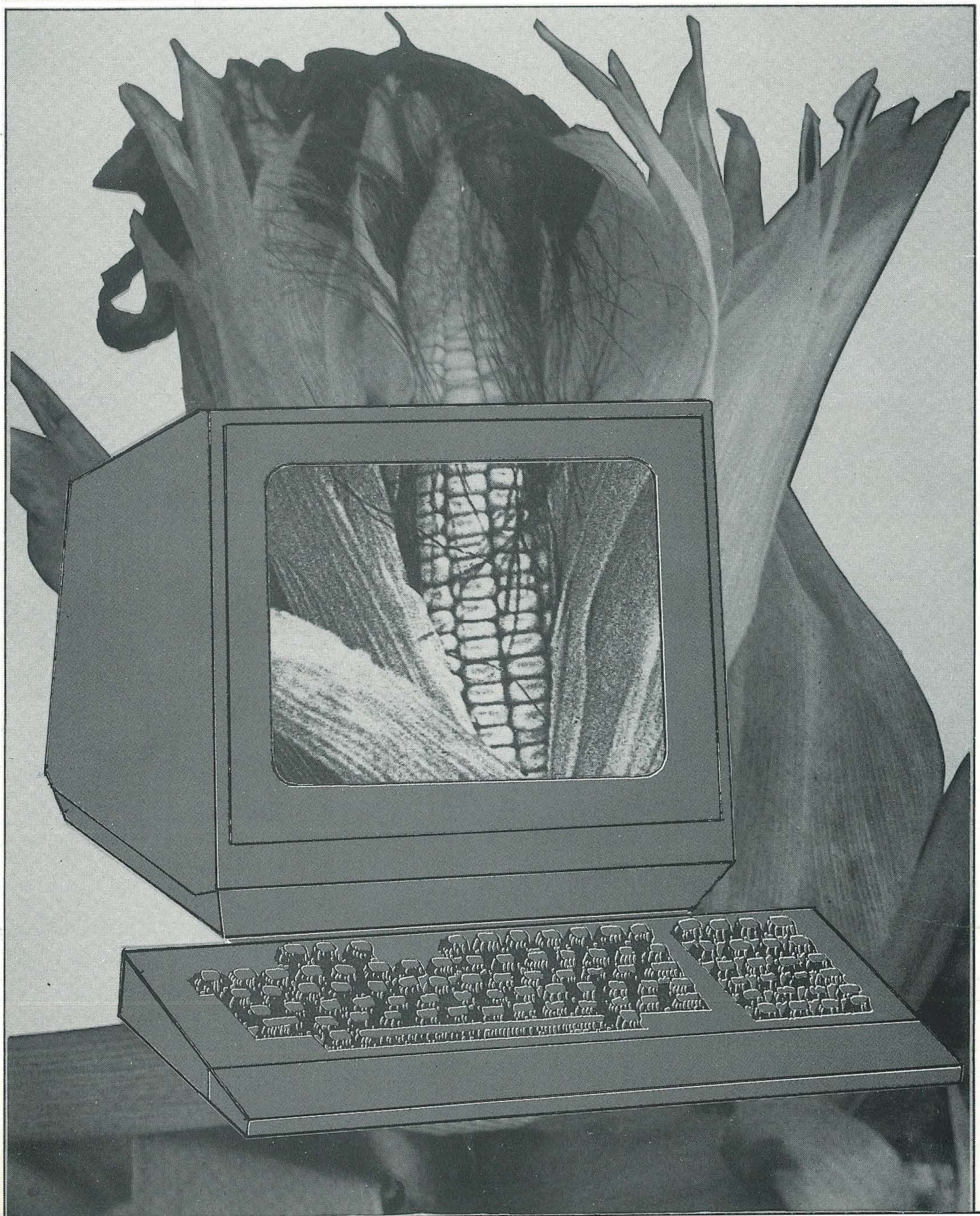
Using data processing tools for preparing agricultural development projects

FAO
INVESTMENT
CENTRE
TECHNICAL
PAPER

2



FOOD
AND
AGRICULTURE
ORGANIZATION
OF THE
UNITED NATIONS

Using data processing tools for preparing agricultural development projects

by
Michel Siméon

FAO INVESTMENT CENTRE TECHNICAL PAPER
2

FOOD AND AGRICULTURE ORGANIZATION OF THE UNITED NATIONS
Rome, 1985

M-63
ISBN 92-5-102255-0

FOREWORD

The FAO Investment Centre's principal function is to assist member countries in the identification and preparation of agricultural and rural development projects for international, and sometimes local, financing. The Investment Centre is also acknowledged as a source of assistance in the development of national capacities for project preparation.

Since the establishment of the FAO/World Bank Cooperative Programme (the precursor of the Investment Centre) twenty years ago, there has been a considerable evolution in project preparation techniques. This has been reflected in successive revisions to the various guidelines on project preparation published by the Investment Centre and widely used by government agencies and consulting firms responsible fôr feasibility studies.

The rapidly increasing accessibility of powerful computers has opened up exciting possibilities for project analysts and economists. Apart from the advantage of reducing the time required for lengthy calculations, the computer makes it possible to extend the range of analyses which are conducted with little extra effort, improves accuracy and may contribute towards the evolution and standardization of methodologies.

The Investment Centre has responded to these new opportunities by developing a series of programmes designed to address the particular requirements of agricultural project analysis. Leadership in this work has been assumed by Michel Siméon, a Senior Economist in the Investment Centre. In this paper, which was prepared in his spare time as a post-graduate thesis, he reviews the types of analysis required in project preparation, identifies various roles for computers, outlines the conceptual basis for several programmes developed both by the Investment Centre and by other organizations, and explains how these may be used. Annexes to the paper include user's guides for some of the programmes.

Since I believe that the paper could be of considerable use to people outside the Investment Centre who share our interest in the development of new methodologies for application in project preparation, I have decided to give it wider circulation by publishing it (initially in French and English) as part of our series of guidelines. The opinions expressed, however, remain those of the author and are not necessarily endorsed by the Organization.

One of the problems of a publication which deals with such a topical subject is that it quickly becomes outdated. I see this paper, therefore, merely as the first of a series which we will publish on the use of computers in agricultural project analysis, releasing additional material as and when new programmes and guides are completed.

Any comments on the material or suggestions which could contribute to the greater usefulness of the paper would be most welcome and should be addressed to the author.

Cedric Fernando
Director
FAO Investment Centre

TABLE OF CONTENTS

ANNEXES

I. INTRODUCTION

A. General Outline

In most developing countries, governments feel the need to improve the situation through a number of specific actions. These usually start with the formulation of a group of proposals, followed by the study of the means necessary to implement them, and the evaluation of results expected. Once such a programme has taken shape, it becomes what is usually known as a "project"; the project in turn becomes an instrument for deciding on investment for the government of the country and the external sources of finance usually invited to participate to form an opinion on the merits of a proposal, and therefore to decide, where appropriate, to go ahead.

The preparation of such project documents for economic and financial evaluation requires that the project be represented by quantitative models on which analysis will be made. Experience shows that the preparation of the models presents many difficulties from the point of view of methodology, data availability, and the burden of making the calculations. The use of computing tools makes the work easier and improves its quality.

B. Working Background

The object of this report is to present the results of instrumental research carried out by the author as part of his professional activities as a project analyst at the Investment Centre of FAO (Food and Agriculture Organization of the United Nations), Rome.

The FAO Investment Centre includes the FAO/World Bank Cooperative Programme (in which the author is working) and the Investment Support Programme, which cooperates with other financing organizations, particularly IFAD (International Fund for Agricultural Development) and the regional development banks (African, Asian, Inter-American).

The Centre has about a hundred professional officers - economists, agriculturists and various specialists (irrigation, credit, animal husbandry, fisheries, forestry, etc.). The Centre's activities consist of the fielding of missions and the preparation of reports to assist developing countries desirous of implementing agricultural development projects, and, with financial contributions from the organizations mentioned above, to prepare documents which will serve as a basis for decisions on financing.

