITUATION. (LOAD
      3SHEDDING IF THE NETT ADJUSTMENT IS NEGATIVE.)'/40X,'(FOR FURTHER D
      4ETAILS SEE ''PSALM'' MANUAL.)'//3X,116(1H-))
       CALL FWRITE(ED1)
       RETURN
       END
      SUBROUTINE PRODUCTION ALLOCATION
      INTEGER PT,CR1,DBF
      INTEGER ED1
      DIMENSION PMAKE(16,3,4),PBUDG(16,3,4),TBUDG(16,3,4)
      DIMENSION IPERM(1030)
      COMMON/INPUT OUTPUT/CR1,LP1,DBF,MT1,MT2,ED1
      COMMON /APAGE/NO WEEK,NO PAGE,ADATE,ATIME
      COMMON/PERM/ICODE(16,3),PTYPE(3),IPNAME(5,16,3),IFACT(4),ISITE
     1                (3,4),CSTOC(16,3),MAXLVL(16,3),RQPROD(16,3,4),
     2                DEM SUM(16,3),DEM SQ(16,3),WKBUDG
     A                ,PDEM(16,3),PREQ(4),RE ORD(16,3),IPSTOP(16,3),
     B                TOT STOCK,TOT PROD,SPROD TOT(3,4),SITE TOT(4),
     C                TOT DEM,SMPROD
      COMMON/BUFFER/NWORDS,NTEMP,NPERM,NRECD,IBUFF(1016)
      EQUIVALENCE (IBUFF(1),PBUDG(1)),(IBUFF(387),TBUDG(1)),(RQPROD(1),
     1            PMAKE(1)),(IPERM(1),ICODE(1))
C
C         INPUT PRODUCTION BUDGET 'PBUDG' AND WEEKLY BUDGET 'WKBUDG'.
C
      CALL FREAD(MT1,40)
C
C         ZEROISE 'PMAKE'.
C
      PMAKE(1,1,1)=0.0
      CALL FMOVE(PMAKE(1,1,1),PMAKE(2,1,1),191)
C
C         ADJUST BUDGETED MAKE USING SMOOTHED PROD. LEVEL 'SMPROD'.
C         ADJUSTED VALUE STORED AS 'TBUDG'.
C
      ADJ=SMPROD/WKBUDG
        DO 11 K=1,4
        DO 11 I=1,3
        DO 11 J=1,16
        TBUDG(J,I,K)=PBUDG(J,I,K)*ADJ
11      CONTINUE
C
C         ALLOCATE PRODUCTION TO FACTORIES. IF 'RE ORD' IS LESS THAN
C         'TBUDG' THEN 'PMAKE' IS SET TO 'TBUDG'. IF 'RE ORD' IS
C         GREATER THAN 'TBUDG' THEN 'PMAKE' IS SCALED TO GIVE 'RE ORD'.
C         IF PRODUCT (J,I) IS NOT BUDGETTED FOR IN CURRENT QUARTER,
C         'RE ORD' IS MADE AT FACTORY OF FIRST PREFERENCE.
C
```

```
        DO 13 I=1,3
        DO 13 J=1,16
        IF(IPSTOP(J,I)-1) 12011,0,12011
        PMAKE(J,I,1),PMAKE(J,I,2),PMAKE(J,I,3),PMAKE(J,I,4)=0.0
        GO TO 13
12011   SUM=0.0
          DO 1201 K=1,4
          PMAKE(J,I,K)=TBUDG(J,I,K)
          SUM=SUM+PMAKE(J,I,K)
1201      CONTINUE
        IF(RE ORD(J,I).GT.0.0.AND.SUM.EQ.0.0) GO TO(121,122,123),I
        IF(RE ORD(J,I).LE.SUM) GO TO 13
        AMULT=RE ORD(J,I)/SUM
          DO 1202 K=1,4
          PMAKE(J,I,K)=PMAKE(J,I,K)*AMULT
1202      CONTINUE
        GO TO 13
121     GO TO (124,124,124,124,125,126,125,125,125,125,125,125,124,124,
     1        124,124),J
122     GO TO (124,124,124,124,125,125,125,125,125,125,125,125,124,124,
     1        124,124),J
123     GO TO (124,124,125,125,125,125,125,125,125,125,125,125,124,124,
     1        124,124),J
124     PMAKE(J,I,1)=RE ORD(J,I)
        GO TO 13
125     PMAKE(J,I,2)=RE ORD(J,I)
        GO TO 13
126     PMAKE(J,I,3)=RE ORD(J,I)
13      CONTINUE
C
C         CALCULATE TOTAL PRODUCTION REQUIRED 'TOT MAKE'.
C
      TOT MAKE=0.0
        DO 13001 K=1,4
        DO 13001 I=1,3
        DO 13001 J=1,16
        TOT MAKE=TOT MAKE+PMAKE(J,I,K)
13001   CONTINUE
C
C         IF NECESSARY ADJUST 'PMAKE' TO FALL WITHIN THE RANGE 85-125%
C         OF WEEKLY PRODUCTION BUDGET 'WKBUDG'.
C
```

```
            PRODN LIMIT=0.85*WKBUDG
            IF(TOT MAKE-PRODN LIMIT) 0,13005,13003
            AMULT=PRODN LIMIT/TOT MAKE
              DO 13002 K=1,4
              DO 13002 I=1,3
              DO 13002 J=1,16
              PMAKE(J,I,K)=PMAKE(J,I,K)*AMULT
      13002   CONTINUE
            GO TO 13005
      13003 PRODN LIMIT=1.25*WKBUDG
            IF(TOT MAKE.LE.PRODN LIMIT) GO TO 13005
            AMULT=PRODN LIMIT/TOT MAKE
              DO 13004 K=1,4
              DO 13004 I=1,3
              DO 13004 J=1,16
              PMAKE(J,I,K)=PMAKE(J,I,K)*AMULT
      13004   CONTINUE
      13005 CONTINUE
      C
      C         WRITE TABLE 6
      C
              DO 14 K=1,4
              CALL PAGE(LP1)
              IF(K-1) 131,0,131
              WRITE(LP1,1301)
      1301    FORMAT(   //5X,'TABLE 6',T45,'PRODUCTION ALLOCATION BY FACTORY'
           1  /5X,7(1H-),T45,32(1H-)//)
              IALLOCN WEEK=NO WEEK+2
              IF(IALLOCN WEEK.GT.52) IALLOCN WEEK=IALLOCN WEEK-52
              GO TO 133
      131     WRITE(LP1,132)
      132     FORMAT(   //5X,'TABLE 6 CONT''D'/5X,7(1H-)//)
      133     WRITE(LP1,134) K,IFACT(K),(ISITE(L,K),L=1,3),IALLOCN WEEK
      134     FORMAT( 5X,1H(,I1,') ALLOCATION FOR FACTORY NO. ',I2,' - ',
           1  3A4,T91,30(1H*)/5X,33(1H-),T91,'* ALLOCATION FOR WEEK NO.',
           2  I3,' *'/T91,30(1H*)//
           3                    T37,'CODE NO.',T56,'PRODUCT SIZE',T76,
           4  'ALLOCATION (TONNES)'/T58,'AND TYPE'/)
              WRITE(LP1,135) ((ICODE(J,I),(IPNAME(L,J,I),L=1,5),PMAKE(J,I,K)
           1                    ,J=1,16),I=1,3)
      135     FORMAT(37X,I5,T53,4A4,A2,T80,F5.1)
      14      CONTINUE
```

```
C
C             EDIT THE BACKING FILE:
C
C                    1) WRITE RECORD NO. 4 TO BACKING FILE TO PRESERVE ITS
C                       CONTENTS.
C
C                    2) REWIND ED1.
C
C                    3) READ RECORD NO. 1 FROM THE FILE INTO /BUFFER/.
C                       (RECD. ORDER ON BACKING FILE IS: RECD.2;RECD.1;RECD.3;
C                                                         RECD.4)
C
C                    4) COPY 'PMAKE' INTO 'IBUFF'. (THIS WILL BECOME 'RQPROD'
C                                                   FOR THE COMING WEEK.)
C
C                    5) WRITE THE COMPLETED RECORD NO. 1 TO TAPE.
C
C                    6) REWIND.
C
C                    7) COPY RECORD NOS. 2,3&4 FROM BACKING FILE TO TAPE.
C
      CALL FWRITE(ED1)
      REWIND ED1
      NRECD=0
      CALL FREAD(ED1,10)
        DO 15 I=455,838
        IBUFF(I)=IPERM(I)
15      CONTINUE
      CALL FWRITE(MT2)
      REWIND ED1
      NRECD=0
      CALL FCOPY(ED1,MT2,20,0)
      CALL FCOPY(ED1,MT2,30,0)
      CALL FCOPY(ED1,MT2,40,0)
      RETURN
      END
      SUBROUTINE FORECAST MONITOR
      INTEGER CR1,DBF
      INTEGER ED1
      DIMENSION  PFCST(16,3),URV(16,3),ALRV(16,3),UDI(16,3),ALDI(16,3),
     1           PDIFF(16,3),ISTATUS(2,16,3),USCORE(16,3),ALSCORE(16,3),
     2           ITABLE(3,16,3),ITTABLE(3),ITSTATUS(2),CUM FCST(16,3),
     3           CUM DIFF(16,3),IFNAME(5)
      COMMON/INPUT OUTPUT/CR1,LP1,DBF,MT1,MT2,ED1
      COMMON /APAGE/NO WEEK,NO PAGE,ADATE,ATIME
      COMMON/PERM/ICODE(16,3),PTYPE(3),IPNAME(5,16,3),IFACT(4),ISITE
     1                   (3,4),CSTOC(16,3),MAXLVL(16,3),ROPROD(16,3,4),
     2                   DEM SUM(16,3),DEM SQ(16,3),WKBUDG
     A                   ,PDEM(16,3),PREQ(4),RE ORD(16,3),IPSTOP(16,3),
     B                   TOT STOCK,TOT PROD,SPROD TOT(3,4),SITE TOT(4),
     C                   TOT DEM,SMPROD
      COMMON/BUFFER/NWORDS,NTEMP,NPERM,NRECD,IBUFF(1016)
      DATA IFNAME(1)/'   TOTAL DEMAND      '/
      EQUIVALENCE (IBUFF(1),PFCST(1)),(IBUFF(97),URV(1)),(IBUFF(193),
     1              ALRV(1)),(IBUFF(289),UDI(1)),(IBUFF(385),ALDI(1)),
     2                                  (IBUFF(481),ISTATUS(1)),(IBUFF
     3            (577),USCORE(1)),(IBUFF(673),ALSCORE(1)),(IBUFF(769),
     4             TURV),(IBUFF(771),TALRV),(IBUFF(773),TUDI),(IBUFF(775
     5           ),TALDI),(IBUFF(777),ITSTATUS(1)),(IBUFF(779),TUSCORE),
     6             (IBUFF(781),TLSCORE),(IBUFF(783),CUM FCST(1)),(IBUFF
     7(879),CUM DIFF(1)),(IBUFF(975),TCUM FCST),(IBUFF(977),TCUM DIFF),
     8(IBUFF(979),TCUM DEM)
C
C          INPUT FROM FILE THE FORECAST VALUES OF WEEKLY DEMAND FOR EACH
C          PRODUCT 'PFCST' AND THEIR RELEVANT REFERENCE AND DECISION
C          VALUES 'URV','ALV','UDI' AND 'ALDI', ALSO
C          'ISTATUS','USCORE' AND 'ALSCORE'.
C
      CALL FREAD(MT1,50)
C
C          IF START OF NEW QUARTER ZEROISE 'CUM FCST' AND 'CUM DIFF'.
C
      IF(NOWEEK-NOWEEK/13.NE.1) GO TO 101
      CUM FCST(1,1),CUM DIFF(1,1)=0.0
      CALL FMOVE(CUM FCST(1,1),CUM FCST(2,1),47)
      CALL FMOVE(CUM DIFF(1,1),CUM DIFF(2,1),47)
      TCUM FCST,TCUM DIFF=0.0
      TCUM DEM=0.0
```

```
101   CONTINUE
C         CALCULATE THE DIFFERENCES BETWEEN ACTUAL AND FORECAST DEMAND
C
      WKFCST=0.0
        DO 11 I=1,3
        DO 11 J=1,16
        PDIFF(J,I)=PDEM(J,I)-PFCST(J,I)
        CUM FCST(J,I)=CUM FCST(J,I)+PFCST(J,I)
        CUM DIFF(J,I)=CUM DIFF(J,I)+PDIFF(J,I)
        WKFCST=WKFCST+PFCST(J,I)
11      CONTINUE
      TDIFF=TOT DEM-WKFCST
      TCUM FCST=TCUM FCST+WKFCST
      TCUM DIFF=TCUM DIFF+TDIFF
      TCUM DEM=TCUM DEM+TOT DEM
C
C         CARRY OUT DECISION INTERVAL SCHEME ON'DIFF'
C
        DO 12 I=1,3
        DO 12 J=1,16
        IF(URV(J,I).EQ. 0.0) GO TO 12
        CALL CUSUM(PDIFF(J,I),URV(J,I),UDI(J,I),ALRV(J,I),ALDI(J,I),
     1             ISTATUS(1,J,I),USCORE(J,I),ALSCORE(J,I),ITABLE
     2             (1,J,I))
12      CONTINUE
C
C        CARRY OUT DECISION INTERVAL SCHEME ON AGGREGATE FORECAST
C
      CALL CUSUM(     TDIFF,TURV,TUDI,TALRV,TALDI,ITSTATUS,TUSCORE,
     1          TLSCORE,ITTABLE)
C
C         WRITE TABLE 7.
C
      CALL PAGE(LP1)
      WRITE(LP1,22)
22    FORMAT( 6(/),5X,'TABLE 7',T47,'MONITOR OF FORECAST VARIABLES'
     1/5X,7(1H-),T47,29(1H-)//)
      WRITE(LP1,23)
23    FORMAT(5X,113(1H.)/5X,1H.,T27,1H.,T58,1H.,T89,1H.,T118,1H./
     15X,1H.,T13,8HVARIABLE,T27,1H.,T35,'WEEKLY (TONNES)',T58,1H.,
     2T64,'CUMULATIVE (TONNES)',T89,1H.,T102,'STATUS',T118,1H./
     35X,1H.,T27,1H.,T58,1H.,T89,1H.,T118,1H./5X,1H.,T27,1H.,T29,
           4'FORECAST  ACTUAL  DIFFFRENCE . FORFCAST  ACTUAL DIFFERFNCE  .',
           5T93,'IN    .  UNDFR .  OUT OF .'/5X,1H.,T27,1H.,T58,1H.,T89,
           6'. CONTROL . REVIFW . CONTROL .'/5X,1H.,T27,1H.,T58,1H.,T89,1H.,
           7T99,1H.,T108,1H.,T118,1H.)
            WRITE(LP1,24) IFNAME,WKFCST,TOT DFM,TDIFF,TCUM FCST,TCUM DEM,
           1TCUM DIFF,ITTABLE
            DO 241 I=1,2
            DO 241 J=1,16
            IF(URV(J,I).EQ.0.0) GO TO 241
            WRITE(LP1,24)    (IPNAME(L,J,I),L=1,5),PFCST(J,I),PDEM(J,I),PDIFF
           1(J,I),CUM FCST(J,I),DEM SUM(J,I),CUM DIFF
           2(J,I),(ITABLF(L,J,I),L=1,3)
      24  . FORMAT(5X,1H.,T8,4A4,A2,T27,1H.,     F8.1,     F9.1,     F9.1,
           1T58,1H.,     F8.1,     F8.1,     F9.1,T89,1H.,T94,A1,T99,1H.,
           2T105,A1,T108,1H.,T113,A1,T118,1H.)
     241    CONTINUF
            WRITE(LP1,25)
      25    FORMAT(5X,1H.,T27,1H.,T58,1H.,T89,1H.,T99,1H.,T108,1H.,T118,
           11H./5X,113(1H.))
            CALL FWRITF(MT2)
            RFTURN
            END
             SUBROUTINE CUSUM(DIFF,URV,UDI,ALRV,ALDI,ITSTATUS,USCORE,ALSCORE,
           1                    ITABLF)
            DIMENSION ITABLE(3),ISTATUS(2)
            DATA  IPLUS/4H+    /,IAST/4H*   /,MINUS/4H-   /,IBLANK/4H    /
            ITABLF(1),ITABLE(2),ITABLE(3)=IBLANK
      C
      C         VARIABLES
      C         ---------
      C
      C                'UDI'  UPPER DECISION INTERVAL
      C                'URV'  UPPER REFERENCE VALUE.
      C               'ALRV'  LOWER REFFRENCE VALUE.
      C               'ALDI'  LOWER DECISION INTERVAL.
      C
      C         'ISTATUS' IS ZERO WHEN THE VARIABLE IS IN CONTROL (DIFF.LE.
      C          REF. VALUES) AND UNITY WHEN AN ACCUMULATION IS IN PROCESS.
      C         'ISTATUS(1)' RELATES TO ACCUM. OF AN UPPER SCORE 'USCORE' AND
      C         'ISTATUS(2)' RELATES TO ACCUM. OF A LOWER SCORE 'ALSCORF'.
      C
      C
```

```
C         CHECK STATUS OF UPPER SCORE 'USCORE'
C
      IF(ISTATUS(1)) 11,0,11
      IF(DIFF-URV) 0,0,101
      USCORE=0.0
      GO TO 14
101   IF(DIFF-UDI) 0,13,13
      ISTATUS(1)=1
      USCORE=DIFF-URV
      GO TO 14
11    USCORE=USCORE+DIFF-URV
      IF(USCORE) 0,0,12
      ISTATUS(1)=0
      USCORE=0.0
      GO TO 14
12    IF(USCORE-UDI) 14,13,13
13    ITABLE(3)=IPLUS
      ISTATUS(1)=1
14    CONTINUE
C
C         CHECK STATUS OF LOWER SCORE 'ALSCORE'
C
      IF(ISTATUS(2)) 16,0,16
      IF(DIFF-ALRV) 15,0,0
      ALSCORE=0.0
      GO TO 19
15    IF(DIFF-ALDI) 18,18,0
      ISTATUS(2)=1
      ALSCORE=DIFF-ALRV
      GO TO 19
16    ALSCORE=ALSCORE+DIFF-ALRV
      IF(ALSCORE) 17,0,0
      ISTATUS(2)=0
      ALSCORE=0.0
      GO TO 19
17    IF(ALSCORE-ALDI) 18,18,19
18    ITABLE(3)=MINUS
      ISTATUS(2)=1
19    CONTINUE
C
C         PLACE CORRECT SYMBOLS IN ITABLE
C
```

```
      IF(ISTATUS(1).EQ.0.AND.ISTATUS(2).EQ.0) GO TO 20
      GO TO 21
 20   ITABLE(1)=IAST
      RETURN
 21   IF(ITABLE(3).NE.IBLANK) RETURN
      ITABLE(2)=IAST
      RETURN
      END
      SUBROUTINE PRODUCTION MONITOR
      INTEGER CR1,DBF,PT
      INTEGER ED1
      DIMENSION PDEV(13,4),BDEV(13,4),PRUDGET(4),      NQUARK(4),
     1           SDES(5,2),CURRENT PREQ(4)
      DIMENSION X LABEL(4),Y LABEL(4)
      DIMENSION TEMP1(3102),TEMP2(191),TEMP3(8)
      COMMON/INPUT OUTPUT/CR1,LP1,DBF,MT1,MT2,ED1
      COMMON/PERM/ICODE(16,3),PTYPE(3),IPNAME(5,16,3),IFACT(4),ISITE
     1                 (3,4),CSTOC(16,3),MAXLVL(16,3),ROPROD(16,3,4),
     2                 DEM SUM(16,3),DEM SQ(16,3),WKBUDG
     A                 ,PDEM(16,3),PREQ(4),RE ORD(16,3),IPSTOP(16,3),
     B                 TOT STOCK,TOT PROD,SPROD TOT(3,4),SITE TOT(4),
     C                 TOT DEM,SMPROD
      COMMON /APAGE/NO WEEK,NO PAGE,ADATE,ATIME
      COMMON/BUFFER/NWORDS,NTEMP,NPERM,NRECD,IBUFF(1016)
      COMMON/GRAPH/LINE(106),IDATA(3,100,5),NOHORMK,IHORMK(16),HORMK(15)
     1            ,NOVRTMK,IVRTMK(16),VRTMK(15),DATAR(3,100,5),FUNC(100
     2            ,5),IFUNC(100,5),NOENTRY(5)
      COMMON/EXTRA/II,NXMIN,NOLINES,IMD,IVCHE K,PT(10),DATAL(100),FDES(5
     1            ,5),DES(5,5),XMIN,XMAX,YMIN,YMAX,JMKR(5),NK(5),JA(5),
     2            JX(5),IFSET,X AXIS(4),Y AXIS(4),STD FORMAT(11)
      COMMON/CONTROL/NO,NF,IPDS,IOF,ISETSCL,ISELPT,LORD,LABS,ISUPRZ,LOGX
     1,LOGY,LNX,LNY,IFMAT,      ILINE,IXSCALE
      DATA  NQUARK(1)/'1ST 2ND 3RD 4TH '/,
     1      SDES(1,1)/40HDEVN. FROM PRODUCTION REQUEST (TONNES)  /,
     2      SDES(1,2)/40HDEVN. FROM PRODUCTION BUDGET (TONNES)   /
      DATA X LABEL(1)/'WEEK NO.                              '/,
     1     Y LABEL(1)/'TONNES (SEE SYMBOL KEY)               '/
      EQUIVALENCE (PDEV(1),IBUFF(1)),(BDEV(1),IBUFF(105)),(PBUDGET(1),
     1            IBUFF(209)),(CURRENT PREQ(1),IBUFF(217))
      EQUIVALENCE (LINE(1),TEMP1(1)),(II,TEMP2(1)),(NO,TEMP3(1))
C
C         ZEROISE GRAPH PLOTTER COMMON BLOCKS - 'GRAPH','EXTRA' AND
C                                                           'CONTROL'.
C
      LINE(1)=0
      TEMP1(1),TEMP2(1),TEMP3(1)=0.0
      CALL FMOVE(TEMP1(1),TEMP1(2),3101)
      CALL FMOVE(TEMP2(1),TEMP2(2),190)
      CALL FMOVE(TEMP3(1),TEMP3(2),7)
C
```