The operations concerned take the form of "projects" and the role of the Investment Centre is to assist governments, first of all in the identification of appropriate projects, and then in their detailed preparation. In the context of this report, the word "project" refers to investment projects (the word "investment" being taken in the widest sense) and will not cover other types of projects such as study projects or technical assistance projects, which are not usually the subject of the same kind of quantitative analysis.

A project takes the form of a report which describes a proposal for well-defined agricultural development operations, usually to be carried out over a period of four to five years, in order to meet certain objectives. The project document provides details of activities envisaged, both from the technical point of view and from the point of view of cost and organization of activities, and also it gives an analysis of the viability (economic and financial) of the project; this feasibility study will enable the government and the financial institution concerned to form an opinion on the merits of the proposal, and to decide upon its implementation.

A project is thus different from a plan, which establishes a general structure of development lines for the whole country, and a programme, which fixes the broad lines of action for a given sector, but without such precise quantification as in the case of a project.

The Investment Centre undertakes a large variety of projects. They can be on a regional or country scale. They can be either strictly sectoral (irrigation, animal husbandry, forestry, specific crops), or aimed at the development of a sector of activities or services (seed production, extension, research, supply of inputs, credit), or designed to promote the harmonious development of a region by tackling the main problems (integrated rural development).

Despite this variety, the broad outlines of the methodology for project preparation remain the same. One part of the work involves the preparation of quantitative models and the making of a number of calculations. These calculations are still frequently done entirely by hand. The computing tools presented here - some of which have been developed by the author - have the double objective of offering the analyst an improved methodological framework for his work, and freeing him from the burden of calculations.

C. Summary of Contents

Chapter II introduces the methodology for preparing agricultural projects as practised in the FAO Investment Centre, with a presentation of the different stages in the life of a project, and a more detailed description of the work corresponding to the preparation stage. This is followed by a discussion of the types of models used for analysis.

Chapter III explains the necessity for data processing tools. These tools, perceived initially by the user as a simple instrument enabling him to lighten the burden of certain material tasks (calculations, printing of tables), has also proved to be a means of improving the quality of work. It therefore has a methodological impact.

Chapter IV describes in detail the results of the instrumental research done by the author, presenting the fundamental mechanisms for calculation and the basic concepts to be used in the concrete implementation of programmes, whether general or specialized (animal husbandry projects). The chapter ends with an analysis of the advantages and limitations of the approaches proposed, and suggests the functional specifications of an integrated system of programmes, a system which does not yet exist.

Chapter V gives concrete information on the implementation of programmes, using the different approaches.

Chapter VI presents, in conclusion, examples of actual use of the programmes, with illustrations, drawn from the author's experience of the different benefits - particularly methodological ones - which have already been obtained from their use, and the problems encountered. There is also a discussion of how installation of these programmes in developing countries opens up possibilities in regard to the preparation of agricultural projects.

Finally, there are annexes consisting of documentation on some of the programmes developed by the author and the equipment used, and also a bibliography giving a short list of methodological reference works, references to programmes developed by other authors (the World Bank in particular), and the references for the studies or project reports presented in the text as examples.

II. METHODOLOGY FOR PREPARATION OF PROJECTS

A. Project Cycle

The project cycle is the series of stages through which a project passes, from the initial idea to preparation, and then execution. It may be considered that the project cycle is broken down into four major phases: identification, preparation, appraisal and execution (ref. 15).

Identification is the phase from the first idea for the project to a general proposal for financing. The project idea could have different origins: it could be in the national development plan, it could be proposed by a government service, it could arise from overall economic studies or reviews of the agricultural sector made by international organizations such as the World Bank or FAO.

For the initial idea to be transformed into a project (at the identification stage) it should be in line with the priorities of the government - which should give some indication of its a priori support for the suggestion - and it should have a sufficiently sound technological basis, for example availability of water and the existence of a recognized site where an irrigation project could be envisaged.

The identification process consists either in preparing proposals, or in analysing specific proposals already submitted, and recognizing that these proposals seem at first sight to be valid, and seem to be the best way of achieving the project objectives. The selection between different possible solutions is generally made at the identification stage, and the concept of the project will not usually be seriously questioned at later stages of the project cycle.

Proposals will be considered mainly under three essential aspects, which are technical feasibility, economic feasibility (including the existence of markets for the products) and the organization of project implementation.

At this stage, the scope and location of the project will also be defined: delimitation of the relevant geographical zone, reasons behind the choices which will decide the relative importance of the different elements of the project.

The identification phase of a project is certainly the most decisive as regards the soundness of the proposals; it is also the most difficult to grasp or codify in a series of guidelines.

Calculations made at this stage for the analysis of the project are usually limited, and in fact the computing tools described in the present work were developed for meeting the needs of the following stage.

Preparation of the project is intended to develop, complete and confirm the proposals made at the identification stage. A second objective would be to make some progress in solving the problems which became apparent at the time of identification.

The project proposal should be sufficiently detailed to permit confirmation of the viability of the project on the technical, economic and financial, and organizational levels. At this stage, the results of studies requested in the identification phase will be available in order to provide all the necessary details.

The result of the preparatory phase is the feasibility study (preparation report), on the basis of which the project will be evaluated by the government and the relevant financing organization.

References to different guidelines prepared by the FAO Investment Centre on project preparation (refs. 9 to 14) and the basic, necessary data (ref. 16) will be found in Annex 7.

The following sections of this chapter discuss in greater detail the process of preparation, and describe the methods used for economic and financial analysis.

Project appraisal, usually made by a team from the organization financing the project, will be a systematic checking of the preparation, leading to the formulation of an independent judgement on the advisability of financing the project.

It is unusual at this stage to encounter problems with regard to basic technical parameters or standards. The aspects studied at this stage are mainly concerned with the proposed organization and administration of the project, and its financing arrangements. Finally, a number of decisions with political implications (prices, subsidies and taxes, responsibilities, recovery of project costs, etc.) will seldom be taken before this stage.

The appraisal report will serve as the basis for negotiation and conclusion of a loan agreement between the financing organization and the government.

The execution of the project will come later, as the government responsibility, in the framework of the administrative organization defined in the preceding stages. The financing body will control the use of funds and the proper application of the credit agreement, usually by the periodical fielding of supervision missions. The monitoring of the project once it is under way, which ideally should be done by a team independent from that responsible for execution, will provide an opportunity to measure its effects and possibly to identify new problems, and will provide the data necessary for the preparation of a termination report. An ex-post evaluation should in principle always be made, but this will only be possible if sufficient information is available on the situation as it existed before the project (or at least at the time when it started).

B. The Preparation Stage: Procedures and Problems

As explained above, the preparation stage is the one intended to demonstrate the feasibility of the project in the three complementary fields: technical, economic/financial, and institutional.

The most difficult problems arising in the preparation stage are those relating to the level of technical detail required, or acceptable, for such a feasibility study. Whereas, for example, engineering studies for a main road will have to be made according to precise and well-defined standards, the standards for the level of information necessary for the preparation of components of agricultural projects are much more flexible and depend on many factors. For example, the level of study acceptable for a rural road will depend on the competence and experience of the services responsible for the construction of this type of road, on the importance of this component in the whole project, on the more or less difficult nature of the terrain, etc.

In the purely agricultural field, estimates concerning crop yields, possible increases, and consumption of inputs will be made on the basis of research on the spot and results already obtained by progressive farmers in the area. Lastly, it will be necessary to make a value judgement, a possible source of uncertainty or argument.

Similarly, in the economic and financial field, problems can arise in regard to information concerning, for example, the structure of farms, the importance of non-agricultural income, prices, markets. Here again, the level of precision required will not be the same, depending on the homogeneity of the project area, the type of development proposed (for example, irrigation or more extensive rainfed crops), and the "weight" of such and such a component in the whole project.

Lastly, in the organizational field, a number of choices depending on political contingencies or value judgements will, in many cases, have a direct bearing on, for example, certain estimates of staffing requirements for the project, or administrative infrastructures, or means of transport.

All the above considerations are intended to explain that - at the level of detailed quantification in the preparatory stage - many elements of a project are defined by value judgements which are always liable to be contested. The work will progress by a series of iterations between the three fields of analysis, particularly between the technical and the economic/financial. This point is very important and will be taken up again in Chapter III, where it will be explained how the use of computing tools can improve this process.

The need for these iterations will be all the greater where the data basis is weaker, but they will always be necessary. An agricultural project in a developing country cannot be studied like, for example, an industrial project in Europe, where the different parts of the study (technical, financial, economic) will often be carried out successively in time, possibly by different teams.

C. Basic Tools of Analysis

1. Choice of a General Method

Without taking the organizational aspects of the project into consideration for the purposes of the present work, let us see how economic and financial aspects in particular are analysed.

Financial analysis concerns individually the agents involved in the project; for example, agricultural producers, an official organization for the management of an irrigated area, the Government. Analysis of the effects of the project on these agents is made in "financial" terms, that is to say, at real market prices (present prices or, where appropriate, projection of prices in the future).

Economic analysis of the project, on the other hand, is made from the point of view of the nation as a whole. The standard method used by international financing organizations like the World Bank is that of shadow prices (see Little and Mirrlees, ref. 1, Squire and Van der Tak, ref. 5, and Price Gittinger, ref. 3, and also the other reference works indicated in Annex 7).

The difference between financial analysis and economic analysis lies in the nature of the variables taken into account and the type of prices used, financial prices being replaced by "economic" prices (taxes and subsidies, which from the viewpoint of the country as a whole are only transfers, will be eliminated from the prices, and the price of imported or exported products will be calculated in relation to world market prices, used as a reference value).

Apart from prices used, the financial analysis will usually cover a producer, that is to say a farm, and economic analysis will cover the whole project.

Criteria used for the economic analysis are the present value of the project and its rate of return (ROR). The criterion of economic return is one of the fundamental criteria for decision on a project. Financial aspects, on the other hand, can always be corrected - at least in theory - by price controls or subsidies. The purpose of the financial analysis is to show that the project - justified from the point of view of the community by the economic analysis - is sufficiently attractive to the potential participants to ensure their support of the proposed programme. The evaluation criteria will be not so much the rate of return, but the overall increase in income and higher rewards for work. In the case of credit, the analysis should also make it possible to check the ability of borrowers to repay their debts.

The analysis will look at the future situation of a participant in the project, compared with his present situation. In the case of economic analysis, changes made to the production system will be evaluated in comparison with what would have happened without the project: thus two projections are compared, with and without the project.

Research under way in the World Bank on the possibilities of improving the analysis and including in it social factors (distribution of income among social classes or use for consumption or investment, for example - ref. 21) is based on the use of reference prices intended to reflect the value of these factors. The method has not yet become current practice. It would not be incompatible with the tools presented in Chapters IV and V, since prices are exogenous data, provided by the user.

Moreover, the effects method (Prou and Chervel, ref. 8; see also refs. 2 and 22), which would require different tools, is not used in current practice for the preparation and appraisal of agricultural projects on which the Investment Centre is working. This method defines the benefits of the project in terms of internal value added, generated by the project itself and by the internal production of factors of production used by the project. Although the effects method is popular in many French-speaking African countries, it has received little attention from international financing organizations. The author has not had the opportunity to study the use of data processing tools intended to implement this method, which is therefore excluded from the present work.

2. Types of Models Used

a. General Characteristics

An agricultural development project aims at expanding production, whether by the addition of new production units identical with those already existing (for example, cultivation of new areas), or by improvement of productivity through the use of a new combination of factors of production (for example, use of improved seeds, or larger quantities of fertilizer), or, more generally, by a combination of these two mechanisms.

Primary project activities - the project investments - will be undertaken during an initial development period. These activities will later have their effects during the so-called operational period. These two periods together will form the total period of analysis, also called "project life". This duration is in practice fixed at twenty years for most projects, except for those taking a particularly long time to bear fruit (large-scale irrigation infrastructures or forest plantations, for example), in which case the period may vary from 30 to 50 years.

Project analysis consists in comparing the expenses and receipts - corresponding to the different activities of the project - occurring in different periods or varying in time. The project will therefore be represented by a number of time series of values corresponding to the different components of the project (see Brown, ref. 4, and Price Gittinger, ref. 3).

Depending on the nature of the project, this representation will be at three levels:

- The level of elementary production activities, that is to say, usually crops (but also, for example, livestock). The crop budget will contain all quantitative information on the combination of factors of production (work, seed, fertilizer, irrigation water, etc.) leading to a certain production (yield). The changes in values in successive years will be the representation of the changes in cropping practices and productivity under the effect of project activities.

- The level of the producer, that is to say, of the farm. The farm budget will represent all the crops and other production activities, in the light of the development of the area under each crop (or the numbers of animals, in the case of livestock-raising). It will also present the investments foreseen at farm level, and generally all expenses or receipts which have not already been taken into account in the crop budgets. The farm budget will be the tool used for financial analysis of the project from the point of the view of the producer.

- The level of the project, that is to say, the budget of all the costs and benefits of the project, for overall analysis. One part of the costs/benefits is derived from directly productive components of the project (farm models), and the other concerns all the other components, such as, for example, infrastructure, administrative services of the project, etc.

In the construction of these models of the different project elements, several techniques may in principle be used. In particular, a choice must be made between simulation and optimization models and between determinist models and the use of stochastic variables. Finally, in the presentation of time series, the question of the time scale to be used arises. These different points, and also the corresponding choices adopted in practice for most projects, are discussed in detail below.

b. Simulation or Optimization

Apart from the fact that the material burden of calculation excludes in practice the use of optimization models (linear programming type) when the calculations are done manually, the use of such models may perhaps be justified sometimes. The most typical example is that of the use of a linear programming model to calculate the cropping pattern for a farm (or region) in the light of constraints on availability of factors of production, or markets for the products. In practice the following problems arise:

- The model is classically static, whereas the reality to be represented is constantly changing; but it is always possible to use the model for one situation corresponding to a given stage of the project.

- The model usually requires more data than a simulation model of the farm budget type; the analyst can seldom resist the temptation to refine it by multiplying possible activities and constraints.

- The model is difficult for non-specialists to understand, particularly for those officers involved in making a decision on financing who have not been initiated into this technique.

- The value judgements governing the fixation of certain parameters of a farm model - in the context of agriculture in developing countries - are often difficult to quantify in the forms required by the optimization model.

- The model is often excessively sensitive; slight changes in the data will make the solution vary considerably; and data is not usually very precise.

Because of all these problems, experience has shown that in most cases the cropping pattern proposed by experienced agriculturists would not differ significantly from that obtained by such a model. The use of optimization models can sometimes be justified at the time of studies preceding the final phase of preparation of certain projects (for example, in the case of irrigation projects, with limited availability of water, in very diversified agricultural areas), but in the author's experience such models are mainly useful for supporting certain technical judgements which will later be integrated in the simulation models. In the context of the projects discussed here, the optimization model should in such cases be considered as a tool of analysis of sensitivity of a complex model rather than a model for calculating an optimum.

c. <u>Determinist or Stochastic Models</u>

Current practice consists in attributing to the input data of the models values determined "from outside", that is to say, fixed. This is the case, for example with yield and input consumption forecasts for crops. But it is obvious that such data represent only averages and that the real figures will depend on many factors not represented in the budgetary simulation model; in particular, yields could depend heavily on climatic conditions.

It could, therefore, be tempting to incorporate this variability into the models, for example, by generating stochastic series of yields based on past series of observations of yields in the light of climatic data. But because of a number of problems, this kind of analysis is hardly ever used in practice:

- There are seldom enough observations available to establish a statistical correlation between the yield of a crop in a project area and the main climatic factors.