```
C         INPUT PRODUCTION DEVIATIONS 'PDEV', BUDGET DEVIATIONS 'BDEV',
C         PRODUCTION BUDGET 'PBUDGET', PRODUCTION REQUEST 'PREQ'.
C
      CALL FREAD(MT1,60)
C
C         OVER WRITE THE SERIES 'PDEV' AND 'BDEV' BY THEIR UPDATED
C         VERSION WHERE  PDEV(1)=PDEV(2) ETC.
C
        DO 11 I=1,4
          DO 1001 J=1,12
          PDEV(J,I)=PDEV(J+1,I)
          BDEV(J,I)=BDEV(J+1,I)
1001      CONTINUE
        PDEV(13,I)=SITE TOT(I)-CURRENT PREQ(I)
        BDEV(13,I)=SITE TOT(I)-PBUDGET(I)
11      CONTINUE
C
C         COPY 'PREQ' FOR USE AS 'CURRENT PREQ' NEXT WEEK.
C
        DO 111 I=1,4
        CURRENT PREQ(I)=PREQ(I)
111     CONTINUE
      CALL FWRITE(MT2)
C         CHECK IF END OF QUARTER. IF NOT,RETURN, OTHERWISE GRAPH
C         PRODUCTION PERFORMANCE.
C
      IF(NOWEEK-NO WEEK/13*13.NE.0) RETURN
      NQUATR=NO WEEK/13
        DO 14 I=1,4
        CALL PAGE(LP1)
        WRITE(LP1,12)  IFACT(I),NQUARK(NQUATR)
12      FORMAT(//// 34X,'PRODUCTION MONITOR OF FACTORY NO.',I3,
     1   ' FOR ',A4,'QUARTER.'/34X,53(1H-)/)
C
C         COPY DATA SERIES INTO DATAR
C
          DO 13 J=1,13
          DATAR(1,J,1)=J
          DATAR(2,J,1)=PDEV(J,I)
          DATAR(1,J,2)=J
          DATAR(2,J,2)=BDEV(J,I)
13        CONTINUE
```

```
C
C          SET REMAINING PARAMETERS FOR GOODS.
C
         CALL CONCARD(2,0,0,0,0,1,1H0,1H0,0,0,0,0,0,0,1,NO WEEK-13)
         NOHORMK,NOVRTMK=0
         NO ENTRY(1),NO ENTRY(2)=13
         XMIN=0.0
         XMAX=50.0
         NO LINES=30
C
C          COPY LINE DESCRIPTORS
C
         CALL FMOVE(SDES(1,1),DES(1,1),10)
C
C          COPY AXIS DESCRIPTORS INTO 'X AXIS' AND 'Y AXIS'.
C
         CALL FMOVE(X LABEL,X AXIS,4)
         CALL FMOVE(Y LABEL,Y AXIS,4)
         CALL GOODS OVERLAY
14       CONTINUE
       RETURN
       END
```

```
            SUBROUTINE INPUT(DUMMY,IDIM,ICHAR,CR1,DBF,LP1)
            INTEGER CR1,DBF
            DIMENSION DUMMY(IDIM),IFMAT(5)
            COMMON/ERROR/ICOPY,ISEQ COPY,INPUT ERROR
            CALL DEFBUF(DBF,20,IFMAT)
            ICOPY=ICHAR
            ISEQ CNT=0
            NCONST=IDIM/7*7
            IF(NCONST.EQ.0) GO TO 111
              DO 11 I=1,1+IDIM-7,7
              ISEQ CNT=ISEQ CNT+1
              ISEQ COPY=ISEQ CNT
              READ(CR1,101) (DUMMY(J),J=I,I+6),IDENT,ISEQ NO
      101     FORMAT(7F0.0,T75,A4,I2)
              CALL SEQ CHECK(IDENT,ISEQ NO,ICHAR,ISEQ CNT,LP1)
      11      CONTINUE
            IF(IDIM .EQ.NCONST) RETURN
      C
      C         COMPLETE FORMAT SPECIFICATION IN 'IFMAT' TO READ FINAL VALUES
      C         OF 'DUMMY' ON LAST CARD OF SERIES, THEN READ LAST CARD.
      C
      111   NO=IDIM -NCONST
            ISEQ CNT=ISEQ CNT+1
              ISEQ COPY=ISEQ CNT
            WRITE(DBF,12) NO
      12    FORMAT(1H(,I1,'F0.0,T75,A4,I2)')
            READ(CR1,IFMAT)  (DUMMY(J),J=NCONST+1,IDIM),IDENT,ISEQ NO
            CALL SEQ CHECK(IDENT,ISEQ NO,ICHAR,ISEQ CNT,LP1)
            RETURN
            END
      SUBROUTINE SEQ CHECK(IDENT,ISEQ NO,ICHAR,ISEQ CNT,LP1)
      COMMON/ERROR/ICOPY,ISEQ COPY,INPUT ERROR
C
C         COMPARE IDENTIFICATION CHARACTERS AND SEQUENCE NO.
C
      K=4
      CALL COMP(K,IDENT,1,ICHAR,1)
      IF(K.EQ.4.AND.ISEQ NO.EQ.ISEQ CNT) RETURN
      CALL PAGE(LP1)
      WRITE(LP1,11) ICHAR,ISEQ CNT,IDENT,ISEQ NO
   11 FORMAT(1H0,'DATA ERROR - PROGRAM EXPECTS THE IDENTIFICATION CHARAC
     1TERS ''',A4,I2,''' BUT HAS FOUND ''',A4,I2,'''.'/'0THE DATA CARDS
     2ARE OUT OF SEQUENCE OR HAVE BEEN GIVEN THE WRONG SEQUENCE NUMBERS.
     3')
       INPUT ERROR=1
       RETURN
       END
      SUBROUTINE CARD CHECK(IERR)
      COMMON/ERROR/ICOPY,ISEQ NO,INPUT ERROR
      INPUT ERROR=1
      IF(IERR) 101,101,0
      WRITE(0,1001) IERR
1001  FORMAT(1H0,'EXECUTION ERROR ',I3)
      RETURN
101   IF(INPUT ERROR) 102,0,102
      INPUT ERROR=1
      CALL PAGE(0)
102   K=1
      IERR=-IERR
      CALL COMP(K,IERR,4,1H*,1)
      IF(K) 12,0,12
      WRITE(0,11) ICOPY,ISEQ NO
11    FORMAT(1H0,'A NON-NUMERIC CHARACTER OTHER THAN A DECIMAL POINT HAS
     1BEEN FOUND IN COLUMNS 1-74 OF THE CARD IDENTIFIED AS ''',A4,I2,'''
     2.')
      RETURN
12    WRITE(0,121)ICOPY,ISEQ NO
121   FORMAT(1H0,'THE CARD IDENTIFIED AS ''',A4,I2,''' APPEARS TO CONTAI
     1N TOO FEW ENTRIES.')
      RETURN
      END
      SUBROUTINE PAGE(LP1)
      COMMON/APAGE/NO WEEK,NO PAGE,ADATE,ATIME
      IALLOCN WEEK=NO WEEK+2
      IF(IALLOCN WEEK.GT.52) IALLOCN WEEK=IALLOCN WEEK-52
      WRITE(LP1,11) NO WEEK,ATIME,ADATE,NO PAGE,IALLOCN WEEK
   11 FORMAT(1H1,'WEEK NO.',I3,T18,'TIME ',A8,T38,'CONSTRUCTION ',
     1'COPPER STOCK AND PRODUCTION SUMMARY',T91,'DATE ',A8,T111,
     2'PAGE NO.',I3/T40,'WITH PRODUCTION ALLOCATION FOR WEEK NO.',
     3I3/T50,'(PSALM - MAIN PROGRAM)')
      NO PAGE=NO PAGE+1
      RETURN
      END
```

```
            SUBROUTINE FREAD(PR1,IRECD)
            INTEGER PR1
            COMMON/BUFFER/NWORDS,NTEMP,NPERM,NRECD,IBUFF(1016)
            IF(IRECD.LE.NRECD) STOP FD
    11      IF(IRECD.EQ.NRECD) RETURN
            READ(PR1)  NWORDS,NTEMP,NPERM,NRECD,(IBUFF(I),I=1,NWORDS-4)
            GO TO 11
            END
```

```
      SUBROUTINE FWRITE(PR2)
      INTEGER PR2
      COMMON/BUFFER/NWORDS,NTEMP,NPERM,NRECD,IBUFF(1016)
C
C          WRITE 'NWORDS' WORDS FROM BUFFER TO PR2.
C
      WRITE(PR2)NWORDS,NTEMP,NPERM,NRECD,(IBUFF(I),I=1,NWORDS-4)
      RETURN
      END
      SUBROUTINE FCOPY(PR1,PR2,IRECD,IEND)
      INTEGER PR1,PR2
      COMMON/BUFFER/NWORDS,NTEMP,NPERM,NRECD,IBUFF(1016)
      IF(IRECD.LE.NRECD) STOP FY
11    IF(IRECD.EQ.NRECD) GO TO 12
      READ(PR1)  NWORDS,NTEMP,NPERM,NRECD,(IBUFF(I),I=1,NWORDS-4)
      GO TO 11
12    WRITE(PR2) NWORDS,NTEMP,NPERM,NRECD,(IBUFF(I),I=1,NWORDS-4)
      IF(IEND.EQ.1) ENDFILE PR2
      RETURN
      END
```

```
            SUBROUTINE DEMAND STATISTICS
            INTEGER CR1,DBF,ED1
            DIMENSION AV DEM(16,3),VAR DEM(16,3),STD DEV DEM(16,3)
            COMMON/BUFFER/NWORDS,NTEMP,NPERM,NRECD,IBUFF(1016)
            COMMON /PERM/ICODE(16,3),PTYPE(3),IPNAME(5,16,3),IFACT(4),ISITE
     1                          (3,4),CSTOC(16,3),MAXLVL(16,3),ROPROD(16,3,4),
     2                          DEM SUM(16,3),DEM SQ(16,3),PDEM(16,3),PREQ(4),
     A                          RE ORD(16,3),IPSTOP(16,3),TOT STOCK,TOT PROD,
     B                          SPROD TOT(3,4),SITE TOT(4),TOT DEM,SMPROD
            COMMON/INPUT OUTPUT/CR1,LP1,DBF,MT1,MT2,ED1
            COMMON/APAGE/NO WEEK,NO PAGE,ADATE,ATIME
            EQUIVALENCE (AV DEM(1),IBUFF(1)),(VAR DEM(1),IBUFF(97)),(STD DEV
     1                DEM(1),IBUFF(193))
C
C               CALCULATE MEAN DEMAND IN QUARTER 'AV DEM', VARIANCE 'VAR DEM'
C               STANDARD DEVIATION 'STD DEV DEM'.
C
              DO 11 I=1,3
              DO 11 J=1,16
              AV DEM(J,I)=DEM SUM(J,I)/13.0
              VAR DEM(J,I)=(DEM SQ(J,I)-13*AV DEM(J,I)*AV DEM(J,I))/12.0
              STD DEV DEM(J,I)=SQRT(VAR DEM(J,I))
11            CONTINUE
            CALL PAGE(LP1)
C
C               WRITE TABLE 8.   (DEMAND STATISTICS)
C
            WRITE(LP1,12)
12          FORMAT( //5X,'TABLE 8',T53,'DEMAND STATISTICS'/5X,7(1H-),
           1T53,17(1H-)////T17,'PRODUCT SIZE',T37,'MEAN WEEKLY DEMAND',T64,
           2'VARIANCE',T81,'STANDARD DEVIATION'/T19,'AND TYPE',T42,'(TONNES)',
           3T86,'(TONNES)'/)
            WRITE(LP1,13) (((IPNAME(L,J,I),L=1,5),AV DEM(J,I),VAR DEM(J,I),
     1                          STD DEV DEM(J,I),J=1,16),I=1,3)
13          FORMAT(T14,4A4,A2,T44,F5.1,T62,F8.1,T87,F5.1)
            RETURN
            END
```

```
      SUBROUTINE GOODS OVERLAY
      CALL GOODS1
      CALL GOODS2
      CALL GOODS3
      RETURN
      END
```

```
       SUBROUTINE GOODS1
       INTEGER REFPT(10),COMMA,PT(10)
      0DIMENSION DES(5,5),FUNC(100,5),IFUNC(100,5),FDES(5,5),HORMK(15),IH
      1ORMK(16),VRTMK(15),IVRTMK(16),LINE(106),NOENTRY(5),JX(5),DATAR(3,1
      200,5),IDATA(3,100,5),DATAI(100),JMKR(5),NK(5),JA(5)
       COMMON/GRAPH/LINE,IDATA,NOHORMK,IHORMK,HORMK,NOVRTMK,IVRTMK,VRTMK,
      1DATAR,FUNC,IFUNC,    NOENTRY
       COMMON/EXTRA/II,NXMIN,NOLINES,IMD,IVCHEK,PT,DATAL,FDES,DES,XMIN,
      1XMAX,YMIN,YMAX,JMKR,NK,JA,JX,IFSET,X AXIS(4),Y AXIS(4),STD FORMAT
      2(11)
       COMMON/CONTROL/NO,NF,IPDS,IOF,ISETSCL,ISELPT,LORD,LABS,ISUPRZ,LOGX
      1,LOGY,LNX,LNY,IFMAT,      ILINE,IXSCALE
      0DATA REFPT/'*   ','+   ','.   ','#   ','X   ','A   ','B   ','C   '
      1,'D   ','E   '/,COMMA/',   '/
       DATA LPAREN/'(   '/,IRPAREN/')   '/
       IF(ISELPT.EQ.1) CALL FMOVE(REFPT,PT,5)
       IVRTMK(1),IVRTMK(2)=0
       CALL FMOVE(IVRTMK(1),IVRTMK(3),7)
       NXMIN,NYMIN=0
       IVCHEK=1
    12 IFSET=0
       IF((IPDS-NO).EQ.1) IFSET=1
       IF(ILINE-1000) 121,121,0
       WRITE(3,123)
123    FORMAT(' VALUE OF "ILINE" EXCEEDS 1000 - PROGRAM HALTED')
       STOP
121    IF(LOGX-1) 90,0,90
       DO 91 J=1,NO+IFSET
       DO 91 I=1,NOENTRY(J)
    91 DATAR(1,I,J)=ALOG10(DATAR(1,I,J))
    90 IF(LOGY-1) 92,0,92
       DO 93 J=1,NO
       DO 93 I=1,NOENTRY(J)
    93 DATAR(2,I,J)=ALOG10(DATAR(2,I,J))
    92 IF(LNX-1) 94,0,94
       DO 95 J=1,NO+IFSET
       DO 95 I=1,NOENTRY(J)
    95 DATAR(1,I,J)=ALOG(DATAR(1,I,J))
    94 IF(LNY-1) 96,0,96
       DO 97 J=1,NO
       DO 97 I=1,NOENTRY(J)
    97 DATAR(2,I,J)=ALOG(DATAR(2,I,J))
```

```
C
C         PRINT DATA SERIES
C
96    CONTINUE
      IF(ISETSCL-1) 0,27,0
C
C         CALCULATE MAX AND MIN VALUES OF ORDINATES
C
      YMAX,YMIN=DATAR(2,1,1)
      DO 25 J=1,NO
      DO 25 I=1,NOENTRY(J)
      IF(DATAR(2,I,J)-YMAX) 26,26,0
      YMAX=DATAR(2,I,J)
      GO TO 25
   26 IF(DATAR(2,I,J)-YMIN) 0,25,25
      YMIN=DATAR(2,I,J)
   25 CONTINUE
C
C         CALCULATE MIN. AND MAX. VALUES OF FUNCTIONS
C
      IF(NF) 106,106,0
      DO 21 I=1,NOENTRY(IPDS)
C     FUNC(I,1)=FUNC1(J1,J2,---JN)
C     FUNC(I,2)=FUNC2(J1,J2,---JN)
C               DITTO
C               DITTO
C     FUNC(I,5)=FUNC5(J1,J2,---JN)
   21 CONTINUE
      FMAX,FMIN=FUNC(1,1)
      DO 101 J=1,NF
      DO 101 I=1,NOENTRY(IPDS)
      IF(FMIN-FUNC(I,J)) 102,102,0
      FMIN=FUNC(I,J)
      GO TO 101
  102 IF(FMAX-FUNC(I,J))  0,101,101
      FMAX=FUNC(I,J)
  101 CONTINUE
      IF(YMIN-FMIN) 103,103,0
      YMIN=FMIN
  103 IF(YMAX-FMAX)  0,106,106
      YMAX=FMAX
C
```

```
C         CALCULATE MAX AND MIN VALUES OF ABSCISSAE. IT IS ASSUMED THESE
C                   ARE IN INCREASING ORDER OF MAGNITUDE
C
106   IF(ILINF-1) 0,27,0
      XMAX=DATAP(1,NOENTRY(1),1)
      XMIN=DATAR(1,1,1)
      DO 27 J=1,NO+IFSET
      IF(XMAX-DATAR(1,NOENTRY(J),J)) 0,28,28
      XMAX=DATAP(1,NOENTRY(J),J)
   28 IF(XMIN-DATAR(1,1,J))27,27,0
      XMIN=DATAP(1,1,J)
   27 CONTINUF
      IF(ISUPR7) 24,0,24
      IF(IFMAT-1) 171,0,171
      DO 20 J=1,NO
   20 WRITE(3,17) (DES(I,J),I=1,5),DATAR(1,1,J),DATAR(2,1,J),(COMMA,DATA
     1R(1,I,J),DATAR(2,I,J),I=2,NOENTRY(J))
   17 FORMAT(////1X,5A8/41(1H-)//(/2H (,E12.5,',',E12.5,1H),3(A1,1H(,E12
     1.5,',',F12.5,1H)),A1))
      GO TO 172
171   CALL DEFBUF(2,80,STD FORMAT)
      CALL DYN FORMAT(XMAX,XMIN,IFIELD X)
      CALL DYN FORMAT(YMAX,YMIN,IFIELD Y)
      IFIELD=IFIELD X+IFIELD Y+4
      NREPEAT=120/IFIELD-1
      NSPACF=(120-120/IFIELD*IFIELD)/2+1
      WRITE(2,175) NSPACE,IFIELD X,IFIELD Y,NREPEAT,IFIELD X,IFIELD Y
175   FORMAT('((/',I2,'X,A1,F',I2,'.1,A1,F',I2,'.1,A1,',I2,'(2A1,F',I2,'
     1.1,A1,F',I2,'.1,A1),A1))')
      DO 173 J=1,NO
      WRITE(3,174) (DES(I,J),I=1,5)
174   FORMAT(////1X,5A8/41(1H-)//)
      WRITE(3,STD FORMAT) LPAREN,DATAR(1,1,J),COMMA,DATAR(2,1,J),IRPAREN
     1,(COMMA,LPAREN,DATAR(1,I,J),COMMA,DATAR(2,I,J),IRPAREN,I=2,
     2NOENTRY(J))
173   CONTINUE
172   IF(IPDS) 24,24,0
      DO 72 J=1,NOENTRY(IPDS)
   72 DATAL(J)=DATAR(1,J,IPDS)
24    RETURN
      END
      SUBROUTINE GOODS2
      INTEGER PT(10)
      DIMENSION FUNC(100,5),IFUNC(100,5),HORMK(15),IHORMK(16),VRTMK(15),
     1IVRTMK(16),LINE(106),NOENTRY(5),JMKR(5),JA(5),DATAR(3,100,5),IDATA
     2(3,100,5),DATAL(100),NK(5),JX(5),FDES(5,5),DES(5,5)
      COMMON/GRAPH/LINE,IDATA,NOHORMK,IHORMK,HORMK,NOVRTMK,IVRTMK,VRTMK,
     1DATAR,FUNC,IFUNC,    NOENTRY
      COMMON/EXTRA/II,NXMIN,NOLINES,IMD,IVCHEK,PT,DATAL,FDES,DES,XMIN,
     1XMAX,YMIN,YMAX,JMKR,NK,JA,JX,IFSET,X AXIS(4),Y AXIS(4),STD FORMAT
     2(11)
      COMMON/CONTROL/NO,NF,IPDS,IOF,ISETSCL,ISELPT,LORD,LABS,ISUPRZ,LOGX
     1,LOGY,LNX,LNY,IFMAT,       ILINE,IXSCALE
      DATA IE/4HI   /,MINUS/4H-   /,COMMA/',   '/,LPAREN/'(   '/,
     1IRPAREN/')   '/
      AXMIN=0.0
      AYMIN=0.0
      IF(YMIN) 0,76,76
      YLOG=10**IFIX(ALOG10(-YMIN))
      AYMIN=IFIX(-YMIN/YLOG)*YLOG
      IF(-YMIN-AYMIN.GT.0.0) AYMIN=AYMIN+YLOG
      YMIN=0.0
      IF(AYMIN.EQ.0.0) GO TO 76
      DO 79 J=1,NO+IFSET
      DO 79 I=1,NOENTRY(J)
   79 DATAR(2,I,J)=DATAR(2,I,J)+AYMIN
      YMAX=YMAX+AYMIN
C
C         AMEND HORMK FOR NEGATIVE DATA
C
      LORD=IE
      NOHORMK=NOHORMK+1
      HORMK(NOHORMK)=0.0
      DO 83 I=1,NOHORMK
   83 HORMK(I)=HORMK(I)+AYMIN
   76 IF(XMIN) 0,77,77
      XLOG=10**IFIX(ALOG10(-XMIN))
      AXMIN=IFIX(-XMIN/XLOG)*XLOG
      IF(-XMIN-AXMIN.GT.0.0) AXMIN=AXMIN+XLOG
      XMIN=0.0
      IF(AXMIN.EQ.0.0) GO TO 77
      DO 80 J=1,NO+IFSET
      DO 80 I=1,NOENTRY(J)
   80 DATAR(1,I,J)=DATAR(1,I,J)+AXMIN
      XMAX=XMAX+AXMIN
C
C           AMEND VRTMK FOR NEGATIVE DATA
C
      LABS=MINUS
      IF(NOVRTMK) 85,85,0
      DO 84 I=1,NOVRTMK
   84 VRTMK(I)=VRTMK(I)+AXMIN
      DO 86 I=1,NOVRTMK
      IF(VRTMK(I)-AXMIN) 86,77,0
      DO 87 J=1,NOVRTMK-I+1
   87 VRTMK(NOVRTMK+2-J)=VRTMK(NOVRTMK+1-J)
      VRTMK(I)=AXMIN
      GO TO 88
   86 CONTINUE
   88 NOVRTMK=NOVRTMK+1
      GO TO 77
   85 VRTMK(1)=AXMIN
      NOVRTMK=1
      IVCHEK=1
   77 L=1
      CALL EXPONENT(XMAX,IEXPX,L,X,IFSET)
      L=2
      IA=0
      CALL EXPONENT(YMAX,IEXPY,L,Y,IA)
      IXMIN=NINT(XMIN/X)
      IXMAX=NINT(XMAX/X)
      NXMIN=NINT(AXMIN/X)
      IYMIN=NINT(YMIN/Y)
      IYMAX=NINT(YMAX/Y)
C
C          PRINT FUNCTION CO-ORDINATES IF ANY, THEN SCALE
C
      IF(NF) 234,234,0
      IF(ISUPPZ.EQ.1) GO TO 232
      IF(IFMAT-1) 231,0,231
      DO 23 J=1,NF
   23 WRITE(3,17)(FDES(I,J),I=1,5),DATAL(1),FUNC(1,J),(COMMA,DATAL(I),FU
     1NC(I,J),I=2,NOFNTRY(IPDS))
   17 FORMAT(////1X,5A8/41(1H-)//(/2H (,E12.5,',',E12.5,1H),3(A1,1H(,E12
     1.5,',',E12.5,1H)),A1))
```