- Still more unusual would be the availability of data to establish the independence of (or the link between) the different variables for which it is desired to generate a series of data. In the absence of such information, it is not possible to construct a model in which there would be more than one stochastic variable, and there are hardly any projects which would fit into such a limited framework.

- The burden of calculation (the model would have to be calculated repeatedly to allow for a proper study of the distribution of results) makes it very clear that this kind of analysis is impossible without an appropriate data processing tool.

The following example, taken from the author's own experience, is a good illustration of the problem. A project was proposed to develop the floodlands in the lake area of Mali (ref. 45). The lowlands are flooded each year by the River Niger and cultivated with sorghum when the floods recede. The flooding and withdrawal of the water are traditionally controlled by earthern dikes built and opened each year by hand. The project consisted in improving control of floodwaters by constructing permanent dikes with a control structure (gate with stop-logs), and by digging canals to allow the sheet of water to spread over the lowlands. The flooded area varies according to the levels reached by the river. Flood statistics over 50 years and precise topographical maps have made it possible to establish the contours of the area flooded (and therefore cultivable) according to the level of the flood, and therefore a distribution of the probability of cultivable area. All data was therefore available to introduce this variable in the form of a stochastic variable. But the sorghum yield also varies considerably in different years, and even if the distribution of the yields was (approximately) known, there was no series of observations of yields to facilitate the study of the correlation (or absence of correlation) between the yield and the flood level. It was therefore impossible to introduce the two variables, area and yield, in the form of stochastic variables. A sensitivity analysis allowing - with a simple determinist model - for the testing of different combinations of values of area and yield led quite easily to the formation of an opinion on the validity of the proposition.

In conclusion, it may be said that a determinist simulation model, which enables an evaluation to be made quite simply of the effects on the results of the model of changes in the hypotheses regarding the data, usually provides sufficient information for the end in view, that is to say, the making of a decision on a development programme. There is a danger that the use of stochastic variables, in the absence of sufficiently reliable data on the conditions of variation of these variables, will provide only an illusion of accuracy, and no better information than that which could have been obtained from a properly conducted sensitivity analysis with a purely determinist model.

d. The Time Factor

In advanced agricultural societies, such as in Europe, the development of a farm often takes the form of a qualitative leap from one kind of production to another, corresponding, for example, to the realization of an investment. The operation can be accurately described by indicating the present situation, before investment, and the future situation. Each of the two situations will often be considered as static. It will be represented by a data series on an annual basis, with a more detailed analysis of the spread of costs and benefits over the year, if appropriate.

An agricultural project in a developing country, on the other hand, is usually accompanied by a more or less slow advance in the technological level of the farmers concerned in the course of several years. The overall programme, moreover, will not affect all the beneficiaries in the first year. On the contrary, their "entry" into the project will be staggered over a period of several years. Such a development plan obviously cannot be represented by two static situations (before and after), but will require the quantification of all intermediate stages. The different parameters of the project (for example, areas and yields of a crop) will therefore be represented by a time series of data, projected over several years. The time scale will be fundamentally the year, but it will sometimes be necessary to make a more detailed analysis of certain factors within the year, such as, for example, the use and availability of labour or irrigation water. In this case, the analysis will not necessarily have to be made for all intermediate years; the year will in any event remain the fundamental time scale.

Another essential aspect of the definition of models is the distinction between the "before project" situation and the "without project" situation. A project is very often situated in a non-stabilized context, that is to say that the agricultural situation of the area concerned is in a state of evolution, and that this evolution would be pursued at a certain rate in the absence of the proposed project. The object of project analysis is not to evaluate a production system in its entirety, but to evaluate the effects of a modification in this production system. The analysis therefore covers additional costs and benefits of the project compared with the situation without project. When the situation without project is not constant, it is therefore necessary to represent the parameters of the project by two time series of data, one corresponding to the projection without project and the other to the projection with project. When the situation without project can reasonably be considered constant, it can be represented by one value alone. Therefore the standard method is to take year zero of a time series to represent the without project situation, and the other values (year 1 and following years), to correspond to the projected situation with project.

3. Problems Encountered in the Construction of Models

a. Quality of Data

The fundamental problem in the construction of models is obviously the availability of sufficiently detailed and reliable data. In the context of developing economies, which do not usually have elaborate statistical systems, the problem will be particularly acute. There will rarely be statistical surveys conducted according to the rules, or sufficiently trustworthy results of research to be extrapolated with any precision outside of experimental stations. As already mentioned above, much of the data will finally be "estimates", that is to say, the result of a value judgement by the technicians responsible for the study.

It is of course always possible to improve the availability of information by making a number of surveys and studies for the preparation of the project. But here again, value judgements will have to be made to decide upon a reasonable compromise between the cost (in time as well as in financial resources) of such studies and the benefits that may be expected from them.

b. Physical Burden of Calculations

Even with very simplified models, the quantity of data to be handled grows very quickly with the number of crops, the level of detail needed for each crop (number of categories of costs), the number of categories of investments, and the number of farm models needed to represent the project.

Take, for example, a simple project in which there are only five crops. On the assumption that the process of development takes five years at the level of one parcel, that all parcels will come into the project in four years (which is very short, since this combines progress in the cropping plan at the level of one farmer with the gradual entry of farmers into the project), and that the situation without project should be considered as constant, and therefore represented by the year zero, the tables - on an annual base - will have ten columns.

If each crop is represented by only five parameters (area, yield, labour, seeds, inputs), there will be 25 data lines for the model, plus a certain number of lines for investments, results of crops (production and inputs, in quantity and value), plus the lines used for other calculations such as credit operations, or the value of the working day. Even a very simplified model will therefore seldom include less than about fifty data lines and results, and a project represented by five farm models, and with a number of investment components and operational costs, will very rapidly be reflected in several hundred data lines, and therefore several thousand values to be added and multiplied.

If the situation without project is not constant, the amount of data is multiplied by two, and the number of results by three (with project, without project, difference).

In practice, many projects will be the sum of several components; there will, for example, be several ecologically different sub-zones, or else one zone with rainfed crops and another with irrigated crops, the combination of animal husbandry with crop-growing, a programme for rural roads and water supply, etc.

The quantity of data therefore grows very quickly, even if one keeps each component at a low (even simplified) level of analysis.

c. Heterogeneity of Models

Even if the lack of standardization found in practice in the presentation of models is not in itself to be condemned on the methodological level - it is, on the contrary, altogether desirable that the project analyst should have the freedom to express his judgement and adapt his models to each particular case - the heterogeneity which results from this presents certain practical disadvantages:

- It will be more difficult to use the study; it will take more time to find whatever specific information is being sought in the report; it will be more difficult to follow the analyst's methods.

- It will be more difficult to control the quality of the study or to check the validity of results obtained.

- It will sometimes be difficult to compare the results of analyses presented differently from one project to another; the data and the results will have been aggregated in different ways, the synthetic indicators will not be the same, and therefore impossible to compare directly.

- Certain underlying assumptions are often implicit, since they appear obvious to the author of the report, because of his background knowledge of the case; a reader without the same familiarity with the subject will sometimes have difficulty in reconciling the results with the data provided.

III. WHY COMPUTING TOOLS ARE NEEDED

A. Methodological Interest

The use of a computing programme to construct project analysis models can have an impact on the methodology used, even if the programme is apparently "neutral" and very flexible. The tool will be a means of standardizing a certain methodology, of developing calculating techniques - even correcting errors - of increasing the number of variables taken into account, and of making better use of the models and sometimes validating data, thanks to the consistency of the models.

1. Standardizing the Methodology

The programmes described in the following chapters all follow - despite the diversity of the approaches used - an identical basic methodology: determinist models of budgetary simulation, based on an annual time series of data. The use of these programmes therefore presupposes the adoption of this fundamental choice.

The availability in the programme of predefined functions to carry out this or that type of analysis will be a strong incentive to a certain standardization. A typical example is that of methods employed for sensitivity analysis of models: systematic use of the technique of "switching values" described in the following section, has become current practice in the FAO Investment Centre and the European Investment Bank, for example, due to the use of programmes making these analyses in a simple way coupled with other operations that analysts were accustomed to using (rate of return).