```
      GO TO 232
231   DO 233 J=1,NF
233   WRITE(3,171)(FDES(I,J),I=1,5)
171   FORMAT(////1X,5A8/41(1H-)//)
      WRITE(3,STD FORMAT) LPAREN,DATAL(1),COMMA,FUNC(1,J),IRPAREN,
     1(COMMA,LPAREN,DATAL(I),COMMA,FUNC(I,J),IRPAREN,I=2,NOENTRY(IPDS))
232   DO 18 J=1,NF
      DO 18 I=1,NOENTRY(IPDS)
      FUNC(I,J)=(FUNC(I,J)+AYMIN)/Y
   18 IFUNC(I,J)=NINT(FUNC(I,J))
C
C          PRINT TITLE FOR GRAPH AND OUTPUT SCALE FACTORS (X10EXP)
C                      ALSO  CALCULATE AND PRINT SCALE
C
234   IF(ISUPRZ.EQ.0) CALL PAGE(LP1)
      WRITE(3,73)
   73 FORMAT(//54X,'SYMBOL KEY'/54X,10(1H-)/)
      IF(IOF-1) 0,69,0
      WRITE(3,70) (PT(J),(DES(I,J),I=1,5),J=1,NO)
   70 FORMAT(/41X,A1,' = ',5A8)
      IF(NF) 58,58,69
   69 WRITE(3,70) (PT(J+5),(FDES(I,J),I=1,5),J=1,NF)
58    WRITE(3,172) X AXIS,Y AXIS
172   FORMAT(//13X,'''X'' AXIS - ',4A8,T80,'''Y'' AXIS - ',4A8//)
      WRITE(3,32) IEXPX,IEXPY
   32 FORMAT(   6X,'(ABSCISSA VALUES X 10EXP',I2,')',T85,'(ORDINATE VALU
     1ES X 10EXP',I2,')'//)
      INFO=2
      IF(AYMIN) 89,89,0
      NYMIN=NINT(AYMIN/Y)
   89 CALL SCALF(IYMIN,IYMAX,II,IV,INFO,IA,NYMIN)
      WRITE(3,33) (LINE(I),I=1,11)
   33 FORMAT(I6,10I10/5X,'*',20('    *'))
      IMD=1
      CALL SCALF(IXMIN,IXMAX,II,IV,IMD,IFSET,IA)
C
C          CHECK EFFECT OF SCALING IF 'GROUPING' HAS OCCURRED. GROUPING
C          CAUSES NOENTRY(J) TO BE REDUCED IN VALUE TO JX(J)
C
      J1=1
      DO 31 J=1,NO+IFSET
      JX(J)=0
      DO 29 I=2,NOENTRY(J)
      IF(IDATA(1,I,J)-IDATA(1,I-1,J)) 0,0,30
      J1=J1+1
      GO TO 29
   30 JX(J)=JX(J)+1
      IDATA(3,JX(J),J)=J1
      J1=1
      IDATA(1,JX(J),J)=IDATA(1,I-1,J)
   29 CONTINUE
      JX(J)=JX(J)+1
      IDATA(3,JX(J),J)=J1
      J1=1
      IDATA(1,JX(J),J)=IDATA(1,NOENTRY(J),J)
   31 CONTINUE
      RETURN
      END
      SUBROUTINE GOODS3
      INTEGER PT(10),AST,EMM,BLANK
      DIMENSION FUNC(100,5),IFUNC(100,5),HORMK(15),IHORMK(16),VRTMK(15),
     1IVRTMK(16),LINE(106),NOENTRY(5),JMKP(5),JA(5),DATAR(3,100,5),IDATA
     2(3,100,5),DATAL(100),NK(5),JX(5),FDES(5,5),DES(5,5)
      COMMON/GRAPH/LINE,IDATA,NOHORMK,IHORMK,HURMK,NOVRTMK,IVRTMK,VRTMK,
     1DATAR,FUNC,IFUNC,    NOENTRY
      COMMON/EXTRA/II,NXMIN,NOLINES,IMD,IVCHEK,PT,DATAL,FDES,DES,XMIN,
     1XMAX,YMIN,YMAX,JMKR,NK,JA,JX,IFSET,X AXIS(4),Y AXIS(4),STD FORMAT
     2(11)
      COMMON/CONTROL/NO,NF,IPDS,IOF,ISETSCL,ISELPT,LORD,LABS,ISUPRZ,LOGX
     1,LOGY,LNX,LNY,IFMAT,      ILINE,IXSCALE
      DATA BLANK/'    '/,EMM/'M   '/,IE/'I   '/,MINUS/'-   '/,AST/'*   '
     1/
C
C          OUTPUT SECTION
C
      IF(IXSCALE.NE.0) NXMIN=-IXSCALE
      JQ=II-NXMIN
      WRITE(3,34) JQ
   34 FORMAT(I5,104(1H*))
   36 FORMAT(1H+,109X,1H(,I3,1H,,I3,1H),'=*')
   35 LINE(1),LINE(2)=BLANK
      CALL FMOVE(LINE(1),LINE(3),52)
      DO 37 J=1,5
      JA(J)=0
   37 JMKR(J)=1
C
C          ABSCISSAE CONTROL LOOP
C
      JQL=-NXMIN
      NLINE=102
      IF(ILINE.EQ.1) NLINE=NOLINES
      DO 44 JQQ=1,NLINE
      LK=0
      JQ=JQQ+II-1
      IF(MOD(JQ,10)) 19,0,19
      IF(JQQ-1) 0,19,0
      JQL=JQL+IMD
   19 NU=1
      IF(IOF-1) 0,40,0
C
```

```
C           DATA SERIES OUTPUT LOOP
C
      DO 39 J=1,NO
      NK(J)=0
     0IF((IDATA(1,JMKR(J),J).NE.JQ).OR.(JQ.GT.IDATA(1,JX(J),J))) GO TO 3
     19
      NK(J)=1
      LINE(1)=AST
      K,JA(J)=JA(J)+1
      JV=IDATA(2,K,J)+1
      LINE(JV)=PT(J)
      IF(MOD(JQ,10)) 41,0,41
      IF(JQ-IT) 42,0,42
      NU=2
      WRITE(3,47) LINE
   47 FORMAT(1H+,4X,106A1)
      LINE(JV)=BLANK
      IF(JV-1) 48,0,48
     0IF((J.EQ.1).AND.(ISELPT.EQ.0)) WRITE(3,36) IDATA(1,JMKR(1),1),
     1IDATA(2,JMKR(1),1)
      GO TO 48
   42 CALL SETFORMAT(NU,JV,JQL,BLANK)
      IF(JV-1) 48,0,48
      IX=IDATA(1,JMKR(1),1)-NXMIN
      IF((J.EQ.1).AND.(ISELPT.EQ.0)) WRITE(3,36) IX,IDATA(2,JMKR(1),1)
      GO TO 48
   41 CALL SELTWO (NU,JV,BLANK)
      IF(JV-1) 48,0,48
      IX=IDATA(1,JMKR(1),1)-NXMIN
      IF((J.EQ.1).AND.(ISELPT.EQ.0)) WRITE(3,36) IX,IDATA(2,JMKR(1),1)
   48 IL=IDATA(3,JMKR(J),J)
      JMKR(J)=JMKR(J)+1
      IF(IL-1) 39,39,0
      K,JA(J)=JA(J)-1
      SUM=0.0
      DO 49 IO=1,IL
      K=K+1
      IK=IDATA(2,K,J)
      SUM=SUM+IK
      IK=IK+1
      LINE(IK)=PT(J)
      WRITE(3,47) LINE
   49 LINE(IK)=BLANK
      JA(J)=K
      IAV=NINT(SUM/IL+1)
      LINE(IAV)=EMM
      WRITE(3,47) LINE
      LINE(IAV)=BLANK
   39 CONTINUE
C
C          FUNCTION OUTPUT LOOP
C
   40 IF(NF) 50,50,0
      IF(JQ-IDATA(1,JX(IPDS),IPDS))0,0,50
      IF(IPDS-NO+1) 0,64,0
      IF(NK(IPDS)) 62,64,62
   64 IF(JQ-IDATA(1,JMKR(IPDS),IPDS))50,63,50
   62 IF(JQ-IDATA(1,JMKR(IPDS)-1,IPDS)) 50,0,50
      JMKR(IPDS)=JMKR(IPDS)-1
      LK=1
   63 DO 51 IN=1,NF
      IY=IFUNC(JMKR(IPDS),IN)+1
      LINE(IY)=PT(IN+5)
      IF(IOF-1) 52,0,52
      LINE(1)=AST
      LINE(IY)=PT(IN+5)
      IF(MOD(JQ,10)) 53,0,53
      IF(JQ-II) 54,0,54
      WRITE(3,47) LINE
      LINE(1),LINE(IY)=BLANK
      GO TO 51
   54 CALL SELFORMAT(NU,IY,JQL,BLANK)
      GO TO 51
   53 CALL SELTWO(NU,IY,BLANK)
      GO TO 51
   52 WRITE(3,47) LINE
      LINE(IY)=BLANK
   51 CONTINUE
     0IF((IOF.EQ.1).OR.(IPDS.EQ.NO+1).OR.(LK.EQ.1)) JMKR(IPDS)=JMKR(IPDS
     1)+1
   50 IF(NU-2) 0,55,0
      IF(JQ.EQ.II) GO TO 55
      IF(MOD(JQ,10)) 56,0,56
      IF(NOHORMK) 68,68,0
      DO 71 I=1,NOHORMK
   71 LINE(IHORMK(I)+1)=LORD
      LINE(1)=AST
      WRITE(3,74) JQL,LINE
   74 FORMAT(I5,106A1)
      GO TO 59
   68 WRITE(3,57) JQL
   57 FORMAT(I5,1H*)
      GO TO 59
   56 WRITE(3,45)
   45 FORMAT('        *')
C
C         CHECK FOR HORMK AND VRTMK
C
   55 IF(NOHORMK) 59,59,0
      DO 60 I=1,NOHORMK
   60 LINE(IHORMK(I)+1)=LORD
      WRITE(3,47) LINE
      LINE(1),LINE(2)=BLANK
      CALL FMOVE(LINE(1),LINE(3),52)
   59 IF(NOVRTMK) 44,44,0
      IF(JQ-IVRTMK(IVCHEK)) 44,0,44
      IVCHEK=IVCHEK+1
      LINE(1),LINE(2)=LABS
      CALL FMOVE(LINE(1),LINE(3),51)
      WRITE(3,47) LINE
      LINE(1),LINE(2)=BLANK
      CALL FMOVE(LINE(1),LINE(3),51)
   44 CONTINUE
      RETURN
      END
      SUBROUTINE SCALE(MINVL,MAXVL,II,IV,IMD,IFSET,NEG)
     0DIMENSION LINE(106),IDATA(3,100,5),IHORMK(16),HORMK(15),
     1IVRTMK(16),VRTMK(15),DATAR(3,100,5),NOENTRY(5),FUNC(100,5),IFUNC(1
     200,5)
      COMMON/GRAPH/LINE,IDATA,NOHORMK,IHORMK,HORMK,NOVRTMK,IVRTMK,VRTMK,
     1DATAR,FUNC,IFUNC,    NOENTRY
      IWRN=0
      IDIFF=MAXVL-MINVL
   17 IF(IDIFF-50) 0,0,1
      IF(IDIFF-20) 0,0,2
      IWRN=1
      II=MINVL/10*10
      IV=II+20
      IF(MAXVL-IV) 0,0,2
      AMULT=5.0
      CALL CHANJSCL(IFSET,IDIFF,IMD,&17,AMULT,II)
      IF(IMD-2) 0,4,0
      IMD=2
      IF(NOVRTMK) 12,12,0
      DO 10 I=1,NOVRTMK
   10 IVRTMK(I)=NINT((VRTMK(I)-II)*5.0)
      IF(NOVRTMK-1)12,12,0
      CALL IVRTRANK
   12 RETURN
    4 IMD=5
      IF(NOHORMK) 13,13,0
      DO 14 J=1,NOHORMK
   14 IHORMK(J)=NINT((HORMK(J)-II)*5.0)
   13 J=0
      DO 3 I=1,11
      LINE(I)=II+J-NEG
    3 J=J+2
      RETURN
    2 IF(MAXVL-50) 0,0,5
      IWRN=1
      II=0
      IV=50
      AMULT=2.0
      CALL CHANJSCL(IFSET,IDIFF,IMD,&17,AMULT,II)
      IF(IMD-2) 0,6,0
      IMD=5
      IF(NOVRTMK) 18,18,0
```

```
21 DO 19 I=1,NOVRTMK
19 IVRTMK(I)=NINT((VRTMK(I)-II)*2.0)
   IF(NOVRTMK-1)18,18,0
   CALL IVRTRANK
18 RETURN
 6 J=0
   DO 11 I=1,11
   LINE(I)=II+J-NEG
11 J=J+5
   IMD=2
   IF(NOHORMK) 15,15,0
   DO 16 J=1,NOHORMK
16 IHORMK(J)=NINT((HORMK(J)-II)*2.0)
15 RETURN
 5 IF(MINVL-50) 1,1,0
   IWRN=1
   II=50
   IV=100
   AMULT=2.0
   CALL CHANJSCL(IFSET,IDIFF,IMD,&17,AMULT,II)
   IF(IMD-2) 0,6,0
   IMD=5
   IF(NOVRTMK) 0,0,21
   RETURN
 1 II=0
   IV=100
   IF(IWRN-1) 7,0,7
   AMULT=1.0
   CALL CHANJSCL(IFSET,IDIFF,IMD,&17,AMULT,II)
 7 IF(IMD-2) 0,8,0
   IMD=10
   IF(IWRN-1) 22,0,22
   IF(NOVRTMK) 22,22,0
   DO 23 I=1,NOVRTMK
23 IVRTMK(I)=NINT(VRTMK(I))
   IF(NOVRTMK-1) 22,22,0
   CALL IVRTRANK
22 RETURN
 8 J=0
   DO 9 I=1,11
   LINE(I)=II+J-NEG
 9 J=J+10
```

```
      RETURN
      END
      SUBROUTINE SELFORMAT(NU,JV,JQ,BLANK)
      INTEGER BLANK
      DIMENSION LINE(106)
      COMMON /GRAPH/LINE
    1 FORMAT(1H+,4X,106A1)
    2 FORMAT(I5,106A1)
    3 FORMAT(5X,106A1)
      GO TO(0,4),NU
      NU=2
      WRITE(3,2) JQ,LINE
      LINE(1),LINE(JV)=BLANK
      RETURN
    4 LINE(1)=BLANK
      WRITE(3,1) LINE
      LINE(JV)=BLANK
      RETURN
      ENTRY SELTWO(NU,JV,BLANK)
      GO TO(0,5),NU
      NU=2
      WRITE(3,3) LINE
      LINE(1),LINE(JV)=BLANK
      RETURN
    5 LINE(1)=BLANK
      WRITE(3,1) LINE
      LINE(JV)=BLANK
      RETURN
      END
```

```
       SUBROUTINE EXPONENT(TMAX,IEXPN,L,X,IFSET)
      0DIMENSION LINE(106),IDATA(3,100,5),DATAR(3,100,5),FUNC(100,5
      1),IFUNC(100,5),VRTMK(15),IVRTMK(16),HORMK(15),IHORMK(16),NOFNTRY(5
      2)
       COMMON/GRAPH/LINE,IDATA,NOHORMK,IHORMK,HORMK,NOVRTMK,IVRTMK,VRTMK,
      1DATAR,FUNC,IFUNC,    NOENTRY
       COMMON/CONTROL/NO,NF,IPDS,IOF,ISETSCL,ISELPT,LORD,LABS,ISUPRZ,LOGX
      1,LOGY,LNX,LNY,IFMAT,      ILINE,IXSCALE
       EXPN=ALOG10(TMAX)
       IFXPN=EXPN
       IF(EXPN-IFXPN) 0,0,1
       IFXPN=IFXPN-2
       GO TO 5
     1 IFXPN=IFXPN-1
     5 X=10.0**IFXPN
       DO 6 J=1,NO+IFSET
       DO 6 I=1,NOFNTRY(J)
       DATAR(L,I,J)=DATAR(L,I,J)/X
     6 IDATA(L,I,J)=NINT(DATAR(L,I,J))
       IF(L-1) 7,0,7
       IF(NOVRTMK) 13,13,0
       DO 9 I=1,NOVRTMK
       VRTMK(I)=VRTMK(I)/X
     9 IVRTMK(I)=NINT(VRTMK(I))
       IF(NOVRTMK-1) 13,13,0
       CALL IVRTRANK
       GO TO 13
     7 IF(NOHORMK) 13,13,0
       DO 10 I=1,NOHORMK
       HORMK(I)=HORMK(I)/X
    10 IHORMK(I)=NINT(HORMK(I))
    13 RETURN
       END
      SUBROUTINE CHANJSCL(IFSET,IDIFF,L,*,AMULT,II)
      DIMENSION LINE(106),IDATA(3,100,5),IHORMK(16),HORMK(15),IVRTMK(16)
     1,VRTMK(15),DATAR(3,100,5),FUNC(100,5),IFUNC(100,5),NOENTRY(5)
      COMMON/GRAPH/LINE,IDATA,NOHORMK,IHORMK,HORMK,NOVRTMK,IVRTMK,VRTMK,
     1DATAR,FUNC,IFUNC,    NOENTRY
      COMMON/CONTROL/NO,NF,IPDS,IOF,ISETSCL,ISELPT,LORD,LABS,ISUPRZ,LOGX
     1,LOGY,LNX,LNY,IFMAT,      ILINE,IXSCALE
      DO 1 J=1,NO+IFSET
      DO 1 I=1,NOENTRY(J)
      IDATA(L,I,J)=NINT((DATAR(L,I,J)-II)*AMULT)
      IF(IDATA(L,I,J)) 0,1,1
      IDIFF=IDIFF+1
      RETURN 1
    1 CONTINUE
      IF(L-2) 2,0,2
      IF(NF) 2,2,0
      DO 3 J=1,NF
      DO 3 I=1,NOENTRY(IPDS)
      IFUNC(I,J)=NINT((FUNC(I,J)-II)*AMULT)
      IF(IFUNC(I,J)) 0,3,3
      IDIFF=IDIFF+1
      RETURN 1
    3 CONTINUE
    2 RETURN
      END
      SUBROUTINE IVRTRANK
     0DIMENSION LINE(106),IDATA(3,100,5),IHORMK(16),HORMK(15),IVRTMK(16)
     1,VRTMK(15)
      COMMON/GRAPH/LINE,IDATA,NOHORMK,IHORMK,HORMK,NOVRTMK,IVRTMK,VRTMK
      K=0
      DO 1 I=2,NOVRTMK
      IF(IVRTMK(I)-IVRTMK(I-1)) 1,1,0
      K=K+1
      IVRTMK(K)=IVRTMK(I-1)
    1 CONTINUE
      IF(IVRTMK(NOVRTMK)-IVRTMK(NOVRTMK-1)) 2,2,0
      K=K+1
      IVRTMK(K)=IVRTMK(NOVRTMK)
    2 RETURN
      END
      SUBROUTINE DYN FORMAT(TMAX,TMIN,IFIELD)
      IF(TMAX) 1,0,1
      TMAX=3.0
      IFIELD=3
      GO TO 2
1     TMAX LOG=ALOG10(ABS(TMAX))
      IFIELD=TMAX LOG+3
2     IF(TMIN.GE.0.0) RETURN
      TMIN LOG=ALOG10(-TMIN)
      IF(TMAX) 3,3,0
      IF(TMIN LOG+1.GT.TMAX LOG) IFIELD=TMIN LOG+4
      RETURN
3     IFIELD=TMIN LOG+4
      RETURN
      END
```

APPENDIX P2

Program Description: PSALM File Editor, User Version (SLM2) - see Appendices M1 & M2 for details of operation and organisation.

```
      PROGRAM(SLM2)
      INPUT 1=CR0
      INPUT 2=MT0/UNFORMATTED(UNKNOWNASYET)/1024
      OUTPUT 3=LP0
      CREATE 4=MT1/UNFORMATTED(UNKNOWNASYET)/1024
      USE 5=/ARRAY
      COMPRESS INTEGER AND LOGICAL
      TRACE 2
      END
      MASTER EDIT
      INTEGER CRO,ARR
      DIMENSION TEMP(10)
      COMMON/APAGE/NO PAGE,ADATE,ATIME,LPO,IGEN NO
      EXTERNAL PARAMETER ERROR
      CRO=1
      MTO=2
      LPO=3
      MT1=4
      ARR=5
      CALL DEFBUF(5,80,TEMP)
      NO PAGE=0
C
C          READ GENERATION NO. (IE. WEEK NO.) OF FILE TO AMENDED.
C
      READ(CRO,13) TEMP
      K=5
      CALL COMP(K,TEMP(1),1,5H*WEEK,1)
      IF(K.EQ.5)GO TO 11
      CALL PAGE
      WRITE(LPO,131) TEMP
      WRITE(LPO,14)
      STOP 'ABANDON RUN - PARAMETER CARD ERROR'
11    READ(ARR,111) IGEN NO
111   FORMAT(5X,I0)
C
C          SPECIFY FILE NAMES AND CORRES. CHANNEL NOS.
C
      CALL FILE(MTO,12HSLM1DATAFILE,IGEN NO,15)
      CALL FILE(MT1,12HSLM1DATAFILE,IGEN NO,15)
      NRECD=0
      CALL DATE(ADATE)
      CALL TIME(ATIME)
C
C          START OF INPUT LOOP. DATA ON PARAMETER CARD IS READ INTO
C          ARRAY 'TEMP'. (LOOP IS TERMINATED BY E.O.F. MARKER '****')
C
12    CALL FTRAP(PARAMETER ERROR)
      READ(CRO,13,END=19) TEMP
13    FORMAT(10A8)
C
C          START NEW PAGE FOR NEXT DIRECTIVE.
```

```
C
          CALL PAGE
          WRITE(LPO,131) TEMP
    131   FORMAT(///6X,'DIRECTIVE:- ',10A8/6X,9(1H-))
C
C              IDENTIFY THE DIRECTIVE AS 'UPDATE' OR 'COPY'.
C
          K=7
          CALL COMP(K,TEMP(1),1,7H*UPDATE,1)
          IF(K.EQ.7) GO TO 15
          K=5
          CALL COMP(K,TEMP(1),1,5H*COPY,1)
          IF(K.EQ.5) GO TO 17
    132   WRITE(LPO,14)
    14    FORMAT(1H0,'ERROR - DIRECTIVE NOT RECOGNISED.')
          STOP 'ABANDON RUN - PARAMETER CARD ERROR'
    15    READ(ARR,16) IRECD,IADD,IFILE,AMT,IFMAT
    16    FORMAT(7X,3I0,F0.0,I0)
          CALL FRESET
          CALL UPDATE(MT0,MT1,CR0,IRECD,IADD,IFILE,AMT,IFMAT)
          GO TO 12
    17    READ(ARR,18) IRECD
    18    FORMAT(5X,I0)
          CALL FRESET
          CALL FCOPY(MT0,MT1,IRECD,0)
          GO TO 12
    19    ENDFILE MT1
          STOP
          END
```

```
      SUBROUTINE UPDATE(PR1,PR2,PR3,IRECD,IADD,IFILE,AMT,IFMAT)
      INTEGER PR1,PR2,PR3
      DIMENSION ITEMP(10),TEMP(10),BUFF(508),FMT(5)
      DIMENSION  TYPE(2)
      DIMENSION FMAT REAL(2)
      COMMON/BUFFER/NWORDS,NTEMP,NPERM,NRECD,IBUFF(1016)
      DATA  TYPE(1)/'CREATES '/,TYPE(2)/'CHANGES '/
      DATA INT/4HINT /
      DATA FMAT REAL(1)/'(508F0.0)       '/,FMAT INT/'(1016I0)'/
      EQUIVALENCE (IBUFF(1),BUFF(1)),(ITEMP(1),TEMP(1))
C
C         IFILE = 0        READ FROM PR1 AND WRITE TO PR2.
C
C               = 1        CREATE FILE ON PR2. (IE. PR1 NOT USED, OUTPUT
C                                                              TO PR2)
C
C               = 2        READ FROM PR1 BUT NO OUTPUT TO PR2.
C
C               = 3        AMENDMENTS TO EXISTING CONTENTS OF  /BUFFER/.
C                          PR1 AND PR2 NOT USED.
C
C
C         READ REQUIRED RECORD FROM PR1.
C
       IF(IFILE.EQ.1.OR.IFILE.EQ.3) GO TO 121
      IF(IRECD.GT.NRECD)  GO TO 11
      WRITE(LP,101)
101   FORMAT(1H0,'ERROR - A RECORD HAS BEEN REQUESTED IN THE ABOVE DIREC
     1TIVE HAVING A NUMBER LESS THAN THAT LAST ACCESSED.'/' RECORDS CAN
     2ONLY BE ACCESSED IN ASCENDING ORDER OF RECORD NUMBER.')
      STOP 'ABANDON RUN - PARAMETER CARD ERROR'
11     IF(IRECD.EQ.NRECD) GO TO 121
      READ(PR1) NWORDS,NTEMP,NPERM,NRECD,(IBUFF(I),I=1,NWORDS-4)
       GO TO 11
121   LP=0
      K=1
      IF(IFILE-1) 1211,0,1211
      READ(PR3,13) IRECD
      GO TO 1212
1211  K=2
1212  WRITE(LP,122)  TYPE(K),IRECD,PR1,PR3,PR2
122   FORMAT(                 ///16X,'THIS RUN ',A7,' RECORD NO.',
     1I4,//16X,'INPUT CHANNEL NO(S) =',I3,1H,,I2//16X,'OUTPUT ',
     2'CHANNEL NO.   =',I3)
      IF(IADD.EQ.2) WRITE(LP,123) AMT
123   FORMAT(/16X,'THE CONSTANT AMOUNT',F7.1,' IS ADDED TO EXISTING',
     1' VALUES.')
      READ(PR3,13) NREP
      WRITE(LP,124) NREP
124   FORMAT(/16X,'NO. OF AMENDMENTS    =',I3)
        DO 23 I=1,NREP
C
C          READ NO. OF ENTRIES 'NENTRY' IN CURRENT REPLACEMENT, START
C          ADDRESS IN 'BUFFER'('ISTART')AND ENTRY TYPE 'ITYPE'.
C          READ FORMAT OF CURRENT REPLACEMENT.
C
        READ(PR3,13) NENTRY,ISTART,ITYPE
13      FORMAT(2I0,A4)
        IF(ISTART.EQ.0) GO TO 1301
        GO TO 1303
1301    READ(PR3,1302) NWORDS,NTEMP,NPERM,NRECD
1302    FORMAT(4I0)
        WRITE(LP,13021) I,NWORDS,NTEMP,NPERM,NRECD
13021   FORMAT(////6X,'AMENDMENT NO.',2X,1H(,I2,1H)/6X,13(1H-)///6X,
     1  'NWORDS =',I5,3X,'NTEMP =',I5,3X,'NPERM =',I5,3X,'NRECD =',I3)
        GO TO 23
1303  IF(IFMAT.EQ.0) GO TO 1305
        READ(PR3,1304) FMT
1304    FORMAT(5A8)
1305  WRITE(LP,131) I
131     FORMAT(//// 6X,'AMENDMENT NO.',2X,1H(,I2,1H)/ 6X,13(1H-)//1H ,4(
     1  5X,'ADDRESS', 5X,'CONTENTS')//)
        NCONST=NENTRY/10*10
C
C          SELECT CORRECT READ STATEMENT ACCORDING TO 'ITYPE'.
C
        IF(INT-ITYPE) 15,0,15
        IF(IFMAT.EQ.0) CALL COPY8(FMT(1),FMAT INT)
        IF(IADD-1) 0,132,136
        READ(PR3,FMT) (IBUFF(J),J=ISTART,ISTART+NENTRY-1)
        WRITE(LP,1341) (K,IBUFF(K),K=ISTART,ISTART+NENTRY-1)
        GO TO 23
132     IF(NENTRY-10) 13411,13411,0
          DO 134 J=ISTART,ISTART+NENTRY-10,10
```

```
         READ(PR3,FMT) (ITEMP(K),K=1,10)
          DO 1331 K=J,J+9
          IBUFF(K)=IBUFF(K)+ITEMP(K-J+1)
1331      CONTINUE
134      CONTINUE
        WRITE(LP,1341) (K,IBUFF(K),K=ISTART,ISTART+NCONST-1)
1341    FORMAT((7X,I4,7X,I7,3(7X,I4,7X,I7)/))
        IF(NENTRY.EQ.NCONST) GO TO 23
13411   READ(PR3,FMT) (ITEMP(J),J=1,NENTRY-NCONST)
         DO 135 J=ISTART+NCONST,ISTART+NENTRY-1
         IBUFF(J)=IBUFF(J)+ITEMP(J-ISTART-NCONST+1)
135      CONTINUE
        IF(NENTRY.LE.10) GO TO 1353
        WRITE(LP,1352) (K,IBUFF(K),K=ISTART+NCONST,ISTART+NENTRY-1)
1352    FORMAT(1H+,49X,2(7X,I4,7X,I7)//(7X,I4,7X,I7,3(7X,I4,7X,I7)/))
        GO TO 23
1353    WRITE(LP,1341) (K,IBUFF(K),K=ISTART,ISTART+NENTRY-1)
        GO TO 23
136      DO 137 J=ISTART,ISTART+NENTRY-1
         IBUFF(J)=IBUFF(J)+AMT
137      CONTINUE
        WRITE(LP,1341) (K,IBUFF(K),K=ISTART,ISTART+NENTRY-1)
        GO TO 23
15     IF(IFMAT.NE.0) GO TO 151
        CALL COPY8(FMT(1),FMAT REAL(1))
        CALL COPY8(FMT(2),FMAT REAL(2))
151     IPOSN=ISTART
        ISTART=ISTART/2+1
        IF(IADD-1) 0,17,21
        READ(PR3,FMT)(BUFF(J),J=ISTART,ISTART+NENTRY-1)
        WRITE(LP,16)  (K,BUFF(K/2+1),K=IPOSN,IPOSN+2*NENTRY-2,2)
16      FORMAT((7X,I4,7X,F7.1,3(7X,I4,7X,F7.1)/))
        GO TO 23
17      IF(NENTRY-10) 191,191,0
         DO  19 J=ISTART,ISTART+NENTRY-10,10
         READ(PR3,FMT) (TEMP(K),K=1,10)
          DO 181 K=J,J+9
          BUFF(K)=BUFF(K)+TEMP(K-J+1)
181       CONTINUE
19       CONTINUE
        WRITE(LP,16)  (K,BUFF(K/2+1),K=IPOSN,IPOSN+2*NCONST-2,2)
        IF(NENTRY.EQ.NCONST) GO TO 23
```

```
191         READ(PR3,FMT)(TEMP(K),K=1,NENTRY-NCONST)
              DO 20 J=ISTART+NCONST,ISTART+NENTRY-1
              BUFF(J)=BUFF(J)+TEMP(J-ISTART-NCONST+1)
20            CONTINUE
            IF(NENTRY.LE.10) GO TO 202
            WRITE(LP,201) (K,BUFF(K/2+1),K=IPOSN+2*NCONST,IPOSN+2*(NENTRY-
          1                                                         NCONST)-2,2)
201         FORMAT(1H+,49X,2(7X,I4,7X,F7.1)//(7X,I4,7X,F7.1,3(7X,I4,7X,F7.1)
          1 /))
            GO TO 23
202         WRITE(LP,16)  (K,BUFF(K/2+1),K=IPOSN,IPOSN+2*NENTRY-2,2)
            GO TO 23
21            DO 22 J=ISTART,ISTART+NENTRY-1
              BUFF(J)=BUFF(J)+AMT
22            CONTINUE
            WRITE(LP,16)  (K,BUFF(K/2+1),K=IPOSN,IPOSN+2*NENTRY-2,2)
23          CONTINUE
           IF(IFILE.GT.1) RETURN
           WRITE(PR2)NWORDS,NTEMP,NPERM,NRECD,(IBUFF(I),I=1,NWORDS-4)
           RETURN
           END
```

```
      SUBROUTINE FCOPY(PR1,PR2,IRECD,IEND)
      INTEGER PR1,PR2
      COMMON/BUFFER/NWORDS,NTEMP,NPERM,NRECD,IBUFF(1016)
      IF(IRECD.GT.NRECD)  GO TO 11
      WRITE(LP,101)
  101 FORMAT(1H0,'ERROR - A RECORD HAS BEEN REQUESTED IN THE ABOVE DIREC
     1TIVE HAVING A NUMBER LESS THAN THAT LAST ACCESSED.'/' RECORDS CAN
     2ONLY BE ACCESSED IN ASCENDING ORDER OF RECORD NUMBER.')
      STOP 'ABANDON RUN - PARAMETER CARD ERROR'
   11 IF(IRECD.EQ.NRECD) GO TO 12
      READ(PR1)  NWORDS,NTEMP,NPERM,NRECD,(IBUFF(I),I=1,NWORDS-4)
      GO TO 11
   12 WRITE(PR2) NWORDS,NTEMP,NPERM,NRECD,(IBUFF(I),I=1,NWORDS-4)
      IF(IEND.EQ.1) ENDFILE PR2
      RETURN
      END
      SUBROUTINE PARAMETER ERROR(IERR)
      COMMON/APAGE/NO PAGE,ADATE,ATIME,LPO,NO WEEK
      DATA KOPY/4H'  ' /
      IF(IERR.GT.0) GO TO 14
      IERR=-IERR
      K=1
      CALL COMP(K,IERR,1,1H*,1)
      IF(K.EQ.1) GO TO 12
      CALL COPY(1,KOPY,2,IERR,4)
      WRITE(LPO,11) KOPY
   11 FORMAT(1H0,'ERROR - THE NON-NUMERIC CHARACTER ',A3,' (CONTAINED WI
     1THIN THE APOSTROPHES) HAS BEEN FOUND WITHIN THE PARAMETER FIELD OF
     2'/9X,'THE ABOVE CARD.')
  111 STOP 'ABANDON RUN - PARAMETER CARD ERROR'
   12 WRITE(LPO,13)
   13 FORMAT(1H0,'ERROR - ONE OR MORE PARAMETERS ARE MISSING FROM THE AB
     1OVE CARD.')
      GO TO 111
   14 RETURN
      END
      SUBROUTINE PAGE
      COMMON/APAGE/NO PAGE,ADATE,ATIME,LPO,NO WEEK
      NO PAGE=NO PAGE+1
      WRITE(LPO,11) NO WEEK,ATIME,ADATE,NO PAGE
11    FORMAT(1H1,'WEEK NO.',I3,T18,'TIME ',A8,T51,'PSALM - FILE EDITOR',
     1T91,'DATE ',A8,T111,'PAGE NO.',I3/T57,'(SLM2)')
      RETURN
      END
      FINISH
```

APPENDIX P3

Program Description: PSALM File Editor (SLM3) -
see Appendix M2 for details
of operation and organisation.

```
PROGRAM(SLM3)
INPUT 1=CR0
INPUT 2=MT0/UNFORMATTED(UNKNOWNASYET)/1024
OUTPUT 3=LP0
CREATE 4=MT1/UNFORMATTED(UNKNOWNASYET)/1024
USE 5=/ARRAY
USE 6=ED0/UNFORMATTED/1024
COMPRESS INTEGER AND LOGICAL
TRACE 2
END
```

```
            MASTER FILE EDITOR
            INTEGER CRO,ARR,EDO
            DIMENSION IDUMMY(310),BUFF  (508),SITE BUDGET(4),INBUDG(3,4)
           1          ,PBUDG(16,3,4),PBUDGET(4)
            COMMON/APAGE/NO PAGE,ADATE,ATIME,NO WEEK
            COMMON/INPUT OUTPUT/CRO,ARR,EDO
            COMMON/ERROR/ICOPY,ISEQ NO,INPUT ERROR
            COMMON/PERM/ICODE(16,3),PTYPE(3),IPNAME(5,16,3),IFACT(4)
           1          ,ISITE(3,4)
            COMMON/BUFFER/TEMP(10),NWORDS,NTEMP,NPERM,NRECD,IBUFF(1016)
            DATA INBUDG(1)/'*BLX*BLZ*BLY*BKX*BKZ*BKY*BBX*BBZ*BBY*BAX*BAZ*BAY'/
            EQUIVALENCE(ICODE(1),IDUMMY(1)),(IBUFF(835),IBASE WK),(IBUFF(1),
           1          BUFF(1)),(IBUFF(836),IHOR WK),(IBUFF(1),PBUDG(1)),
           2         (IBUFF(145),WKBUDG),(IBUFF(209),PBUDGET(1))
            EXTERNAL PARAMETER ERROR,CARD CHECK
      C
      C        THIS IS THE PROGRAM DESCRIBED IN SECTION 5 OF THE 'PSALM' USER
      C        MANUAL.
      C
      C                                        ----------
            CRO=1
            MTO=2
            LPO=3
            MT1=4
            ARR=5
            EDO=6
            NO PAGE=0
            NRECD=0
            CALL DATE(ADATE)
            CALL TIME(ATIME)
      C
      C
      C          NAME 'TEMP' AS A DUMMY PERIPHERAL USING 'DEFBUF' AND INSTATE
      C          OWN ERROR TRAP 'PARAMETER ERROR'.
      C
            CALL DEFBUF(ARR,80,TEMP)
            CALL FTRAP(PARAMETER ERROR)
      C
      C          READ FIRST DIRECTIVE '*WEEK' AND CHECK ITS IDENTITY.
      C
            READ(CRO,11) TEMP
      11    FORMAT(10A8)
      CALL IDENTIFY(IDIRECTIVE)
      IF(IDIRECTIVE.EQ.1) GO TO 111
      CALL PAGE
      WRITE(LPO,122) TEMP
      WRITE(LPO,27)
      STOP
C
C         READ PARAMETER - 'NO WEEK'.
C
111   READ(ARR,12) NO WEEK
12    FORMAT(5X,I0)
      CALL PAGE
      WRITE(LPO,122) TEMP
C
C         NAME INPUT AND OUTPUT MAG. TAPE FILES.
C
      CALL FILE(MT0,12HSLM1DATAFILE,NO WEEK,15)
      CALL FILE(MT1,12HSLM1DATAFILE,NO WEEK,15)
C
C         READ RECORD 10 FROM INPUT FILE AND COPY ALPHAMERIC CONTENTS
C         INTO /PERM/ COMMON BLOCK USING DUMMY ARRAY 'IDUMMY'. WRITE
C         RECORD 10 TO NEW FILE, AND COPY RECORD 20 TO BACKING FILE.
C
      CALL FREAD(MT0,10)
      CALL FMOVE(IBUFF,IDUMMY,155)
      CALL FWRITE(MT1)
      CALL FCOPY(MT0,ED0,20,0)
C
C         WITHIN THE PROGRAM THE FOLLOWING DIRECTIVES HAVE THE CODES
C         SHOWN.
C
C              DIRECTIVE                     CODE
C              ---------                     ----
C
C              *WEEK                          1
C              *ADJUSTMENTS                   2
C              *FORECASTS                     3
C              *LIST ADJUSTMENTS              4
C              *LIST FORECASTS                5
C              *BUDGETS                       6
C              *END                           7
C
```