Standardization can also cover the way in which the data is organized in the models and the results presented, which will in turn facilitate their use by people other than their author. The reader familiar with this presentation will know exactly where to find any piece of key information and will understand exactly the type of analysis that has been made.

For the project analyst, a certain standardization makes it easier to organize the data collection. This aspect will be more or less marked depending on the more or less open character of the model (see Chapter IV, the different approaches used for construction of the programmes).

Conversely, the absence of specific mechanisms in a programme to treat certain aspects of the analysis could reflect either the lack of priority accorded to these aspects, or the absence of decision on the method to be used. A typical example is that of taking inflation into account in the analysis of cashflow projections in a farm model. The models are - in the context of the work under discussion - prepared on the basis of prices in constant money, that is to say, only relative changes in price are taken into account. But it is obvious that, when there is a loan at a rate not indexed to inflation, the borrower takes this inflation into account when making his decision, since it lessens the burden of repayment in real terms. Experience shows that - depending on the project, the rate of inflation, the credit policy - the methods used to analyse this aspect of the problem can be very different, ranging from taking absolutely no account of the phenomenon, to the calculation of complete projections in current prices, not to mention deflation of the debt service (the other variables being in constant money) or the use of a "reference" interest rate. Each method has certain disadvantages: the projection in current prices is difficult to interpret, since the effects of inflation mask the effects of the project, and where inflation rates are high and fluctuating, projections are, to say the least, hypothetical. The deflation of the debt service, which also means projecting the inflation rate over the period of repayment, has the disadvantage of not presenting the values in the same monetary terms as the rest of the models (all the variables of the model are in constant money, except debt service, which is corrected by a deflation factor). The use of a reference interest rate - which, depending on the circumstances, will be the difference between the nominal rate and the inflation rate, or else an "economic" rate reflecting what should be the real cost of the loan in the eyes of the analyst - is, from this point of view, still more remote from the financial reality of the model.

None of the programmes presented in this work include specific functions for dealing with the above problem due to uncertainty on the methodology to be used. In the same way, any organization which prefers one method to all others could make it easier to use by incorporating specific arrangements in the calculation programmes.

2. Improving Techniques

The use of a computing tool will serve as a vehicle for the introduction of new techniques of analysis; it will even, in some cases, replace mistaken practices by a correct method of calculation. Each of these points is illustrated below by an example drawn from experience: the introduction of "switching values" as a special tool of analysis and the calculations of aggregation ("phasing").

Switching values: From the point of view of profitability analysis, a project becomes a series of data lines representing the values - year by year - of project components, that is to say, the different elements with a cost or producing a benefit. The algebraic sum of these lines represents the net value of the project. The project will be characterized by the present value of this balance for a given discount rate, and by its rate of return.

The standard technique for sensitivity analysis consists in varying some of the project components by a specific percentage (10%, 20%), and calculating the new values of the present value and the rate of return.

Example: (for a given discount rate)
present value - 150,000 ROR: 14.5%
Investments + 10%, value production - 10%:
present value 120,000 - ROR: 12.2%

This method, formerly very frequently used, is being increasingly abandoned in favour of that of switching values. The switching value of a variable is the percentage rate by which all the values of this variable must be varied - the other variables remaining constant - to cancel the net present value of the project, or - which is the same thing - to bring the ROR to the value of the discount rate.

A farm model, for example, is summed up by the following six variables: value of production (gross), other income, investment, production costs, family labour cost, and increase in production value in the absence of the project because of variations in prices (variations already taken into account also in the with project situation). For a financial analysis, the family labour cost appears as receipts, since the total labour cost has been counted in the production costs at the price of a wage-earning labour force. Without project costs and productions are represented by the values of the year zero of the variables. The balance for year zero, projected over the analysis period, is called "without project balance". The result of the calculation of profitability will be presented as follows:

Rate of return: 30%
(discount rate: 12%)

	Present values	Switching values
Production value	41,076	-15.1
Other income	5,191	-119.2
Production costs	-22,078	28.0
Investments	-8,602	71.9
Family labour cost	2,173	-284.8
Without project increase in production value	-3,732	165.8
Without project balance	-7,841	78.9
Net balance	6,187	

Switching values make it possible to appreciate - in one table alone - the relative sensitivity of each of the project variables. In the above example, it is clear that the model is very sensitive to changes in production value (yield multiplied by price), and also sensitive - but to a lesser extent - to production costs, and far less sensitive to the other parameters. This type of analysis therefore makes it possible to identify very easily the important variables of a model, and to check whether the precision of the data is compatible with the degree of sensitivity of the variable.

It is of course perfectly possible to make such an analysis without using a computer (the calculation being limited to rules of three on present values). But experience has shown that habits change slowly and, as already mentioned above, the systematic use of this technique has developed within many organizations through the use of a calculating programme. The programme thus becomes a training instrument for diffusion of a new technique.

Aggregation calculations: It is quite usual to have to make calculations of the following type:

The yield of a crop is 1 t/ha. Under a project, it increases in four years to 3 t (for example, 1.5 t/ha the first year, 2 t the second, 2.5 t the third). If the project develops 100 ha in year 1 and 200 more in year 2, production will be equal to:

	Year						
	0	1	2	3	4	5	20
Yield	1	1.5	2	2.5	3		
100 ha in year 1	100	150	200	250	300	300 -->	
200 ha in year 2	200	200	300	400	500	600 -->	
Total	300	350	500	650	800	900 -->	

The series of yields should be lagged by one year, to be multiplied by the number of hectares in year 2 (hence the term "phasing" or "building-up" for this type of calculation). The supplementary value of the production is obtained by subtracting the value of year zero from the values of each year of the time series. This type of calculation is described in more detail in the following chapter, section C. It may be seen there that, when the without project situation is not constant, the calculation must be made in a different way, depending on whether the result is required in total values or in incremental values (compared with the without project situation): it would be a mistake to pass from one result to the other, subtracting the value of year zero. Experience shows that this type of mistake easily slips into calculations, and that the introduction of a computer programme has made it possible both to draw attention to the problem - of which more than one analyst was not fully aware - and to correct the error, by providing a practical means of making the calculation correctly.

3. Increasing the Number of Variables taken into Account

By offering the project analyst an organized framework for calculation, by freeing him from the burden of calculations and related problems of table editing, the computing tool makes it possible to develop more detailed and more complete models. For example, more crops can be taken into account (in a project with vegetable gardening, instead of simplifying by representing only three or four typical productions, perhaps twice as much will be represented, which might well correspond to a more realistic representation of the problems), the different inputs necessary for production can be more clearly shown, an overall analysis can be made of, for example, needs and supplies (seeds, fertilizers, for example), and an analysis of availability of labour or irrigation water, etc., can be made more precisely (over shorter periods of time).

The most typical case is that of a better understanding of the without project situation. As mentioned above, economic analysis covers additional benefits corresponding to additional costs, and therefore the difference between the two projected situations: the project, and the without project situation. The without project situation is frequently considered as being constant and identical with the before project situation, represented by year zero on the time scale. This could be perfectly justified in some cases, but experience shows that it is not always so. It is undeniable that, if the situation is fully taken into account, twice as many calculations will have to be done (two projections instead of one). Inadequacy of data can sometimes justify - as an approximation - the hypothesis that the situation without project is constant. But clearly one is often forced to make a value judgement, which is certainly influenced by the material implications of the volume of work, even if this is sometimes unconscious. The calculating tool has two advantages: it frees the analyst from the burden of calculations and - to the extent to which the programme has been conceived in this way - it obliges him to explain his assumptions, in particular those concerning the without project situation. A particularly striking example is that of livestock projects analysed on the basis of a demographic model of the herd (see IV.F). The herd is defined by its composition and by a number of parameters (natality, mortality, sales, purchases). The project will lead to changes in these parameters (for example, a vaccination programme will lower the mortality rate), which in turn will lead to changes in the structure of the herd and its productions. It is quite common to limit the analysis to such a projection, whereas the calculation shows that very often, even if the herd parameters are constant, the demographic structure is such that the herd is constantly changing, it is not stabilized (this is usually the case, for example, with the Sahelian herds). The right computing tool can easily make the two projections, which will sometimes give unexpected results, since the calculation is sufficiently complex for it to be difficult to foresee the results in the light of the data. Another example is that of the forestry project described below (point 5), in which the data processing tool takes into account not only the forests managed by the project, but also the evolution of natural forests, which would certainly not have been possible without such a tool.