```
C           THE FIRST FOUR DIRECTIVES ABOVE CAN OCCUR IN ANY ORDER WITH
C           ANY FREQUENCY. TO DEAL WITH THIS A LOOP HAS BEEN CREATED
C           WHICH IS LEFT ONLY WHEN THE '*BUDGETS' OR '*END' DIRECTIVE
C           IS FOUND.
C
      LOOP MARKER=0
C
C           READ NEXT DIRECTIVE AND IDENTIFY.
C
121   READ(CRO,11) TEMP
      CALL PAGE
      WRITE(LPO,122) TEMP
122   FORMAT(///6X,'DIRECTIVE:- ',10A8/6X,9(1H=))
      CALL IDENTIFY(IDIRECTIVE)
      IF(IDIRECTIVE.GT.1) GO TO 123
      WRITE(LPO,27)
      STOP
C
C           IF 'IDIRECTIVE'.LT.6 READ RECORD 30 INTO CORE AND SET
C           'LOOP MARKER' TO 1.
C
123   IF(IDIRECTIVE.GE.6) GO TO 22
      IF(LOOP MARKER.EQ.0) CALL FREAD(MTO,30)
      LOOP MARKER=1
      IF(IDIRECTIVE-3) 0,17,20
C
C           DIRECTIVE HAS BEEN IDENTIFIED AS '*ADJUSTMENTS'.
C           READ PARAMETERS
C
      READ(ARR,13) ISTART,NADJUST
13    FORMAT(12X,2I0)
C
C           CALCULATE START ADDRESS IN'IBUFF' AND READ ADJUSTMENTS; THEN
C           LIST ADJUSTMENTS.
C
      IPOSN=313+ISTART-IBASE WK
      READ (CRO,14) (BUFF(J),J=IPOSN,NADJUST+IPOSN-1)
14    FORMAT(508F0.0)
C
C           LIST CURRENT ADJUSTMENTS.
C
      WRITE(LPO,16) (J, BUFF(J-ISTART+IPOSN),J=ISTART,ISTART+
     1                                                         NADJUST-1)
16    FORMAT(///1H ,7X,4(6X,'WEEK',5X,'ADJUSTMENT')/1H ,7X,4(5X,'NUMBER
     1',13X)///(1H ,7X,4(6X,I3,8X,F5.1,3X)/))
C
C         RETURN TO START OF LOOP.
C
      GO TO 121
C
C         DIRECTIVE HAS BEEN IDENTIFIED AS '*FORECASTS'.
C         READ PARAMETERS.
C
17    READ(ARR,18) ISTART,NFCST
18    FORMAT(10X,2I0)
C
C         IF 'ISTART' AND 'NFCST' ARE BOTH ZERO, READ 'IBASE WK' AND
C         'IHOR WK', AND RETURN TO START OF LOOP.
C
      IF(ISTART.EQ.0.AND.NFCST.EQ.0) GO TO 181
      GO TO 183
181   READ(CRD,182) IBASE WK,IHOR WK
182   FORMAT(2I0)
      WRITE(LPO,1821) IBASE WK,IHOR WK
1821  FORMAT(///18X,'BASE WEEK =',I4,', HORIZON PERIOD =',I4,'WKS.')
      GO TO 121
183   CONTINUE
C
C         CALCULATE START ADDRESSS IN 'IBUFF' AND READ FORECASTS. THEN
C         LIST FORECASTS.
C
      IPOSN=ISTART-IBASE WK+1
      READ(CRD,14) (BUFF(J),J=IPOSN,IPOSN+NFCST-1)
C
C         LIST ABOVE CHANGES TO FORECASTS.
C
      WRITE(LPO,19) (J, BUFF(J-ISTART+IPOSN),J=ISTART,ISTART+NFCST-1)
19    FORMAT(///1H ,7X,4(6X,'WEEK',6X,'FORECAST',1X)/1H ,7X,4(5X,'NUMBER
     1',14X)///(1H ,7X,4(6X,I3,8X,F5.1,3X)/))
C
C         RETURN TO START OF LOOP.
C
      GO TO 121
20    IF(IDIRECTIVE-5) 0,21,22
```

```
C
C         DIRECTIVE IDENTIFIED AS '*LIST ADJUSTMENTS'.
C
      WRITE(LP0,16) (J,BUFF(J-IBASE WK+313),J=IBASE WK,IBASE WK+IHOR
     1                                                           WK-1)
      GO TO 121
C
C         DIRECTIVE IDENTIFIED AS '*LIST FORECASTS'
C
21    WRITE(LP0,19) (J,BUFF(J-IBASE WK+1),J=IBASE WK,IBASE WK+IHOR WK-1)
      GO TO 121
C
C         IF 'LOOP MARKER'.NE.ZERO WRITE RECORD 30 TO BACKING FILE,
C         OTHERWISE COPY RECORD 30 TO BACKING FILE.
C
22    IF(LOOP MARKER) 23,0,23
      CALL FCOPY(MT0,ED0,30,0)
      GO TO 231
23    CALL FWRITE(ED0)
231   IF(IDIRECTIVE-7) 232,0,0
C
C         DIRECTIVE IDENTIFIED AS '*END'. COPY REMAINDER OF INPUT
C         FILE TO MT1.
C
      ENDFILE ED0
      REWIND ED0
      NRECD=10
      CALL FCOPY(ED0,MT1,20,0)
      CALL FCOPY(ED0,MT1,30,0)
      CALL FCOPY(MT0,MT1,40,0)
      CALL FCOPY(MT0,MT1,50,0)
      CALL FCOPY(MT0,MT1,60,1)
      STOP
C
C         DIRECTIVE IDENTIFIED AS '*BUDGETS'.
C         INPUT RECORD 40 TO /BUFFER/ AND READ NEW BUDGET FROM CARDS
C         USING SUBROUTINE 'INPUT', FOR FURTHER DETAILS SEE 'PSALM
C         USER MANUAL'.
C
232   CALL FTRAP(CARD CHECK)
      CALL FREAD(MT0,40)
      INPUT ERROR=0
```

```
        DO 24 I=1,4
        DO 24 J=1,3
        PBUDG(16,J,I)=0.0
        CALL INPUT(PBUDG(1,J,I),15,INBUDG(J,I),CRO,ARR,LPO)
24      CONTINUE
      CALL DEFBUF(ARR,60,TEMP)
C
C         IF INPUT CONTAINS ERRORS, THE OLD BUDGETS ARE COPIED TO THE
C         BACKING FILE AND THE NEXT DIRECTIVE IS READ.
C
      IF(INPUT ERROR.EQ.0)GO TO 25
      BACKSPACE MTO
      CALL FCOPY(MTO,EDO,40,1)
      GO TO 262
C
C         CALCULATE TOTAL WEEKLY BUDGET 'TOT BUDGET' AND WEEKLY SITE
C         BUDGETS 'SITE BUDGET'.
C
25      TOT BUDGET=0.0
        DO 26 K=1,4
        SITE BUDGET(K)=0.0
          DO 26 I=1,3
          DO 26 J=1,16
          SITE BUDGET(K)=SITE BUDGET(K)+PBUDG(J,I,K)
          TOT BUDGET=TOT BUDGET+PBUDG(J,I,K)
26        CONTINUE
C
C         LIST NEW PRODUCTION  BUDGETS.
C
        DO 261 K=1,4
        CALL PAGE
        WRITE(LPO,2601) K,IFACT(K),(ISITE(L,K),L=1,3)
2601    FORMAT(///5X,1H(,I1,') PRODUCTION BUDGET FOR FACTORY NO. ',I2,
     1  ' - ',3A4/  5X,40(1H-)///T37,'CODE NO.',T56,'PRODUCT SIZE',T78,
     2  'BUDGET (TONNES)'/T58,'AND TYPE'/)
        WRITE(LPO,2602)((ICODE(J,I),(IPNAME(L,J,I),L=1,5),PBUDG(J,I,K),
     1                    J=1,16),I=1,3)
2602    FORMAT(37X,I5,T53,4A4,A2,T80,F5.1)
261     CONTINUE
C
C         1) WRITE RECORD 40 TO BACKING FILE.
C         2) REWIND BACKING FILE AND RE-SET 'NRECD' TO 10.
```

```
C          3) READ RECORD 20, INSERT 'WKBUDG' AND WRITE TO MT1.
C          4) COPY RECORDS 30 AND 40 TO MT1.
C          5) COPY RECORD 50 FROM MTO TO MT1.
C          6) READ RECORD 60, INSERT   =PBUDG'   AND WRITE TO MT1.
C
      CALL FWRITE(EDO)
  262 ENDFILE EDO
      REWIND EDO
      NRECD=10
      CALL FREAD(EDO,20)
      IF(INPUT ERROR.EQ.0) WKBUDG=TOT BUDGET
      CALL FWRITE(MT1)
      CALL FCOPY(EDO,MT1,30,0)
      CALL FCOPY(EDO,MT1,40,0)
      CALL FCOPY(MTO,MT1,50,0)
      CALL FREAD(MTO,60)
      IF(INPUT ERROR.EQ.0) CALL FMOVE(SITE BUDGET,PBUDGET,4)
      CALL FWRITE(MT1)
      ENDFILE MT1
C
C          READ NEXT DIRECTIVE WHICH SHOULD BE '*END'.
C
      CALL PAGE
      READ(CRO,11) TEMP
      WRITE(LPO,122) TEMP
      CALL  IDENTIFY(IDIRECTIVE)
      IF(IDIRECTIVE.NE.7) WRITE(LPO,27)
   27 FORMAT(//6X,'ERROR - ABOVE DIRECTIVE IS OUT OF CONTEXT. (SEE ''PSA
     1LM'' USER MANUAL SECTION 5.3')
      STOP
      END
      SUBROUTINE IDENTIFY(IDIRECTIVE)
      INTEGER CRO,ARR,EDO
      COMMON/INPUT OUTPUT/CRO,ARR,EDO
      COMMON/BUFFER/TEMP(10),NWORDS,NTEMP,NPERM,NRECD,IBUFF(1016)
      K=5
      CALL COMP(K,TEMP(1),1,5H*WEEK,1)
      IF(K-5)11,0,11
      IDIRECTIVE=1
      RETURN
   11 CALL COMP8(TEMP(1),8H*ADJUSTM,K)
      IF(K-1) 12,0,12
      IDIRECTIVE=2
      RETURN
   12 CALL COMP8(TEMP(1),8H*FORECAS,K)
      IF(K-1) 13,0,13
      IDIRECTIVE=3
      RETURN
   13 CALL COMP8(TEMP(1),8H*LIST AD,K)
      IF(K-1) 14,0,14
      IDIRECTIVE=4
      RETURN
   14 CALL COMP8(TEMP(1),8H*LIST FO,K)
      IF(K-1) 15,0,15
      IDIRECTIVE=5
      RETURN
   15 CALL COMP8(TEMP(1),8H*BUDGETS,K)
      IF(K-1) 16,0,16
      IDIRECTIVE=6
      RETURN
   16 K=4
      CALL COMP(K,TEMP(1),1,4H*END,1)
      IF(K-4)17,0,17
      IDIRECTIVE=7
      RETURN
   17 WRITE(LPO,18)
   18 FORMAT(//  ' ERROR - ABOVE DIRECTIVE NOT RECOGNISED.')
      STOP
      END
```

```
              SUBROUTINE INPUT(DUMMY,IDIM,ICHAR,CR1,DBF,LP1)
              INTEGER CR1,DBF
              DIMENSION DUMMY(IDIM),IFMAT(5)
              COMMON/ERROR/ICOPY,ISEQ COPY,INPUT ERROR
              CALL DEFBUF(DBF,20,IFMAT)
              ICOPY=ICHAR
              ISEQ CNT=0
              NCONST=IDIM/7*7
              IF(NCONST.EQ.0) GO TO 111
                DO 11 I=1,1+IDIM-7,7
                ISEQ CNT=ISEQ CNT+1
                ISEQ COPY=ISEQ CNT
                READ(CR1,101) (DUMMY(J),J=I,I+6),IDENT,ISEQ NO
        101     FORMAT(7F0.0,T75,A4,I2)
                CALL SEQ CHECK(IDENT,ISEQ NO,ICHAR,ISEQ CNT,LP1)
        11      CONTINUE
              IF(IDIM .EQ.NCONST) RETURN
        C
        C         COMPLETE FORMAT SPECIFICATION IN 'IFMAT' TO READ FINAL VALUES
        C         OF 'DUMMY' ON LAST CARD OF SERIES, THEN READ LAST CARD.
        C
        111   NO=IDIM -NCONST
              ISEQ CNT=ISEQ CNT+1
                ISEQ COPY=ISEQ CNT
              WRITE(DBF,12) NO
        12    FORMAT(1H(,I1,'F0.0,T75,A4,I2)')
              READ(CR1,IFMAT)  (DUMMY(J),J=NCONST+1,IDIM),IDENT,ISEQ NO
              CALL SEQ CHECK(IDENT,ISEQ NO,ICHAR,ISEQ CNT,LP1)
              RETURN
              END
```

```
      SUBROUTINE SEQ CHECK(IDENT,ISEQ NO,ICHAR,ISEQ CNT,LP1)
      COMMON/ERROR/ICOPY,ISEQ COPY,INPUT ERROR
C
C          COMPARE IDENTIFICATION CHARACTERS AND SEQUENCE NO.
C
      K=4
      CALL COMP(K,IDENT,1,ICHAR,1)
      IF(K.EQ.4.AND.ISEQ NO.EQ.ISEQ CNT) RETURN
      CALL PAGE(LP1)
      WRITE(LP1,11) ICHAR,ISEQ CNT,IDENT,ISEQ NO
11    FORMAT(1H0,'DATA ERROR - PROGRAM EXPECTS THE IDENTIFICATION CHARAC
     1TERS ''',A4,I2,''' BUT HAS FOUND ''',A4,I2,'''.'/'0THE DATA CARDS
     2ARE OUT OF SEQUENCE OR HAVE BEEN GIVEN THE WRONG SEQUENCE NUMBERS.
     3')
      INPUT ERROR=1
      RETURN
      END
      SUBROUTINE CARD CHECK(IERR)
      COMMON/ERROR/ICOPY,ISEQ NO,INPUT ERROR
      IF(IERR) 101,101,0
      WRITE(0,1001) IERR
 1001 FORMAT(1H0,'EXECUTION ERROR ',I3)
      RETURN
  101 IF(INPUT ERROR) 102,0,102
      INPUT ERROR=1
      CALL PAGE(0)
  102 K=1
      IERR=-IERR
      CALL COMP(K,IERR,4,1H*,1)
      CALL COMP(K,ICOPY,1,1H*,1)
      IF(K) 12,0,12
      WRITE(0,11) ICOPY,ISEQ NO
   11  FORMAT(1H0,'A NON-NUMERIC CHARACTER OTHER THAN A DECIMAL POINT HAS
     1BEEN FOUND IN COLUMNS 1-74 OF THE CARD IDENTIFIED AS ''',A4,I2,'''
     2.')
       RETURN
   12  WRITE(0,121)ICOPY,ISEQ NO
  121  FORMAT(1H0,'THE CARD IDENTIFIED AS ''',A4,I2,''' APPEARS TO CONTAI
     1N TOO FEW ENTRIES.')
       RETURN
       END
      SUBROUTINE FREAD(PR1,IRECD)
      INTEGER PR1
      COMMON/BUFFER/TEMP(10),NWORDS,NTEMP,NPERM,NRECD,IBUFF(1016)
      IF(IRECD.LE.NRECD) STOP FD
11    IF(IRECD.EQ.NRECD) RETURN
      READ(PR1)  NWORDS,NTEMP,NPERM,NRECD,(IBUFF(I),I=1,NWORDS-4)
      GO TO 11
      END
      SUBROUTINE FWRITE(PR2)
      INTEGER PR2
      COMMON/BUFFER/TEMP(10),NWORDS,NTEMP,NPERM,NRECD,IBUFF(1016)
C
C         WRITE 'NWORDS' WORDS FROM BUFFER TO PR2.
C
      WRITE(PR2)NWORDS,NTEMP,NPERM,NRECD,(IBUFF(I),I=1,NWORDS-4)
      RETURN
      END
      SUBROUTINE FCOPY(PR1,PR2,IRECD,IEND)
      INTEGER PR1,PR2
      COMMON/BUFFER/TEMP(10),NWORDS,NTEMP,NPERM,NRECD,IBUFF(1016)
      IF(IRECD.LE.NRECD) STOP FY
11    IF(IRECD.EQ.NRECD) GO TO 12
      READ(PR1)  NWORDS,NTEMP,NPERM,NRECD,(IBUFF(I),I=1,NWORDS-4)
      GO TO 11
12    WRITE(PR2) NWORDS,NTEMP,NPERM,NRECD,(IBUFF(I),I=1,NWORDS-4)
      IF(IEND.EQ.1) ENDFILE PR2
      RETURN
      END
      SUBROUTINE PARAMETER ERROR(IERR)
      COMMON/APAGE/NO PAGE,ADATE,ATIME,LPO,NO WEEK
      DATA KOPY/4H' ' /
      IF(IERR) 101,101,0
      WRITE(LPO,1001) IERR
1001  FORMAT(' EXECUTION ERROR',I4)
      RETURN
101   IERR=-IERR
      K=1
      CALL COMP(K,IERR,1,1H*,1)
      IF(K.EQ.1) GO TO 12
      CALL COPY(1,KOPY,2,IERR,4)
      WRITE(LPO,11) KOPY
11    FORMAT(1H0,'ERROR - THE NON-NUMERIC CHARACTER ',A3,' (CONTAINED WI
     1THIN THE APOSTROPHES) HAS BEEN FOUND WITHIN THE PARAMETER FIELD OF
     2'/9X,'THE ABOVE CARD.')
111   STOP 'ABANDON RUN - PARAMETER CARD ERROR'
12    WRITE(LPO,13)
13    FORMAT(1H0,'ERROR - ONE OR MORE PARAMETERS ARE MISSING FROM THE AB
     1OVE CARD.')
      GO TO 111
      END
      SUBROUTINF PAGE
      COMMON/APAGF/NO PAGE,ADATE,ATIME,NO WEEK
      COMMON/INPUT OUTPUT/CRO,ARR,EDO
      NO PAGF=NO PAGE+1
      WRITE(LPO,11) NO WEEK,ATIME,ADATE,NO PAGE
   11 FORMAT(1H1,'WEEK NO.',I3,T18,'TIME ',A8,T51,'PSALM - FILE EDITOR',
     1T91,'DATE ',A8,T111,'PAGE NO.',I3/T57,'(SLM3)')
      RETURN
      END
      FINISH
```

APPENDIX P4

Program Description: Box-Jenkins (p,d,q) model - program PDQMODEL incorporating program AUTOCOR. (Minor modifications to AUTOCOR are necessary to produce a "stand alone" program giving, additionally, autocorrelations of 1st and 2nd differences of the input data series).

Notes for Users:

PDQMODEL will fit a sequence of user specified models to a data series and will compute the sum of squared errors associated with each model. If required (IAUTO = 1; if not required IAUTO = 0) the autocorrelation structure of the residuals remaining after a series has been fitted may be computed. Optionally (IMEAN = 1; if not required IMEAN = 0) the mean of the series may be subtracted from individual members of the series before model fitting commences.

Input Requirements

Sequence no.	Parameter	Format	Description
1.	P,D Q, SEAS	(4I0)	P,D & Q specify the model to be fitted and SEAS is the seasonal difference (P,Q ≯ 2)
2.	NOPAR, IAUTO, IMEAN	(4I0)	NOPAR is number of weights to be fitted in model of type specified by P,D,Q. IMEAN and IAUTO are described above.
3.	Data series - the format chosen is specific to the PSALM study and would require a simple modification for general use.		
4.	V(1),V(2),W(1),W(2)	(4F0.0)	V & W correspond to ϕ & θ respectively. (Note that DO loop input requires at least one entry for V & for W).
5.	Repeat step 4 a further (NOPAR - 1) times & then go to 6.		
6.	If IAUTO = 0 go to step 8, otherwise go to step 7.		
7.	NS, SEAS, ISUPR	(2I2,I1)	NS=1 (no. of data series), SEAS is period of seasonality to be used in differencing residual series & ISUPRZ = 1 suppresses a listing of the differenced series, otherwise ISUPRZ = 0.

Note. The statement at label 19 of sub routine AUTOCOR should be changed to read DO 100 J=1,3 in the "stand alone" program.

8. (a) To continue fitting models to residual series return to step 1 and continue as above but omitting step 3 subsequently.

(b) To terminate the run, return to step 1 and carry out steps 1 & 2 entering zero for P,D,Q, SEAS, NOPAR, IAUTO and IMEAN.