This impact of the use of a computing tool on the level of detail of the analysis will also be reflected - and this could in some cases considerably improve the quality of the work - in a better understanding of the problems in the stage of project identification. It was mentioned above (II.A) that calculations made at this stage are usually very limited, and that the broad conceptual outlines of the project decided at this stage will seldom be questioned later. It is clear that, when data is not yet available, or when certain choices of economic policy have not been made or some development options have not been taken, the analyst hesitates to embark on the construction of detailed models of a more or less hypothetical nature: he is discouraged by the disproportion between the work required and the results expected. Here again, if the work required is considerably lightened by the use of the computer, he will be able more easily to build analysis models which will help him from this stage onwards to identify certain problems such as, in particular, the probable sensitivity of the results to such and such a little-known parameter, such as yield, price, or outlets for a certain product, or else the non-availability of such and such an important datum. This will be reflected in better orientation of work to be done between identification and preparation.

In the same way, certain fundamental options are taken in a more or less explicit way, without alternative solutions having really been envisaged. One might, for example, embark on the study of an irrigation scheme, based on substantial investment leading to complete control of the water without having really asked the question of whether a smaller investment, leading to partial control of the water and a less ambitious cropping programme, would not be more interesting in the end. Here again, the possibility of building models easily - even based on partly hypothetical data - will in some cases be the favourable element that will make all the difference, and will lead to a deeper analysis of the proposal.

4. <u>Making Better Use of the Models</u>

One of the major advantages of mechanized calculation of models is that their sensitivity to changes in hypothesis can be more fully explored; that is to say, it becomes easier to reply to questions such as: What will be the result if I change the level of this yield, if the outlet for this very profitable crop is limited to so many tons, if the prices of such and such a product do not increase as expected?

It is clear that, as soon as a project becomes a little bit complicated, the material constraint of making calculations means that this type of analysis is very limited.

The following example is drawn from the author's own experience during the preparation of an agricultural credit project in Paraguay (ref. 48). It was anticipated that the major demand would come from the area of the country where soybeans were grown, to finance the destumping of land already cleared and the purchase of tractors, harvesters, and other equipment. The project was analysed through nine farm models, illustrating different investments.

Since credit was granted on the basis of farm development plans prepared by agronomists from the Agricultural Bank, it had been supposed that the project would lead to not only an increase in the area under cultivation, but also a certain increase in productivity. During the routine discussion of the report before its publication, the Review Committee considered that this hypothesis of increasing productivity could be questioned, and that the models presented did not show clearly the respective impact of the increase in area and of the increase in productivity, on the project results. Answering the question, "What would the project results be in the absence of any improvement in productivity?" meant complete recalculation of the project, that is to say nine models, and their aggregation at project level over a period of three years. If the project had been prepared manually, the question would probably have remained unanswered. Since the data and the definition of models were stored in a computer system, it took only - once the agronomist had prepared the new data - half an hour to modify the yield and production cost data, to recalculate all the models and the whole project, and to obtain new results which were presented in the report as an additional element of sensitivity analysis.

5. Validating the Data

By their very nature, the simulation models present a certain internal consistency which means that one cannot use just any data. If a crop budget or a farm model shows negative results, while in the field it can be seen that the relevant activities are making good progress, it is either because the model is not representative or because the data is incorrect. Conversely, if the results correspond to what one knows of reality, this is an indication that the assumptions are probably correct.

The part played by the consistency of models in the validation of data will be more or less important depending on the nature of the models and the projects. The two examples below - one concerning livestock-raising and the other a forestry project - illustrate cases where this role is particularly marked.

In the case of animal husbandry projects, the mechanics of calculating the demographic models of herds means that the different data to be observed (structure of herd by class of age, rate of fertility, mortality, sales) are directly interlinked. If, for example, there is survey data on herd structure and fertility rates and statistics on demographic growth in the past, the model will allow for adjustment, within the limit of precision of available information, of the different parameters of rates of sale, to obtain a herd growth rate compatible with the available statistical data. On the other hand, if this leads to the use of rates of sale which, after checking with the livestock-raisers, prove to be unrealistic, this will mean that other data is invalid (fertility or demographic growth statistics).

In other cases, as in the example of the forestry project described below, the consistency of the model has been justified by the results of the sensitivity analysis. It represented a forestry project in the mountains of Nepal (ref. 47) intended to promote among village communities the establishment of new plantations and the regeneration, and later rational management, of remaining natural forests, on the following assumptions:

- The natural forest was divided into two parts, the distant forest and the near forest. For each part, estimates were made of area, standing volumes, growth rates, and average distance from the village (and therefore, time spent collecting wood).

- The projected demand for firewood (in the light of population projections) was supposed to be covered, in order of priority, by the forests developed by the project, by the nearby natural forests, and finally by the distant natural forests.

The model, in the particular case of this project, was not treated with any of the computing tools presented below (they did not yet exist) but needed the preparation of a specific programme. The data used came from a superficial survey in a hundred villages, by the staff of the Nepal forest service, and was analysed by the team preparing the project.

The model showed that, without the project, all the forests would disappear in twenty years, which accorded with other estimates made on the basis of the historic evolution of the area. Also, this model proved to be remarkably insensitive to changes in assumptions concerning some of the parameters on which the information was not very reliable, such as, for example, growth rates or average distance from forest to village. The conclusion was that the data was certainly trustworthy to some extent, and that the conclusions of the model were valid despite the lack of precision in some of the data.

It would have been difficult for the analysis to take into account the impact of the project on the utilization of the natural forest without the use of a computer, again because of the burden of calculation (projections over 50 years, in view of the slow growth of the species concerned). The model showed that, in the long term, one of the major effects of the project would be to safeguard the distant forest (the nearby forest would disappear anyway, even quicker than without the project, because of the protection - initially - of the part to be transformed into a forest managed under the project) which would otherwise have disappeared, and would in time, thanks to the increase in standing volumes resulting from more moderate exploitation, lead to considerably increased production from this natural forest, which was not directly included in the project.

To sum up a "conventional" analysis, based only on a cost/benefit analysis of the areas managed by the project, would have shown very different results, perhaps even inadequate to justify the project.

6. <u>Limitations</u>

The computing tool is only an aid to the normal work of analysis. It allows the methodology to be improved, but only to the extent that the basis for an improved methodology already exists. It makes the use of a greater number of variables and the carrying out of a greater mass of calculations possible; it does not replace the analyst's normal know-how. The tool is and must remain only a tool.

The last chapter (Chapter VI, Section B) outlines the difficulties and limitations which the practice has shown up, difficulties linked to the particular character of any computing tool (the need for checking the data properly or the risk of producing results which will have all the appearances of the necessary validity, but will be incorrect because the data will not have the required qualitites, etc.).

A fundamental aspect whose importance can never be over-emphasized is the following: when the analysis methodology is not clearly established and the choice in the matter is left to the judgement of the analyst, the data processing tool should have the role of facilitating these choices, and explaining them, and not of taking decisions in a way that is hidden to the user. Again, the tool must remain only a tool.

B. Immediate Material Advantages

The computing tool - and we have just seen to what extent it could have an impact on analysis methodology and on the quality of the work of preparation - is often perceived by the user, and justified when a decision is being made to purchase equipment, from the point of view of the immediate material advantages alone. These include the time saved for the project analyst, who finds himself relieved of the material task of making a large number of calculations, and the saving of difficult and therefore expensive work in the typing of tables of figures. These advantages should not, however, be underestimated because of their trivial nature. The fact that they are often the only ones taken into account illustrates their importance.