```
PROGRAM(B511)
INPUT 1=CR0
OUTPUT 2=LP0
OUTPUT 3=TP0
COMPRESS INTEGER AND LOGICAL
TRACE 2
END
```

```
          MASTER PDQMODEL
          INTEGER P,D,Q,SEAS,T
          DIMENSION DA(250),V(3),W(2),ERR(250)
          DATA DA(250),ERR(250)/2*0.0/
C
C              READ PARAMETERS P,D,Q AND SEAS IF APPLICABLE - THEN NOPAR
C
          READ(1,1) P,D,Q,SEAS,NOPAR,IAUTO,IMEAN
1         FORMAT(4I0)
          NO=1
          DO 2 L=1,3
          READ(1,3) N,(DUMMY,DA(I),I=NO,NO+N-1)
3         FORMAT(////I3/(16F0.0))
2         NO=NO+N
          NO=NO-1
C
C              IF 'D'=0 CALCULATE MEAN OF SERIES AND SUBTRACT THIS FROM
C                                      EACH ENTRY
C
          IF(IMEAN) 24,24,0
          SUM=0.0
          DO 97 I=1,NO
       97 SUM=SUM+DA(I)
          MEAN=SUM/NO
          DO 98 I=1,NO
       98 DA(I)=DA(I)-MEAN
C
C              TAKE DIFFERENCES OF EVERY 'SEAS'TH ENTRY IF SEAS.GT.0
C
24        IF(SEAS) 4,4,0
          NB=NO
          NO=NO-SEAS
          DO 5 I=1,NO
          DA(NB)=DA(NB)-DA(NB-SEAS)
        5 NB=NB-1
          DO 50 J=1,NO
50        DA(J)=DA(J+SEAS)
C
C              TAKE 'D' DIFFERENCES AS SPECIFIED BY (P,D,Q) MODEL TO BE
C                                          FITTED
C
4         IF(D) 7,7,0
```

```
            NB=NO
            DO 6 J=1,D
            NO=NO-1
            DO 6 I=1,NO
            DA(NB)=DA(NB)-DA(NB-1)
          6 NB=NB-1
            DO 51 J=1,NO
 51         DA(J)=DA(J+D)
          7 WRITE(2,20) SEAS,P,D,Q
 20         FORMAT(1H1,3X,63(1H*)/4X,1H*,61X,1H*/4X,'* PERIOD OF SEASONALITY '
           1'SEAS'' HAS BEEN INPUT AS ',I2,' INTERVALS *'/4X,1H*,61X,1H*/4X,
           *'* P',                                                            '
           2ARAMETERS OF (P,D,Q) MODEL HAVE BEEN SPECIFIED AS (',I1,1H,,I1,1H,
           3,I1,1H),2X,1H*/4X,1H*,61X,1H*/4X,63(1H*)//)
            WRITE(2,18)
 18         FORMAT(1H //13X,'SUM OF SQUARED ERRORS CORRESPONDING TO USE OF WEI
           1GHTS V(1)---V(N),W(1)---W(N) IN (P,D,Q) MODEL'/13X,94(1H-)//
           *'    .',T40,1H.,T87,
           21H.,T117,1H./'    .',T7,'NO. OF DATA AFTER DIFFERENCING',T40,1H.,T6
           30,'WEIGHTS',T87,1H.,T92,'SUM OF SQUARED ERRORS',T117,1H./3X,114(1H
           4.)/'    .',T40,1H.,T87,1H.,T117,1H./'    .',T40,1H.,T47,'V(1)',2X,1H
           5:,3X,'V(2)',3X,1H:,3X,'W(1)',3X,1H:,2X,'W(2)',T87,1H.,T117,1H./'
           6 .',T40,1H.,T87,1H.,T117,1H.)
C
C                ENTER LOOP TO BE REPEATED NOPAR TIMES
C
            DO 100 K=1,NOPAR
            DO 17 I=1,2
 17         V(I),W(I)=0.0
            READ(1,16) (V(I),I=1,P),(W(I),I=1,Q)
 16         FORMAT(4F0.0)
C
C                CALCULATE ERR(T) FOR 'T' VALUES 1 TO CURRENT VALUE OF NO
C
            DO 10 T=1,NO
            SV=DA(T)
            IF(P) 0,11,0
            DO 12 I=1,P
 12         SV=SV+V(I)*DA(IT(T-I))
 11         EW=0.0
            IF(Q) 0,13,0
            DO 14 I=1,Q
```

```
   14     EW=EW+W(I)*ERR(IT(T-I))
   13     ERR(T)=SV-EW
   10     CONTINUE
          SUM=0.0
          DO 35 T=1,NO
   35     SUM=SUM+ERR(T)*ERR(T)
          WRITE(2,19) NO,(V(I),I=1,2),(W(I),I=1,2),SUM
   19     FORMAT('    .',T19,I3,T40,1H.,T46,F5.2,T56,F5.2,T67,F5.2,T77,F5.2,T
         187,1H.,T96,F8.1,T117,1H.)
  100     CONTINUE
          WRITE(2,21)
   21     FORMAT(3X,114(1H.))
          IF(IAUTO) 25,25,0
          CALL AUTOCORRELATION(NO,ERR)
C
C              CHECK IF FURTHER MODEL IS TO BE FITTED. IF SO, COPY 'ERR' INTO
C              'DA' AND THEN SET 'ERR'=0.0
C              IF 'IMEAN.GT.0' CALCULATE MEAN OF RESIDUAL SERIES AND SUBTRACT
C              THIS FROM EACH ENTRY, THEN PROCEED.
C
       25 READ(1,1) P,D,Q,SEAS,NOPAR,IAUTO,IMEAN
          IF(NOPAR) 23,23,0
          CALL FMOVE(ERR,DA,250)
          IF(IMEAN) 69,69,0
          SUM=0.0
          DO 67 I=1,NO
       67 SUM=SUM+DA(I)
          MEAN=SUM/NO
          DO 68 I=1,NO
       68 DA(I)=DA(I)-MEAN
       69 CONTINUE
          ERR(1)=0.0
          CALL FMOVE(ERR(1),ERR(2),249)
          GO TO 24
   23     STOP
          END
```

```
      SUBROUTINE AUTOCORRELATION(NO,D)
      REAL MEAN
      INTEGER SEAS
      DIMENSION D(200),COR(21),TITLE(3,3),LIST(3)
     0DATA  TITLE(1,1)/24HORIGINAL DATA             /,TITLE(1,2)/24HFIRST
     1DIFFERENCES       /,TITLE(1,3)/24HSECOND DIFFERENCES       /,LIST(1
     1)/12H- SUPRESSED /
      READ(1,2) NS,SEAS,ISUPRZ
    2 FORMAT(2I2,I1)
      DO 100 NX=1,NS
      WRITE(2,20) SEAS
   20 FORMAT(1H1,3X,63(1H*)/4X,1H*,61X,1H*/4X,'* PERIOD OF SEASONALITY '
     1'SEAS'' HAS BEEN INPUT AS ',I2,' INTERVALS *'/4X,1H*,61X,1H*/4X,63
     2(1H*))
C
C              TAKE DIFFERENCES OF EVERY 'SEAS'TH ENTRY IF SEAS.GT.0
C
      IF(SEAS) 19,19,0
      NB=NO
      NO=NO-SEAS
      DO 21 I=1,NO
      D(NB)=D(NB)-D(NB-SEAS)
   21 NB=NB-1
      DO 22 I=1,NO
   22 D(I)=D(I+SEAS)
C
C              DIFFERENCE LOOP (J=1,3)
C
   19 DO 100 J=1,1
      IF(J-1) 5,5,0
C.
C              TAKE DIFFERENCES AFTER CALC. AUTOCORRELATIONS
C
      DO 6 L=1,NO-1
    6 D(L)=D(L+1)-D(L)
      NO=NO-1
      IF(ISUPRZ) 5,0,5
      WRITE(2,18) J,(TITLE(I,J),I=1,3),(LIST(I),I=1,3)
   18 FORMAT(1H ////6X,'TABLE ',I1,'A',T46,'LISTING OF ',3A8,3A4/6X,8(1H
     1-)///)
      GO TO 17
C
```

```
C                WRITE DATA LISTING
C
    5 NF=NO/10*10+1
      WRITE(2,7) J,(TITLE(I,J),I=1,3),(I,I=1,10)
    70FORMAT(1H ////6X,'TABLE ',I1,'A',T46,'LISTING OF ',3A8/6X,8(1H-)//
     1//16X,'DATUM NO.',2X,10I8/)
      DO 13 II=1,NO-10,10
      IV=II+9
   13 WRITE(2,14) II,IV,(D(I+II-1),I=1,10)
   14 FORMAT(/15X,I3,3H - ,I3,5X,10F8.1)
      IF(MOD(NO,10)) 0,17,0
      WRITE(2,14) NF,NO,(D(I),I=NF,NO)
C
C                CALCULATE MEAN
C
   17 SUM=0.0
      DO 8 I=1,NO
    8 SUM=SUM+D(I)
      MEAN=SUM/NO
C
C                CALC. AUTOCORRELATIONS UP TO LAG OF 20 WEEKS
C
      DO 9 LAG=1,21
      SUM=0.0
      DO 10 I=1,NO-LAG+1
   10 SUM=SUM+(D(I)-MEAN)*(D(I+LAG-1)-MFAN)
      COR(LAG)=SUM/NO
      IF(LAG-1) 9,9,0
      COR(LAG)=COP(LAG)/COR(1)
    9 CONTINUE
C
C                WRITE AUTOCORRELATIONS, MEAN AND VARIANCE
C
     0WRITE(2,11) J,(TITLE(I,J),I=1,3),(I,I=1,10)
   110FORMAT(/////6X,'TABLE ',I1,'B',T38,'SAMPLE AUTOCORRELATIONS FOR ',
     13A8/6X,8(1H-)////17X,'LAGS',6X,10I8/)
      DO 15 II=1,11,10
      IV=II+9
   15 WRITE(2,16) II,IV,(COR(I+II),I=1,10)
   16 FORMAT(/16X,I2,3H - ,I2,4X,10F8.2)
      WRITE(2,12) MEAN,COR(1)
   12 FORMAT(////6X,'MEAN = ',F6.1/6X,'VARIANCE = ',F7.2)
```

```
100 CONTINUE
    RETURN
    END
```

```
      FUNCTION IT(IX)
      IF(IX) 0,0,1
      IT=250
      RETURN
    1 IT=IX
      RETURN
      END
      FINISH
```

APPENDIX P5

Program Description: Program TRIG - used to compute the errors associated with adaptive smoothing using Trigg's tracking signal as the exponential smoothing coefficient. Also the function used in conjunction with GOODS (see Appendix M3) to compute the adaptive forecasts for 1970.

```
      PROGRAM(TRIG)
      INPUT 1=CR0
      OUTPUT 2=LP0
      COMPRESS INTEGER AND LOGICAL
      TRACE 2
      END
C   PROGRAM DESCRIPTION
C   ------- -----------
C
C         AFTER INITIALISATION IN THE FIRST PERIOD, A FORECAST IS MADE
C     IN A GIVEN PERIOD USING EXPONENTIAL SMOOTHING WITH ALPHA SET TO
C     THE VALUE OF TRIGG'S TRACKING SIGNAL AS CALCULATED FOR THE PREV-
C     IOUS PERIOD. THE VALUE OF THE T.S. IN THE GIVEN PERIOD IS CALCU-
C     LATED FOR USE IN THE NEXT.
C
```

```
              MASTER
              DIMENSION IHIST(22),ACT(200),LINE(20),PROB(22)
              DATA FREQ/8H FREQU. /,APROB/8H PROB.  /
        C
        C   INPUT - READ THREE YEARS' DATA.
        C   -----
              NOACT=1
                DO 101 I=1,3
                READ(1,1001) N
        1001    FORMAT(I3)
                READ(1,1002) (DUMMY,ACT(I),I=NOACT,NOACT+N-1)
        1002    FORMAT(16F0.0)
        101     NOACT=NOACT+N
              NOACT=NOACT-1
        C
        C   INITIALISATION PROCEDURE
        C   -------------- ---------
        C
        C     VARIABLE       PERIOD    VALUE
        C     --------       ------    -----
        C     OLDSMER          1       SET TO VALUE GIVING TRACKING SIGNAL OF 0.4
        C     OLDMAD           1       SET TO SQUARE ROOT OF AVERAGE SQUARED
        C                              ERROR = 32.5 FOR ALPHA=0.4 AND ORDINARY
        C                              EXPONENTIAL SMOOTHING.
        C     OLDFCST          1       CALC. FOR PERIOD 2 USING ALPHA=0.4 AND
        C     (IE.ANUFCST              OLDFCST SET TO AVERAGE VALUE OF SERIES
        C     IN PERIOD 1)
        C     OLDTS            1       SET TO 0.4
        C     ERROR            2       CALC. FROM OLDFCST AND ACT.
        C     ANUSMER          2       CALC. FROM OLDSMER AND ERROR
        C     ANUMAD           2       CALC. FROM OLDMAD AND ABS(F
        C     NUTS             2       CALC. FROM ANUSMER AND AN'
        C     ANUFCST          2       CALC. FROM OLDFCST AND A'
        C
        C   PERIOD ONE
        C   ------ ---
        C
              OLD MAD=32.5
              OLDSMER=0.4*OLDMAD
              OLDFCST=120.0
              NUFCST=0.4*ACT(1)+0.6*OLDFCST
              OLDTS=0.4
```

```
      IHIST(1),IHIST(2)=0
      CALL FMOVE(IHIST(1),IHIST(3),10)
C
C   SUBSEQUENT PERIODS
C   ---------- -------
C
      SUMSQ=0.0
        DO 13 I=2,NOACT
        ERR=ACT(I)-OLDFCST
        SUMSQ=SUMSQ+ERR*ERR
        ANUSMER=0.15*ERR+0.85*OLDSMER
        ANUMAD=0.15*(ABS(ERR))+0.85*OLDMAD
        ANUTS=ANUSMER/ANUMAD
        AOLDTS=ABS(OLDTS)
        ANUFCST=AOLDTS  *ACT(I)+(1-AOLDTS)*OLDFCST
C
C       CHANGE IDENTITY OF VARIABLES FOR USE NEXT PERIOD.
C
        OLDFCST=ANUFCST
        OLDTS=ANUTS
        OLDMAD=ANUMAD
        OLDSMER=ANUSMER
C
C       SORT ERROR INTO CLASS INTERVAL TO BUILD UP DISTRIBUTION OF ERROR
C
        INTVL=((ERR+100.0)/10.0)+2.0
        IF(INTVL-2) 0,11,11
        INTVL=1
        GO TO 12
11      IF(INTVL-21) 12,12,0
        INTVL=22
12      IHIST(INTVL)=IHIST(INTVL)+1
13      CONTINUE
C
C   OUTPUT SECTION
C   ------ -------
C
      N=-100
        DO 14 I=1,20
        LINE(I)=N
14      N=N+10
      WRITE(2,15) (LINE(I),I=1,20)
```

```
15     FORMAT(1H1//45X,32HDISTRIBUTION OF FORECAST ERRORS./45X,12(1H-),1X
      1,2H--,1X,8(1H-),1X,7(1H-)///10X,103(1H.)/10X,1H.,10X,1H.,90X,1H./1
      20X,1H.,16H  CLASS    .        ,20I4,5X,1H./10X,1H.,16H              . LT
      3 ,20(4H  TO),6H GT   .)
       N=-90
         DO 16 I=1,20
         LINE(I)=N
16       N=N+10
       WRITE(2,17) (LINE(I),I=1,20)
17     FORMAT(10X,12H. INTERVAL .,5H -100,20I4,6H 100 ./10X,1H.,10X,1H.,9
      10X,1H./10X,103(1H.)/10X,1H.,10X,1H.,90X,1H.)
C
C      WRITE FREQUENCIES.
C
       WRITE(2,18) FREQ,(IHIST(I),I=1,22)
18     FORMAT(10X,2H. ,A8,2H .,I4,1X,21I4,2H ./10X,1H.,10X,1H.,90X,1H./10
      1X,103(1H.))
C
C      CALCULATE PROBABILITIES. (DIVISOR IS 'NOACT-1' BECAUSE ACT(1) IS
C      NOT USED TO CALCULATE A FORECAST ERROR IN THE FIRST PERIOD)
C
       NOACT=NOACT-1
         DO 19 I=1,22
19       PROB(I)=FLOAT(IHIST(I))/FLOAT(NOACT)
       WRITE(2,20)APROB,(PROB(I),I=1,22),SUMSQ
20     FORMAT(10X,1H.,10X,1H.,90X,1H./10X,2H. ,A8,2H .,1X,F3.2,1X,21(1X,F
      13.2),2H ./10X,1H.,10X,1H.,90X,1H./10X,103(1H.)///10X,'SUM OF SQUAR
      2ED ERRORS = ',F9.1)
       STOP
       END
       FINISH
      FUNCTION FUNC1(I,IPDS)
      COMMON LINE(106),IDATA(3,100,5),NO,NOHORMK,IHORMK(16),HORMK(15),
     1NOVRTMK,IVRTMK(16),VRTMK(15),DATAR(3,100,5)
C
C       ON EACH ENTRY, FUNC1 IS SET TO 'OLD FCST' AND A NEW FORECAST IS
C     CALCULATED ('A NU FCST'), WHICH IS THEN SET TO 'OLD FCST' FOR USE
C     IN NEXT ENTRY.
C
C       INITIAL VALUES ARE TAKEN FROM AN EARLIER RUN USED TO ESTABLISH
C     THE SUM OF SQUARED ERRORS.
C
      IF(I-1) 11,0,11
      OLD SM ER=-20.831
      OLD MAD=45.294
      OLD TS=0.45991
      FUNC1,OLD FCST=123.95
      ERR=62.1-OLD FCST
      GO TO 12
11    FUNC1=OLD FCST
      ERR=DATAR(2,I,IPDS)-OLD FCST
12    A NU SM ER=0.15*ERR+0.85*OLD SM ER
      A NU MAD=0.15*ABS(ERR)+0.85*OLD MAD
      A NU TS=A NU SM ER/A NU MAD
      AOLD TS=ABS(OLD TS)
      A NU FCST=AOLD TS*DATAR(2,I,IPDS)+(1-AOLD TS)*OLDFCST
      OLD FCST=A NU FCST
      OLD TS=A NU TS
      OLD MAD=A NU MAD
      OLD SM ER=A NU SM ER
      RETURN
      END
```

```
          FUNCTION FUNC2(I,IPDS)
          COMMON LINE(106),IDATA(3,100,5),NO,NOHORMK,IHORMK(16),HORMK(15),
         1NOVRTMK,IVRTMK(16),VRTMK(15),DATAR(3,100,5)
C
C              IF NEGATIVE SUBSCRIPTS ARE POSSIBLE THE FORECAST EQUATIONS
C              USE FUNCTION 'IS' TO AMEND THIS SITUATION. WHEN (I.GT.MXNEG)
C              THIS FUNCTION IS NOT USED.
C
          IF(I-1) 11,0,11
          READ(1,10) MXNEG,(DATAR(2,J,IPDS),J=60,60+MXNEG)
   10     FORMAT(I0/(10F0.0))
   11     FUNC2=DATAR(2,IS(I-1),IPDS)*0.6+48.24
          RETURN
          END
```

```
      FUNCTION IS(IX)
      IF(IX.LE.0) GO TO 11
      IS=IX
      RETURN
11    IS=60-IX
      RETURN
      END
      FINISH
```

APPENDIX P6

Program Description: Program RHO. Details are given within the program.

```
      PROGRAM (RHO )
      INPUT 1=CR0
      OUTPUT 2=LP0
      USE 3=/ARRAY
      OUTPUT 4=TP0
      COMPRESS INTEGER AND LOGICAL
      TRACE 2
      END
C
C                   SPEARMAN'S RHO ON OBSERVATION MATRIX.
C                   -------------------------------------
C
C          THIS PROGRAM WILL PROCESS OBSERVATION MATRICES CONTAINING UP
C     TO AND INCLUDING 25 OBSERVATIONS AND 20 VARIABLES. INPUT TAKES THE
C     FORM OF INTEGER SCORES, WHICH ARE SUBSEQUENTLY EXAMINED FOR TIES
C     CALCULATED FOR ALL N*(N+1)/2 VALUES POSSIBLE BETWEEN 'N' VARIABLES
C
C                                  ***********
C    INPUT SECTION
C    ----- -------
C
C          THE SIZE OF THE MATRIX IS DECLARED ON INPUT AS BEING
C     'NROW*NCOL' WHERE 'NROW' IS NO. OF VARIABLES,
C                  AND 'NCOL' IS NO. OF OBSERVATIONS.
C
C          THE OBSERVATION MATRIX IS READ ROW BY ROW. A VARIABLE DESCRIP-
C     TION IS WRITTEN IN CARD COLUMNS 1-30 FOLLOWED BY THE SCORES (25I0)
C
```

```
            MASTER
            DIMENSION  VDES(20,4),ISCORE(20,25),IEXTNT(5),AMIDRNK(5),T(20),
           1           RHO(20,20),RANK(20,25)
      C
      C     READ AND WRITE TEST REF.(A8)
      C
            READ(1,101) VDES(1,1)
      101    FORMAT(A8)
            WRITE(2,102) VDES(1,1)
      102   FORMAT(/'1TEST REF. ',A8/
           1       ' ---- ----'//)
            READ(1,11) NROW,NCOL
      11    FORMAT(2I0)
              DO 12 I=1,NROW
              READ(1,111) (VDES(I,L),L=1,4),(ISCORE(I,J),J=1,NCOL)
      111     FORMAT(3A8,A6,25I0)
      12      CONTINUE
      C
      C     PRINT INPUT DATA.
      C
            WRITE(2,13)
      13    FORMAT(//52X,18HOBSERVATION MATRIX/52X,11(1H-),1X,6(1H-)////3X,
           1116(1H.)/3X,1H.,36X,1H.,77X,1H./3X,5H. NO.,12X,8HVAPIABLE,12X,1H.,
           234X,9HSCORE NO.,34X,1H./3X,5H. ---,12X,8(1H-),12X,1H.,34X,9(1H-),
           3         34X,1H./  3X,1H.,36X,1H.,77X,1H./3X,1H.,36X,1H.,77X,1H.)
            WRITE(2,14) (I,I=1,25)
      14    FORMAT(3X,1H.,36X,1H.,25I3,2X,1H./  3X,1H.,36X,1H.,77X,1H./3X,1H.,
           136X,1H.,77X,1H.)
              DO 15 I=1,NROW
              WRITE(2,141) I,(VDES(I,L),L=1,4),(ISCORE(I,J),J=1,NCOL)
      141     FORMAT(3X,1H.,I3,2X,3A8,A6,2H .,25I3)
              WRITE(2,142)
      142     FORMAT(1H+,117X,1H./3X,1H.,36X,1H.,77X,1H.)
      15      CONTINUE
            WRITE(2,151)
      151   FORMAT(3X,116(1H.))
      C
      C    ROW TRANSFORMATIONS
      C    --- ---------------
      C
              DO 24 IROW=1,NROW
      C
```

```
C          CHANGE SCORES TO RANGE BETWEEN 1 AND 5. (ON INPUT SCORES MAY
C          RANGE BETWEEN -2 AND +2.
           DO 16 ICOL=1,NCOL
16         ISCORE(IROW,ICOL)=ISCORE(IROW,ICOL)+3
         CONTINUE
C
C          SORT SCORES INTO ARRAY 'IEXTNT', AFTER ZEROISING LATTER.
C
           DO 17 I=1,5
17         IEXTNT(I)=0
           DO 18 ICOL=1,NCOL
18         IEXTNT(ISCORE(IROW,ICOL))=IEXTNT(ISCORE(IROW,ICOL))+1
         NRNK=5
C
C          CHECK IF IEXTNT(I) IS ZERO. IF SO SET IEXTNT(J)=IEXTNT(J+1)
C          FOR J=1,N-1; SO ENSURING IEXTNT(1) TO IEXTNT(NRNK) ARE GREATER
C          THAN ZERO. ALSO REDUCE ALL SCORE VALUES GREATER THAN (I) BY
C          ONE. THUS AT END OF SEQUENCE ALL SCORES HAVE BECOME RANKS
C          (POSSIBLY TIED) OF NATURAL ORDER.
C
           DO 19 I=1,5
           IF(IEXTNT(I))0,0,19
           IF(I.GT.NRNK) GO TO 19
           IF(I.EQ.5) GO TO 183
              DO 181 J=I,4
181           IEXTNT(J)=IEXTNT(J+1)
           CONTINUE
              DO 182 ICOL=1,NCOL
              IF(ISCORE(IROW,ICOL).GT.I) ISCORE(IROW,ICOL)=ISCORE(IROW,
      1                                                                ICOL)-1
182           CONTINUE
183        IEXTNT(NRNK)=0
           NRNK=NRNK-1
19         CONTINUE
         CONTINUE
         IRNK,NTYS=0
C
C           CALCULATE THE MID-RANKS OF THE TIED SCORES. THESE DEPEND ON
C           THE NO. OF SCORES HAVING THE SAME VALUE.(IE. THE EXTENT OF THE
C           TIE) THE SUM OF RANKS OF ABSOLUTE VALUE CORRESPONDING TO THE
C           TIE IS CALCULATED USING THE FORMULA,
C                             N*(A+L)/2    WHERE  N=EXTENT OF TIE
```

```
C                                           A=FIRST ABSOLUTE RANK.
C                                           L=LAST      DITTO    .
C
          DO 21 I=1,NRNK
          IF(IEXTNT(I)-1) 21,201,0
          NTYS=NTYS+1
          SUM=IEXTNT(I)*(1+2*IRNK+IEXTNT(I))/2.0
          AMIDRNK(I)=SUM/IEXTNT(I)
201       IRNK=IRNK+IEXTNT(I)
          IF(IEXTNT(I).EQ.1) AMIDRNK(I)=IRNK
21        CONTINUE
        CONTINUE
C
          GIVE SCORES THEIR MID-RANK VALUES.
C
          DO 22 ICOL=1,NCOL
22        RANK(IROW,ICOL)=AMIDRNK(ISCORE(IROW,ICOL))
        T(IROW)=0.0
C
C         CALCULATE CORRECTION FACTOR FOR ROW  -  T(IROW)
C
          DO 23 I=1,NRNK
          IF(IEXTNT(I).EQ.1) GO TO 23
          T(IROW)=T(IROW)+(IEXTNT(I)**3-IEXTNT(I))/12.0
23        CONTINUE
24      CONTINUE
C
C    CALCULATION OF SPEARMAN'S RHO.
C    ----------- -- ---------- ----
C
      RHO(1,1)=0.0
      CALL FMOVE(RHO(1,1),RHO(2,1),399)
      CUBE N=(NCOL**3-NCOL)/6.0
        DO 25 I=1,NROW-1
        DO 25 J=I+1,NROW
        SUM=0.0
C          CALCULATE SUM OF SQUARED DIFFERENCES BETWEEN RANKINGS.
C
           DO 241 ICOL=1,NCOL
           DIFF=RANK(I,ICOL)-RANK(J,ICOL)
241        SUM=SUM+DIFF*DIFF
         ANUMERATOR=(CUBE N)-SUM-T(I)-T(J)
```

```
          DENOM1=(CUBE N)-2*T(I)
          DENOM2=(CUBE N)-2*T(J)
          RHO(J,I)=ANUMERATOR/SQRT(DENOM1*DENOM2)
25        CONTINUE
        CONTINUE
C
C         SET RHO(I,J)=1 FOR ALL I=J
C
          DO 251 IROW=1,NROW
251       RHO(IROW,IROW)=1.0
C
C    PAPER TAPE OUTPUT
C    ----- ---- ------
C
        WRITE(4,252)
252     FORMAT(8HDOC TAPE)
        NPAR=(NROW*NROW+NROW)/2
        WRITE(4,2521) NPAR
2521    FORMAT(23HOBSERVATION MATRIX,,RHO/9HMATRIX,1,,I3,1H,,10HRHO,RW,1,1
       1)
        WRITE(4,253) ((RHO(I,J),J=1,I),I=1,NROW)
253     FORMAT((2X,10(F5.2,1X)))
        WRITE(4,255)
255     FORMAT(11HEND OF DATA/6HSWITCH/4H****)
C
C    FORMATTING SECTION
C    ---------- -------
C
        CALL DEFBUF(3,2000,ISCORE)
C
C       WRITE 'RHO' TO DEFBUF IN 'F4.2' FORMAT AND READ BACK IN 'A' FORMAT
C
        WRITE(3,26)((RHO(I,J),I=1,NROW),J=1,NROW)
26      FORMAT(400F5.2)
        READ(3,261) ((RHO(I,J),I=1,NROW),J=1,NROW)
261     FORMAT(400A5)
C
C       ALL 'RHO' VALUES ARE NOW HELD AS 5 CHAR. NOS. (IE. '12.45' WITH
C       DEC. POINT IN POSITION 3)
C
C          COPY BLANKS INTO UPPER TRIANGLE
C
```

```
             DO 27 I=1,NROW-1
             DO 27 J=I+1,NROW
      27     CALL COPY8(RHO(I,J),8H        )
            CONTINUE
      C
      C      OVER-WRITE DECIMAL POINT WITH SIGN  IE. 0.98 BECOMES  98 AND
      C                                              -.61 BECOMES -61
      C
             DO 28 I=1,NROW
             DO 28 J=1,I
      C
      C      CHECK IF CHAR. 2 IS '1'. IF SO CHECK IF CHAR. 1 IS '-'. IN
      C      EITHER CASE COPY APPROPRIATE VALUE INTO 'RHO' AND PROCEED.
      C
             L=1
             CALL COMP(L,RHO(I,J),2,1H1,1)
             IF(L-1) 272,0,272
             L=1
             CALL COMP(L,RHO(I,J),1,1H-,1)
             IF(L-1) 271,0,271
             CALL COPY(3,RHO(I,J),1,3H-1R,1)
             GO TO 28
      271    CALL COPY(3,RHO(I,J),1,3H+1R,1)
             GO TO 28
      C
      C      IF 'RHO' IS FRACTIONAL, COPY CHAR. 4 & 5 OF 'DUMMY' INTO POSNS.
      C      2 & 3 OF 'RHO' AND SO PRESERVE SIGN.
      C
      272    CALL COPY8(DUMMY,RHO(I,J))
             CALL COPY(2,RHO(I,J),2,DUMMY,4)
      28     CONTINUE
            IF(NROW.EQ.20) GO TO 32
      C
      C      SET REMAINING ROWS AND COLUMNS TO BLANK.
      C
             DO 30 I=NROW+1,20
             DO 30 J=1,20
      30     CALL COPY8(RHO(I,J),8H        )
             DO 31 I=1,NROW
             DO 31 J=NROW+1,20
      31     CALL COPY8(RHO(I,J),8H        )
      C
```

```
C    OUTPUT SECTION
C    ------ -------
C
32    WRITE(2,33) (I,I=1,20)
33    FORMAT(1H1//45X,33HSPEARMAN'S RHO CORRELATION MATRIX/45X,10(1H-),1
     1X,3H---,1X,11(1H-),1X,6(1H-)//52X,17HBETWEEN VARIABLES/52X,7(1H-),
     21X,9(1H-)////13X,95(1H.)/13X,1H.,10X,1H.,82X,1H./13X,1H.,10H VARIA
     3BLE ,1H.,82X,1H./13X,1H.,10X,1H.,82X,1H./13X,1H.,3X,3HNO.,4X,1H.,
     420I4,2X,1H./13X,1H.,10X,1H.,82X,1H./13X,95(1H.)/13X,1H.,10X,1H.,82
     5X,1H.)
      WRITE(2,34) (I,(RHO(I,J),J=1,20),I=1,20)
34    FORMAT((13X,1H.,4X,I2,4X,1H.,20(1X,A3),2X,1H./13X,1H.,10X,1H.,82X,
     11H.))
      WRITE(2,35)
35    FORMAT(13X,95(1H.))
      STOP
      END
      FINISH
```

APPENDIX P7

Program Description: Program GOODS. See Appendix M3 for details of operation and organisation.

```
PROGRAM(GOOD)
INPUT 1=CR0
USE 2=/ARRAY
OUTPUT 3=LP7
COMPRESS INTEGER AND LOGICAL
TRACE 2
END
```

```
       MASTER GOODS
        INTEGER REFPT(10),PT(10),AST,COMMA,EMM,BLANK
       0DIMENSION DES(5,5),FUNC(100,5),IFUNC(100,5),FDES(5,5),HORMK(15),IH
       1ORMK(16),VRTMK(15),IVRTMK(16),LINE(106),NOENTRY(5),JMKR(5),JA(5),
       2JX(5),DATAR(3,100,5),IDATA(3,100,5),NK(5),DATAL(100)
        DIMENSION STD FORMAT(11)
        COMMON LINE,IDATA,NO,NOHORMK,IHORMK,HORMK,NOVRTMK,IVRTMK,VRTMK,
       1DATAR,FUNC,IFUNC,NF,NOENTRY,IPDS
        COMMON /X/CENTRE(10)
       0DATA REFPT/'*   ','+   ',':   ','#   ','X   ','A   ','B   ','C   '
       1,'D   ','E   '/,AST/ '*   '/,BLANK/'    '/,COMMA/',   '/,EMM/'M
       2'/,NULL/'0   '/,IF/'I   '/,MINUS/'-   '/
        DATA LPAREN/'(   '/,IRPAREN/')   '/
C
C           PRINT TITLE SHEET ON FIRST RUN
C
        WRITE(3,15)
     15 FORMAT(1H1,22(/))
        CALL TITLE
        CALL DATE(D1)
        WRITE(3,16) D1
     16 FORMAT(10(/),16X,'DATE ',A8/16X,13(1H-))
C
C           CONTROL CARD AND ASSOCIATED INPUT - ALSO DATA INPUT
C
       0READ(1,1) N,NO,NF,IPDS,IOF,ISETSCL,ISELPT,LORD,LABS,ISUPRZ,LOGX,
       1LOGY,LNX,LNY,IFMAT,ISEP,ILINE
      1 FORMAT(I2,6I1,2A1,8I1)
        NG=N
        ISTART=0
        DO 100 JW=1,NG
        CALL FMOVE(REFPT,PT,5)
        IVRTMK(1),IVRTMK(2)=0
        CALL FMOVE(IVRTMK(1),IVRTMK(3),7)
        NXMIN,NYMIN=0
        NOHORMK,NOVRTMK=0
        IF(ISTART) 0,2,0
       0READ(1,1) N,NO,NF,IPDS,IOF,ISETSCL,ISELPT,LORD,LABS,ISUPRZ,LOGX,
       1LOGY,LNX,LNY,IFMAT,ISEP,ILINE
C
C           PRINT NOTES ON DATA, DETAILS OF ANNOTATION ETC.
C
```

```
    2 ISTART=1
      IF(ISUPRZ-1) 0,14,0
      WRITE(3,61)
   61 FORMAT(1H1)
      CALL TITLE
   14 IF(ISETSCL-1) 7,0,7
      READ(1,8) XMIN,XMAX,YMIN,YMAX
    8 FORMAT(4F0.0)
    7 IF(ISELPT-1) 9,0,9
      READ(1,10) PT
   10 FORMAT(10A1)
    9 J=1
      CALL COMP(J,LORD,1,NULL,1)
      IF(J) 11,0,11
      READ(1,13) NOHORMK
   13 FORMAT(I0)
      READ(1,81) (HORMK(I),I=1,NOHORMK)
   81 FORMAT(10F0.0)
   11 J=1
      CALL COMP(J,LABS,1,NULL,1)
      IF(J) 12,0,12
      READ(1,13) NOVRTMK
      READ(1,81) (VRTMK(I),I=1,NOVRTMK)
      IVCHEK=1
   12 IFSET=0
      IF((IPDS-NO).EQ.1) IFSET=1
      IF(ILINE-1) 121,0,121
      READ(1,122) XMIN,XMAX,NOLINES
  122 FORMAT(2F0.0,I0)
      IF(ILINE-1000) 121,121,0
      WRITE(3,123)
  123 FORMAT(' VALUE OF "ILINE" EXCEEDS 1000 - PROGRAM HALTED')
      STOP
C
C          READ DATA
C
  121 DO 3 N=1,NO+IFSET
      READ(1,4) (DES(I,N),I=1,5)
    4 FORMAT(5A8)
      IF(ISEP-1) 311,0,311
      READ(1,5) NOENTRY(N),(DATAR(1,I,N),I=1,NOENTRY(N))
      READ(1,312) (DATAR(2,I,N),I=1,NOENTRY(N))
  312 FORMAT(10F0.0)
      GO TO 3
  311 READ(1,5) NOENTRY(N),(DATAR(1,I,N),DATAR(2,I,N),I=1,NOENTRY(N))
    5 FORMAT(I0/(10F0.0))
    3 CONTINUE
      IF(LOGX-1) 90,0,90
      DO 91 J=1,NO+IFSET
      DO 91 I=1,NOENTRY(J)
   91 DATAR(1,I,J)=ALOG10(DATAR(1,I,J))
   90 IF(LOGY-1) 92,0,92
      DO 93 J=1,NO
      DO 93 I=1,NOENTRY(J)
   93 DATAR(2,I,J)=ALOG10(DATAR(2,I,J))
   92 IF(LNX-1) 94,0,94
      DO 95 J=1,NO+IFSET
      DO 95 I=1,NOENTRY(J)
   95 DATAR(1,I,J)=ALOG(DATAR(1,I,J))
   94 IF(LNY-1) 96,0,96
      DO 97 J=1,NO
      DO 97 I=1,NOENTRY(J)
   97 DATAR(2,I,J)=ALOG(DATAR(2,I,J))
96    IF(NF) 24,24,0
      READ(1,4) ((FDES(I,N),I=1,5),N=1,NF)
      DO 72 J=1,NOENTRY(IPDS)
   72 DATAL(J)=DATAR(1,J,IPDS)
   24 IF(ISETSCL-1) 0,27,0
C
C          CALCULATE MAX AND MIN VALUES OF ORDINATES
C
      YMAX,YMIN=DATAR(2,1,1)
      DO 25 J=1,NO
      DO 25 I=1,NOENTRY(J)
      IF(DATAR(2,I,J)-YMAX) 26,26,0
      YMAX=DATAR(2,I,J)
      GO TO 25
   26 IF(DATAR(2,1,J)-YMIN) 0,25,25
      YMIN=DATAR(2,1,J)
   25 CONTINUE
C
C          CALCULATE MIN. AND MAX. VALUES OF FUNCTIONS
C
      IF(NF) 106,106,0
      DO 21 I=1,NOENTRY(IPDS)
C     FUNC(I,1)=FUNC1(J1,J2,---JN)
C     FUNC(I,2)=FUNC2(J1,J2,---JN)
C               DITTO
C               DITTO
C     FUNC(I,5)=FUNC5(J1,J2,---JN)
   21 CONTINUE
      FMAX,FMIN=FUNC(1,1)
      DO 101 J=1,NF
      DO 101 I=1,NOENTRY(IPDS)
      IF(FMIN-FUNC(I,J)) 102,102,0
      FMIN=FUNC(I,J)
      GO TO 101
  102 IF(FMAX-FUNC(I,J))  0,101,101
      FMAX=FUNC(I,J)
  101 CONTINUE
      IF(YMIN-FMIN) 103,103,0
      YMIN=FMIN
  103 IF(YMAX-FMAX)  0,106,106
      YMAX=FMAX
C
C          CALCULATE MAX AND MIN VALUES OF ABSCISSAE. IT IS ASSUMED THESE
C                    ARE IN INCREASING ORDER OF MAGNITUDE
C
  106 IF(ILINE-1) 0,27,0
      XMAX=DATAR(1,NOENTRY(1),1)
      XMIN=DATAR(1,1,1)
      DO 27 J=1,NO+IFSET
      IF(XMAX-DATAR(1,NOENTRY(J),J)) 0,28,28
      XMAX=DATAR(1,NOENTRY(J),J)
   28 IF(XMIN-DATAR(1,1,J))27,27,0
      XMIN=DATAR(1,1,J)
   27 CONTINUE
C
C          PRINT DATA SERIES
C
      IF(ISUPRZ) 177,0,177
      IF(IFMAT.EQ.0) GO TO 171
      DO 20 J=1,NO
   20 WRITE(3,17) (DES(I,J),I=1,5),DATAR(1,1,J),DATAR(2,1,J),(COMMA,DATA
     1R(1,I,J),DATAR(2,I,J),I=2,NOENTRY(J))
   17 FORMAT(////1X,5A8/41(1H-)//(/2H (,E12.5,',',E12.5,1H),3(A1,1H(,E12
     1.5,',',F12.5,1H)),A1))
      GO TO 177
171   CALL DEFBUF(2,80,STD FORMAT)
      CALL DYN FORMAT(XMAX,XMIN,IFIELD X)
      CALL DYN FORMAT(YMAX,YMIN,IFIELD Y)
      IFIELD=IFIELD X+IFIELD Y+4
      NREPEAT=120/IFIELD-1
      NSPACF=(120-120/IFIELD*IFIELD)/2+1
      WRITE(2,175) NSPACE,IFIELD X,IFIELD Y,NREPEAT,IFIELD X,IFIELD Y
175   FORMAT('((/',I2,'X,A1,F',I2,'.1,A1,F',I2,'.1,A1,',I2,'(2A1,F',I2,'
     1.1,A1,F',I2,'.1,A1),A1))')
      DO 173 J=1,NO
      WRITE(3,174) (DES(I,J),I=1,5)
174   FORMAT(////1X,5A8/41(1H-)//)
      WRITE(3,STD FORMAT) LPAREN,DATAR(1,1,J),COMMA,DATAR(2,1,J),IRPAREN
     1,(COMMA,LPAREN,DATAR(1,I,J),COMMA,DATAR(2,I,J),IRPAREN,I=2,
     2NOENTRY(J))
173   CONTINUF
177   CONTINUF
      AXMIN=0.0
      AYMIN=0.0
      IF(YMIN) 0,76,76
      YLOG=10**IFIX(ALOG10(-YMIN))
      AYMIN=IFIX(-YMIN/YLOG)*YLOG
      IF(-YMIN-AYMIN.GT.0.0) AYMIN=AYMIN+YLOG
      YMIN=0.0
      IF(AYMIN.EQ.0.0) GO TO 76
      DO 79 J=1,NO+IFSET
      DO 79 I=1,NOENTRY(J)
   79 DATAR(2,I,J)=DATAR(2,I,J)+AYMIN
      YMAX=YMAX+AYMIN
C
C        AMEND HORMK FOR NEGATIVE DATA
C
      LORD=IE
      NOHORMK=NOHORMK+1
      HORMK(NOHORMK)=0.0
      DO 85 I=1,NOHORMK
   83 HORMK(I)=HORMK(I)+AYMIN
   76 IF(XMIN) 0,77,77
      XLOG=10**IFIX(ALOG10(-XMIN))
      AXMIN=IFIX(-XMIN/XLOG)*XLOG
      IF(-XMIN-AXMIN.GT.0.0) AXMIN=AXMIN+XLOG
      XMIN=0.0
      IF(AXMIN.EQ.0.0) GO TO 77
      DO 80 J=1,NO+IFSET
      DO 80 I=1,NOENTRY(J)
   80 DATAR(1,I,J)=DATAR(1,I,J)+AXMIN
      XMAX=XMAX+AXMIN
C
C             AMEND VRTMK FOR NEGATIVE DATA
C
      LABS=MINUS
      IF(NOVRTMK) 85,85,0
      DO 84 I=1,NOVRTMK
   84 VRTMK(I)=VRTMK(I)+AXMIN
      DO 86 I=1,NOVRTMK
      IF(VRTMK(I)-AXMIN) 86,77,0
      DO 87 J=1,NOVRTMK-I+1
   87 VRTMK(NOVRTMK+2-J)=VRTMK(NOVRTMK+1-J)
      VRTMK(I)=AXMIN
      GO TO 88
   86 CONTINUE
   88 NOVRTMK=NOVRTMK+1
      GO TO 77
   85 VRTMK(1)=AXMIN
      NOVRTMK=1
      IVCHEK=1
   77 L=1
      CALL EXPONENT(XMAX,IEXPX,L,X,IFSET)
      L=2
      IA=0
      CALL EXPONENT(YMAX,IEXPY,L,Y,IA)
      IXMIN=NINT(XMIN/X)
      IXMAX=NINT(XMAX/X)
      NXMIN=NINT(AXMIN/X)
      IYMIN=NINT(YMIN/Y)
      IYMAX=NINT(YMAX/Y)
C
C           PRINT FUNCTION CO-ORDINATES IF ANY, THEN SCALE
C
      IF(NF) 22,22,0
      IF(ISUPRZ-1) 0,22,0
      IF(IFMAT-1) 231,0,231
      DO 23 J=1,NF
   23 WRITE(3,17)(FDES(I,J),I=1,5),DATAL(1),FUNC(1,J),(COMMA,DATAL(I),FU
     1NC(I,J),I=2,NOENTRY(IPDS))
      GO TO 232
231   DO 233 J=1,NF
      WRITE(3,176)(FDES(I,J),I=1,5)
176   FORMAT(///1X,5A8/41(1H-)//)
      WRITE(3,STD FORMAT) LPAREN,DATAL(1),COMMA,FUNC(1,J),IRPAREN,
     1(COMMA,LPAREN,DATAL(I),COMMA,FUNC(I,J),IRPAREN,I=2,NOENTRY(IPDS))
233   CONTINUE
  232 DO 18 J=1,NF
      DO 18 I=1,NOENTRY(IPDS)
      FUNC(I,J)=(FUNC(I,J)+AYMIN)/Y
   18 IFUNC(I,J)=NINT(FUNC(I,J))
C
C           PRINT TITLE FOR GRAPH AND OUTPUT SCALE FACTORS (X10EXP)
C                       ALSO  CALCULATE AND PRINT SCALE
C
   22 WRITE(3,61)
      CALL TITLE
      WRITE(3,73)
   73 FORMAT(//54X,'SYMBOL KEY'/54X,10(1H-)/)
      IF(IOF-1) 0,69,0
      WRITE(3,70) (PT(J),(DES(I,J),I=1,5),J=1,NO)
   70 FORMAT(/41X,A1,' = ',5A8)
      IF(NF) 58,58,69
   69 WRITE(3,70) (PT(J+5),(FDES(I,J),I=1,5),J=1,NF)
   58 WRITE(3,32) IEXPX,IEXPY
   32 FORMAT(// 6X,'(ABSCISSA VALUES X 10EXP',I2,')',T85,'(ORDINATE VALU
     1ES X 10EXP',I2,')'//)
      INFO=2
      IF(AYMIN) 89,89,0
      NYMIN=NINT(AYMIN/Y)
   89 CALL SCALE(IYMIN,IYMAX,II,IV,INFO,IA,NYMIN)
      WRITE(3,33) (LINE(I),I=1,11)
   33 FORMAT(I6,10I10/5X,'*',20('    *'))
      IMD=1
      CALL SCALE(IXMIN,IXMAX,II,IV,IMD,IFSET,IA)
C
C           CHECK EFFECT OF SCALING IF 'GROUPING' HAS OCCURRED. GROUPING
C           CAUSES NOENTRY(J) TO BE REDUCED IN VALUE TO JX(J)
C
```