Another advantage - more potential than real in the present state of development of computer systems - is the possibility of transferring data and models of preparation en bloc to the team responsible for appraising the project. This would save them having to do the work all over again; they could just make the necessary adjustments or corrections in the light of the results of the appraisal.

IV. DEVELOPMENT OF THE INSTRUMENTS: IDENTIFICATION OF FUNDAMENTAL MECHANISMS AND CONCEPTION OF PROGRAMMES

A. Introduction - Characteristics of the Ideal Tool - Summary of this Chapter

The computing tool needed is a programme - or a collection of programmes - that will both relieve teams responsible for preparing agricultural development projects of the physical task of making innumerable calculations, and also improve the quality of their analysis, because the models will be better used and, if necessary, further developed.

The tool should, above all, be fairly easy to use, so that it does not itself create an additional constraint. This easiness should be on different levels: the conceptual level and the level of detailed use. A good programme will be conceptually easy for the user to understand, that is to say, the nature of the operations conducted by the programme will have to be the same as those conducted in the normal course of work; the project analyst should not be obliged to change his working methods exclusively in order to adapt himself to the way in which a programme has been conceived. On the level of detailed use of the programme, it is again important to remain as close as possible to the user's way of working and his normal vocabulary. If expressions or new words have to be defined, it is important to choose precise and self-explanatory words, which do not lend themselves to mistaken interpretation, and are easy to remember. It is also important to adopt, in programme/user communication, a style that will be maintained consistently throughout the programme. These observations may appear elementary, but experience shows that the end is not always easy to attain, and the degree of use of the programme, and therefore its impact on the work of analysis, will often depend to a very large extent on factors of this order.

An important element is the facility of initiation into the use of the programme. With a good programme, it should be possible to learn very quickly (say, in a one-hour demonstration at the most) to do a number of simple operations alone, which already amounts to actual use of the tool. More sophisticated applications will be gradually developed.

It is also important not to have to remember a large amount of information in order to use the programme efficiently, rather than having recourse to a handbook, it would be better to include in the programme itself the necessary amount of documentation, easily accessible to the user.

Since the programme has to be used for a great variety of projects, and to make analyses at different levels of detail, it should be fairly versatile. In particular, it should be possible, if necessary, to build up models gradually, that is to say, to put in only certain data, make certain calculations, add data, correct and develop the model, etc.

Lastly, since the work is carried out in several phases, and in places geographically very distant from each other (for example, a project studied in Latin America, whose preparation report will be finalized in Rome, and later appraised by the World Bank in Washington), it would be desirable that the same computing tool should be accessible at different stages in the work. This raises both the problem of availability of equipment and that of transportability of programmes and data, that is to say, the greater or lesser facility of installing a programme on different equipment.

It will be seen later in this chapter, and in Chapter V, that the reality is still relatively far from the above ideal, and that each programme is the result of choices and compromises: the greatest versatility of use does not always mean ease of use, and the more possibilities a programme offers, the larger it is and the more problems will be raised in regard to capacity of equipment and transferability.

The computing tools presented in this work are, as mentioned in Chapter II, mainly programmes intended to formulate and treat determinist models of budgetary simulation. Even if these programmes can be used in other fields, they have been developed specifically for the preparation (or appraisal) of agricultural projects in developing countries, which is reflected in the way in which the data is organized and the nature of the calculations made. But within this same framework there are programmes conceived in a very different way.

The following sections of this chapter present three different conceptual approaches, which are the table manipulation, the general structuring of the data, and the detailed structuring. The particular case of animal husbandry projects is then presented. The last section sets out the respective advantages and limitations of the approaches proposed, and outlines the functional specifications of an integrated system of programmes intended to come as close as possible to the ideal characteristics mentioned above.

Before presenting these different concepts, which correspond fundamentally to different levels of data structuring, the following section touches on a very important practical problem, that of programme-user communication.

B. Interface with the User

There are several ways of conceiving the interface (that is to say, the liaison) between a computer programme and its user.

The first, historically, is the use of punch cards, corresponding to those of batch processing systems: in order to maximize the use of large-scale systems, the different tasks to be undertaken are put into a queue, the information (that is to say, the data and the instructions for executing the programme) being fed into the system through a punch card reader. This method, still very popular in some fields of application, is not at all suitable for analysis programmes and work, where it is desirable that the analyst should be able to change a datum very quickly, obtain new results, develop his model, etc. This type of work is hardly compatible with the above-mentioned systems, which usually require the transcription of the data on forms (whose format is often too rigid), the despatch of the forms to a punching centre, the return of the cards to the processing centre, and the return of the user to the processing centre after some hours to collect the results, sometimes only to find that mistakes have been made in the cards, which means that nothing has yet been calculated, and so on.

The development of computer systems means that it should no longer be necessary to have recourse to such methods. All the programmes presented in this work, and still used, are so-called interactive programmes, accessible at terminals on time-sharing or monostation systems and written in conversational style: the programme requests, through a message to the terminal, the input of information on the keyboard from the user and reacts correspondingly.

In such a context, the user/programme dialogue can take two different forms, which correspond to very different structures of the underlying programme. The first form is that of the question-and-answer system, with the selection of options presented in the form of a menu; the second is that of communicating with the programme by way of a certain number of predefined key words, that is to say, a language specific to the programme (command language or modelization language). Each of these two approaches has advantages and disadvantages, each has its defenders, sometimes unwavering, and its opponents. Lastly, and whatever the system chosen, one very important aspect of the user interface is the treatment of errors.

1. <u>Menu System</u>

The programme is organized on the functional level in a number of modules, and each module in sub-modules, functions, etc. At each level of the structure, a menu makes it possible to select the required option; when a certain function requires the supply of parameters, or data, the programme calls them explicitly, and the user enters them on the keyboard. The questions and the menus can easily be accompanied by explanations and comments displayed on the terminal screen, facilitating still further the use of the programme.

Such a system has the main advantage of being very easy to use, to the extent that messages are sufficiently explicit. The user only needs to know how to call the programme and start it, and then just let himself be guided. This is very "soothing" for users who are not familiar with data processing and are frightened of making mistakes, the consequences of which they cannot assess.

The disadvantage is that the system can easily become very cumbersome, and sometimes relatively slow, since every stage, every datum has to be explained.

Another problem is the difficulty there could be in "getting out of" a certain module of the programme, if the user realizes he has made a mistake: he will usually be obliged to go to the end of the sequence, then change over to another module to correct the effects of the error. The structure of logical modules should be based on the user's way of working, and should make a "normal" sequence of operations easy, but such a result is sometimes difficult to achieve. Sometimes it is necessary to go back three or four menus in a hierarchy to get out of one operation, and then select three of four successive options to arrive at the next function it is intended to execute.

The menu system is, therefore, all the more difficult to work properly when the programme presents a greater diversity of functions and requires more data. When a system is developed including several programmes - corresponding either to different approaches or to specific problems - it is desirable that the programmes should always be structured in the same way, always using the same vocabulary.

The approach selected for programmes developed by the author and described in the following chapter is based on a systematic separation between data and calculations. As required, the results can be separated from data, or on the contrary can be integrated in the data base. One will always therefore find, in one way or another, the following modules:

1) Enter data.

2) Correct data.

3) Print data.

4) Make calculations.

5) Print results.

6) End.

The first module will always be called first when the programme is launched. It provides an opportunity to enter data on the keyboard, or to retrieve old data stored in the system (on a disc, for example). It will then be possible to call the modules in any order, undefined, until the "end" *module is selected to stop the session. The most serious disadvantage* of this method, which is easy to understand and use, is at the level of data input; for example, if an error occurs in the process of entering data, it is extremely tiresome to have to follow a long sequence of operations - if this happens to be the case - to complete the work of data input, before being able to correct the error by operating module 2 (data editing). It is far more convenient for the user to be able to interrupt the data input, correct his error and forget it, and take up the remainder at the point of interruption.