```
      J1=1
      DO 31 J=1,NO+IFSET
      JX(J)=0
      DO 29 I=2,NOENTRY(J)
      IF(IDATA(1,I,J)-IDATA(1,I-1,J)) 0,0,30
      J1=J1+1
      GO TO 29
   30 JX(J)=JX(J)+1
      IDATA(3,JX(J),J)=J1
      J1=1
      IDATA(1,JX(J),J)=IDATA(1,I-1,J)
   29 CONTINUE
      JX(J)=JX(J)+1
      IDATA(3,JX(J),J)=J1
      J1=1
      IDATA(1,JX(J),J)=IDATA(1,NOENTRY(J),J)
   31 CONTINUE
C
C          OUTPUT SECTION
C
      JQ=II-NXMIN
   34 FORMAT(I5,104(1H*))
      WRITE(3,34) JQ
   36 FORMAT(1H+,109X,1H(,I3,1H,,I3,1H),'=*')
   35 LINE(1),LINE(2)=BLANK
      CALL FMOVE(LINE(1),LINE(3),52)
      DO 37 J=1,5
      JA(J)=0
   37 JMKR(J)=1
C
C          ABSCISSAE CONTROL LOOP
C
      JQL=-NXMIN
      NLINE=102
      IF(ILINE.EQ.1) NLINE=NOLINES
      DO 44 JQQ=1,NLINE
      LK=0
      JQ=JQQ+II-1
      IF(MOD(JQ,10)) 19,0,19
      IF(JQQ-1) 0,19,0
      JQL=JQL+IMD
   19 NU=1
```

```
      IF(IOF-1) 0,40,0
C
C          DATA SERIES OUTPUT LOOP
C
      DO 39 J=1,NO
      NK(J)=0
     0IF((IDATA(1,JMKR(J),J).NE.JQ).OR.(JQ.GT.IDATA(1,JX(J),J))) GO TO 3
     19
      NK(J)=1
      LINE(1)=AST
      K,JA(J)=JA(J)+1
      JV=IDATA(2,K,J)+1
      LINE(JV)=PT(J)
      IF(MOD(JQ,10)) 41,0,41
      IF(JQ-II) 42,0,42
      NU=2
      WRITE(3,47) LINE
   47 FORMAT(1H+,4X,106A1)
      LINE(JV)=BLANK
      IF(JV-1) 48,0,48
     0IF((J.EQ.1).AND.(ISELPT.EQ.0)) WRITE(3,36) IDATA(1,JMKR(1),1),
     1IDATA(2,JMKR(1),1)
      GO TO 48
   42 CALL SELFORMAT(NU,JV,JQL,BLANK)
      IF(JV-1) 48,0,48
      IX=IDATA(1,JMKR(1),1)-NXMIN
      IF((J.EQ.1).AND.(ISELPT.EQ.0)) WRITE(3,36) IX,IDATA(2,JMKR(1),1)
      GO TO 48
   41 CALL SELTWO (NU,JV,BLANK)
      IF(JV-1) 48,0,48
      IX=IDATA(1,JMKR(1),1)-NXMIN
      IF((J.EQ.1).AND.(ISELPT.EQ.0)) WRITE(3,36) IX,IDATA(2,JMKR(1),1)
   48 IL=IDATA(3,JMKR(J),J)
      JMKR(J)=JMKR(J)+1
      IF(IL-1) 39,39,0
      K,JA(J)=JA(J)-1
      SUM=0.0
      DO 49 IQ=1,IL
      K=K+1
      IK=IDATA(2,K,J)
      SUM=SUM+IK
      IK=IK+1
```

```
            LINE(IK)=PT(J)
            WRITE(3,47) LINE
         49 LINE(IK)=BLANK
            JA(J)=K
            IAV=NINT(SUM/IL+1)
            LINE(IAV)=EMM
            WRITE(3,47) LINE
            LINE(IAV)=BLANK
         39 CONTINUE
      C
      C           FUNCTION OUTPUT LOOP
      C
         40 IF(NF) 50,50,0
            IF(JQ-IDATA(1,JX(IPDS),IPDS))0,0,50
            IF(IPDS-NO+1) 0,64,0
            IF(NK(IPDS)) 62,64,62
         64 IF(JQ-IDATA(1,JMKR(IPDS),IPDS))50,63,50
         62 IF(JQ-IDATA(1,JMKR(IPDS)-1,IPDS)) 50,0,50
            JMKR(IPDS)=JMKR(IPDS)-1
            LK=1
         63 DO 51 IN=1,NF
            IY=IFUNC(JMKR(IPDS),IN)+1
            LINE(IY)=PT(IN+5)
            IF(IOF-1) 52,0,52
            LINE(1)=AST
            LINE(IY)=PT(IN+5)
            IF(MOD(JQ,10)) 53,0,53
            IF(JQ-II) 54,0,54
            WRITE(3,47) LINE
            LINE(1),LINE(IY)=BLANK
            GO TO 51
         54 CALL SELFORMAT(NU,IY,JQL,BLANK)
            GO TO 51
         53 CALL SELTWO(NU,IY,BLANK)
            GO TO 51
         52 WRITE(3,47) LINE
            LINE(IY)=BLANK
         51 CONTINUE
           0IF((IOF.EQ.1).OR.(IPDS.EQ.NO+1).OR.(LK.EQ.1)) JMKR(IPDS)=JMKR(IPDS
           1)+1
         50 IF(NU-2) 0,55,0
            IF(JQ.EQ.II) GO TO 55
      IF(MOD(JQ,10)) 56,0,56
      IF(NOHORMK) 68,68,0
      DO 71 I=1,NOHORMK
   71 LINE(IHORMK(I)+1)=LORD
      LINE(1)=AST
      WRITE(3,74) JQL,LINE
   74 FORMAT(I5,106A1)
      GO TO 59
   68 WRITE(3,57) JQL
   57 FORMAT(I5,1H*)
      GO TO 59
   56 WRITE(3,45)
   45 FORMAT('      *')
C
C         CHECK FOR HORMK AND VRTMK
C
   55 IF(NOHORMK) 59,59,0
      DO 60 I=1,NOHORMK
   60 LINE(IHORMK(I)+1)=LORD
      WRITE(3,47) LINE
      LINE(1),LINE(2)=BLANK
   59 IF(NOVRTMK) 44,44,0
      CALL FMOVE(LINE(1),LINE(3),52)
      IF(JQ-IVRTMK(IVCHEK)) 44,0,44
      IVCHEK=IVCHEK+1
      LINE(1),LINE(2)=LABS
      CALL FMOVE(LINE(1),LINE(3),51)
      WRITE(3,47) LINE
      LINE(1),LINE(2)=BLANK
      CALL FMOVE(LINE(1),LINE(3),51)
   44 CONTINUE
  100 CONTINUE
      STOP
      END
      SUBROUTINE SCALE(MINVL,MAXVL,II,IV,IMD,IFSET,NEG)
     0DIMENSION LINE(106),IDATA(3,100,5),IHORMK(16),HORMK(15),
     1IVRTMK(16),VRTMK(15),DATAR(3,100,5),NOENTRY(5),FUNC(100,5),IFUNC(1
     200,5)
      COMMON LINE,IDATA,NO,NOHORMK,IHORMK,HORMK,NOVRTMK,IVRTMK,VRTMK,
     1DATAR,FUNC,IFUNC,NF,NOENTRY
      IWRN=0
      IDIFF=MAXVL-MINVL
   17 IF(IDIFF-50) 0,0,1
      IF(IDIFF-20) 0,0,2
      IWRN=1
      II=MINVL/10*10
      IV=II+20
      IF(MAXVL-IV) 0,0,2
      AMULT=5.0
      CALL CHANJSCL(IFSET,IDIFF,IMD,&17,AMULT,II)
      IF(IMD-2) 0,4,0
      IMD=2
      IF(NOVRTMK) 12,12,0
      DO 10 I=1,NOVRTMK
   10 IVRTMK(I)=NINT((VRTMK(I)-II)*5.0)
      IF(NOVRTMK-1)12,12,0
      CALL IVRTRANK
   12 RETURN
    4 IMD=5
      IF(NOHORMK) 13,13,0
      DO 14 J=1,NOHORMK
   14 IHORMK(J)=NINT((HORMK(J)-II)*5.0)
   13 J=0
      DO 3 I=1,11
      LINE(I)=II+J-NEG
    3 J=J+2
      RETURN
    2 IF(MAXVL-50) 0,0,5
      IWRN=1
      II=0
      IV=50
      AMULT=2.0
      CALL CHANJSCL(IFSET,IDIFF,IMD,&17,AMULT,II)
      IF(IMD-2) 0,6,0
      IMD=5
      IF(NOVRTMK) 18,18,0
   21 DO 19 I=1,NOVRTMK
   19 IVRTMK(I)=NINT((VRTMK(I)-II)*2.0)
      IF(NOVRTMK-1)18,18,0
      CALL IVRTRANK
   18 RETURN
    6 J=0
      DO 11 I=1,11
      LINE(I)=II+J-NEG
   11 J=J+5
      IMD=2
      IF(NOHORMK) 15,15,0
      DO 16 J=1,NOHORMK
   16 IHORMK(J)=NINT((HORMK(J)-II)*2.0)
   15 RETURN
    5 IF(MINVL-50) 1,1,0
      IWRN=1
      II=50
      IV=100
      AMULT=2.0
      CALL CHANJSCL(IFSET,IDIFF,IMD,&17,AMULT,II)
      IF(IMD-2) 0,6,0
      IMD=5
      IF(NOVRTMK) 0,0,21
      RETURN
    1 II=0
      IV=100
      IF(IWRN-1) 7,0,7
      AMULT=1.0
      CALL CHANJSCL(IFSET,IDIFF,IMD,&17,AMULT,II)
    7 IF(IMD-2) 0,8,0
      IMD=10
      IF(IWRN-1) 22,0,22
      IF(NOVRTMK) 22,22,0
      DO 23 I=1,NOVRTMK
   23 IVRTMK(I)=NINT(VRTMK(I))
      IF(NOVRTMK-1) 22,22,0
      CALL IVRTRANK
   22 RETURN
    8 J=0
      DO 9 I=1,11
      LINE(I)=II+J-NEG
    9 J=J+10
```

```
      RETURN
      END
      SUBROUTINE SELFORMAT(NU,JV,JQ,BLANK)
      INTEGER BLANK
      DIMENSION LINE(106)
      COMMON LINE
    1 FORMAT(1H+,4X,106A1)
    2 FORMAT(I5,106A1)
    3 FORMAT(5X,106A1)
      GO TO(0,4),NU
      NU=2
      WRITE(3,2) JQ,LINE
      LINE(1),LINE(JV)=BLANK
      RETURN
    4 LINE(1)=BLANK
      WRITE(3,1) LINE
      LINE(JV)=BLANK
      RETURN
      ENTRY SELTWO(NU,JV,BLANK)
      GO TO(0,5),NU
      NU=2
      WRITE(3,3) LINE
      LINE(1),LINE(JV)=BLANK
      RETURN
    5 LINE(1)=BLANK
      WRITE(3,1) LINE
      LINE(JV)=BLANK
      RETURN
      END
      SUBROUTINE EXPONENT(TMAX,IEXPN,L,X,IFSET)
     0DIMENSION LINE(106),IDATA(3,100,5),DATAR(3,100,5),FUNC(100,5
     1),IFUNC(100,5),VRTMK(15),IVRTMK(16),HORMK(15),IHORMK(16),NOENTRY(5
     2)
      COMMON LINE,IDATA.NO,NOHORMK,IHORMK,HORMK,NOVRTMK,IVRTMK,VRTMK,
     1DATAR,FUNC,IFUNC,NF,NOENTRY,IPDS
      EXPN=ALOG10(TMAX)
      IEXPN=EXPN
      IF(EXPN-IEXPN) 0,0,1
      IEXPN=IEXPN-2
      GO TO 5
    1 IEXPN=IEXPN-1
    5 X=10.0**IEXPN
      DO 6 J=1,NO+IFSET
      DO 6 I=1,NOENTRY(J)
      DATAR(L,I,J)=DATAR(L,I,J)/X
    6 IDATA(L,I,J)=NINT(DATAR(L,I,J))
      IF(L-1) 7,0,7
      IF(NOVRTMK) 13,13,0
      DO 9 I=1,NOVRTMK
      VRTMK(I)=VRTMK(I)/X
    9 IVRTMK(I)=NINT(VRTMK(I))
      IF(NOVRTMK-1) 13,13,0
      CALL IVRTRANK
      GO TO 13
    7 IF(NOHORMK) 13,13,0
      DO 10 I=1,NOHORMK
      HORMK(I)=HORMK(I)/X
   10 IHORMK(I)=NINT(HORMK(I))
   13 RETURN
      END
      SUBROUTINE CHANJSCL(IFSET,IDIFF,L,*,AMULT,II)
      DIMENSION LINE(106),IDATA(3,100,5),IHORMK(16),HORMK(15),IVRTMK(16)
     1,VRTMK(15),DATAR(3,100,5),FUNC(100,5),IFUNC(100,5),NOENTRY(5)
      COMMON LINE,IDATA,NO,NOHORMK,IHORMK,HORMK,NOVRTMK,IVRTMK,VRTMK,
     1DATAR,FUNC,IFUNC,NF,NOENTRY,IPDS
      DO 1 J=1,NO+IFSET
      DO 1 I=1,NOENTRY(J)
      IDATA(L,I,J)=NINT((DATAR(L,I,J)-II)*AMULT)
      IF(IDATA(L,I,J)) 0,1,1
      IDIFF=IDIFF+1
      RETURN 1
    1 CONTINUE
      IF(L-2) 2,0,2
      IF(NF) 2,2,0
      DO 3 J=1,NF
      DO 3 I=1,NOENTRY(IPDS)
      IFUNC(I,J)=NINT((FUNC(I,J)-II)*AMULT)
      IF(IFUNC(I,J)) 0,3,3
      IDIFF=IDIFF+1
      RETURN 1
    3 CONTINUE
    2 RETURN
      END
      SUBROUTINE TITLE
      DIMENSION LINE(106)
      COMMON LINE
      COMMON /X/CENTRE(10)
      DATA IBLANK/4H    /
      CALL DEFBUF(2,80,CENTRE)
    3 READ(1,1) MK,(LINE(I),I=1,77),ICNT
    1 FORMAT(I1,77A1,I2)
      IF(ICNT.EQ.0) GO TO 4
      ICTR=(80-ICNT)/2
      WRITE(2,5) (IBLANK,I=1,ICTR),(LINE(I),I=1,ICNT-1)
    5 FORMAT(80A1)
      II=ICTR+ICNT-1
      READ(2,5) (LINE(I),I=1,II)
    2 FORMAT(21X,80A1)
      WRITE(3,2) (LINE(I),I=1,II)
      GO TO 6
    4 WRITE(3,2) (LINE(I),I=1,77)
    6 IF(MK.EQ.1) GO TO 3
      RETURN
      END
      SUBROUTINE IVRTRANK
     0DIMENSION LINE(106),IDATA(3,100,5),IHORMK(16),HORMK(15),IVRTMK(16)
     1,VRTMK(15)
      COMMON LINE,IDATA,NO,NOHORMK,IHORMK,HORMK,NOVRTMK,IVRTMK,VRTMK
      K=0
      DO 1 I=2,NOVRTMK
      IF(IVRTMK(I)-IVRTMK(I-1)) 1,1,0
      K=K+1
      IVRTMK(K)=IVRTMK(I-1)
    1 CONTINUE
      IF(IVRTMK(NOVRTMK)-IVRTMK(NOVRTMK-1)) 2,2,0
      K=K+1
      IVRTMK(K)=IVRTMK(NOVRTMK)
    2 RETURN
      END
      SUBROUTINE DYN FORMAT(TMAX,TMIN,IFIELD)
      IF(TMAX) 1,0,1
      TMAX=3.0
      IFIELD=3
      GO TO 2
1     TMAX LOG=ALOG10(ABS(TMAX))
      IFIELD=TMAX LOG+3
2     IF(TMIN.GE.0.0) RETURN
      TMIN LOG=ALOG10(-TMIN)
      IF(TMAX) 3,3,0
      IF(TMIN LOG+1.GT.TMAX LOG) IFIELD=TMIN LOG+4
      RETURN
3     IFIELD=TMIN LOG+4
      RETURN
      END
      FINISH
      SUBROUTINE CONCARD(I1,I2,I3,I4,I5,I6,I7,I8,I9,I10,I11,I12,I13,I14,
     1                   I15,I16)
      COMMON/CONTROL/NO,NF,IPDS,IOF,ISETSCL,ISELPT,LORD,LABS,ISUPRZ,LOGX
     1,LOGY,LNX,LNY,IFMAT,      ILINE,IXSCALE
      NO=I1
      NF=I2
      IPDS=I3
      IOF=I4
      ISETSCL=I5
      ISELPT=I6
      LORD=I7
      LABS=I8
      ISUPRZ=I9
      LOGX=I10
      LOGY=I11
      LNX=I12
      LNY=I13
      IFMAT=I14
      ILINE=I15
      IXSCALE=I16
      RETURN
      END
      FINISH
```