2. Command Language

Access to the different functions available in the programme is accomplished by typing a key word or a series of instructions, followed by the necessary data. If the data is not supplied, the programme may either attribute values by default, or display a message requesting the data (as in a menu system).

The major advantage of such an approach is its greater flexibility: there is direct access to any function of the programme, by a specific instruction. The work of typing can also be limited by allowing the abbreviation of key words. An experienced user can thus put in his instructions very rapidly, and if all the parameters have values by default *within the programme, it will be enough to specify the values exclusively in* the cases where they are different.

Unfortunately, this flexibility and this rapidity of use have their prices: the user must know the instructions of the "language" and the default parameter values. This is why such systems are often more difficult for the novice, will require more intensive training, and will not be utilized in depth in the absence of substantial documentation (the latter problem can be minimized if the system contains instructions making it possible to call this documentation to the screen in an easy way).

Combined systems could be devised, where a first level of selection of operations to be performed would be made through a command language and where the detailed necessary information would then be called by the programme, using either questions and answers, or forms which have to be completed (or corrected on the screen if the values are already there, whether they are values by default or data previously introduced).

One advantage of the system by command language is that it is easy - from the user's point of view - to pass from the immediate execution of instructions to the programming of procedures, since the definition of a sequence of operations is nothing more than storage, for later retrieval, of the series of instructions which had been provided by direct execution.

This leads to a third possible way of communication with the system, which could be called modelization language.

3. Modelization Language

Command language can be used, no longer directly, in reply to a message from the system, but to write a text (for example, with a standard programme for text processing), which correponds to the definition of tasks that the user wishes to see performed by a programme. The text can later be processed by the programme, and the series of corresponding opeations will be executed. A simple way of modifying the model thus defined consists in correcting the instructions or the data at the level of the source text, then processing it again with the programme. The procedure - although it has the same disadvantages as command language, that is to say, the need for a more intensive training than in the case of menus - can be improved by the preparation of "standard texts" corresponding to the definition of various models, and possibly containing instructions for use, texts which can be copied and corrected to define a specific model.

Among the programmes presented in Chapter 5, some combine the use of menus and text files, other use command language and modelization language.

4. Treatment of Errors

Whatever the system selected for the programme/user interface, the way in which errors are treated should be the same, and this is a fundamental aspect of the qualities of a programme.

The principles are simple:

- Errors made by the user should never lead to the termination of the programme or the loss of data.

- The programme should, as far as possible, test the instructions and the data it receives, and detect logical errors (and particularly when data can only take certain values, test the data by comparison with correct values).

- Messages indicating error should be explicit, not ambiguous, making it possible to easily correct the error.

In practice these guidelines are not so simple to follow: it could be that some implicit hypotheses are not identified and the programme could either halt or give invalid results if the data takes on unexpected values; or else certain messages, which are obvious to the programmer or to some users, appear incomprehensible to others (but this should be easy to correct); or else certain errors on the user's part - not foreseen when the programme was developed - are impossible or too difficult to correct (even if they are minor errors, not disturbing the validity of the results, this could be particularly frustrating for the user).

C. First Approach: Table Manipulation

1. General Conception

The data used to represent a project can be represented in the form of "variables", a variable being a time series of values, for a number of periods corresponding to the duration of the analysis, and associated with a name. The time basis will be annual, and a year zero is included to indicate the *before project* values of the variables. When the without project situation is to be considered constant, the year zero of the project variable is thus enough to represent also the without project situation. When it is intended to specify a phenomenon on a more detailed time basis, by month, for example, the simplest thing is to specify a variable for each sub-period. This organization of data can be visualized as a table where the lines are the variables, the name of the variable being the title of the line, and where the columns correspond to years, hence the name "table manipulation" of this approach.

Each variable represents an element of the costs or benefits of a project, either in physical quantities or in value. A farm model, a project, will be nothing more than the algebraic total of a certain number of these variables; calculation of the models will consist basically of conducting operations on certain variables (input data) and thus creating new variables (results).

It should be possible to make the calculations and use the other available functions by two different methods: the direct method, in which a function is selected, the necessary data provided, and the function performed on the spot; and the "programmed" method, in which a series of operations can be defined - with all the data necessary for each operation - and afterwards carried out, like a programme. The programmed method makes it possible to define a model based on the elementary functions available and to recalculate the model as often as desired, that is to say, whenever changes are introduced in the original data.

A simplified example will illustrate this general concept: a project to build an irrigation system for rice cultivation may be summed up in the following variables: investments (construction of system, perhaps access road, administrative buildings, etc.), area under rice (year by year), yield, expenditure (extension, for example). Production will be a calculated variable (area multiplied by yield), as will value of production. The price will, depending on circumstances, be a constant in the definition of a calculating operation, or else another variable (if it is not constant). It only remains to substract the different cost lines from the value of the production, to obtain the net value of the project.

The power of a programme will be directly related to the nature of the calculation functions available. An essential aspect of the development of the programme will therefore be the identification of the necessary calculation mechanisms, which are described in the following section.

2. Necessary Calculation Functions

The functions described below must be included in a programme of the table manipulation type in order to perform the necessary calculations for the great majority of models which the author has worked on. Exceptions are specific cases such as the herd models (see IV.F. below) or special models (forestry project described in III.A.5). Some of these functions have been developed to meet specific demands from programme users. It is the existence of such functions that makes the difference between the programmes presented in the following chapter, and the commercial programmes that have been developed for business management application (of which the best-known example is the VISICALC programme).

The functions can be put into four categories. A fifth category would cover the calculation functions which can also be envisaged for generating values at the time of data input.

a. Algebraic and Elementary Handling Functions

If X and Y are variables, and a, b and c constants, the following functions will be indispensable or at least useful:

i) Algebraic calculations: aX + b, aXY + b, aX/Y + b, aX + bY + c.

ii) Incremental values compared with year zero.

iii) Cumulative values from one year to another.

iv) Advancing or delaying a series of values for a number of years (for sensitivity analysis of the time spent on achieving certain project objectives, for example); it can be decided that the value stored in year zero or in the last year - depending on the direction of displacement - copies itself.

v) Conditional algebraic calculation: functions of the type calculating a X + b Y + c if X is positive, or if X is negative, or if X and Y are positive or nil, etc., mean that values can be calculated according to the sign of a variable. For example, if X is a variable representing the balance between labour requirements and availability of family labour for a farm, a function such as calculating a X if X < 0 (otherwise the result of the calculation is zero) will make it possible to calculate the cost of hired labour (the constant a being the price).

vi) Transfer to the following period: calculating a X + b Y + c; if the result is negative, tranferring it to the following period. Such a function would, for example, mean that, if X is the consumption and Y the production of a product which can be stored on a farm - in case of surplus - the quantities to be purchased could be calculated:

Var.		Year			
		0	1	2	3
1	Consumption	10	16	20	20
2	Production	12	12	12	12
3	Consumption/production	-2	4	8	8
4	Purchase	0	2	8	8

In the above table, variable 3 is the difference between variables 1 and 2, whereas variable 4 is calculated with the function of transfer (X = 1, Y = 2, a = 1, b = -1, c = 0, result in 4).

b. Phasing Calcultation

This type of calculation (already described in Chapter III, Section A.2) is fundamental in the study of agricultural projects, where the increase in production develops over several years.

The function is used to aggregate data corresponding to a unit of an activity by the number of units entering the project each year. A "unit variable" will correspond, for example, to the yield of a perennial crop each year from the date of plantation, or else the expansion each year of the production of a farm entering a project in year 1, in the light of the duration of extension activity or of gradual realization of investments. The "phasing variable" gives the number of hectares planted each year, or the number of farms gradually entering the project. The data series of the unit variable is multiplied by the number of units in year 1, then shifted by one year towards the right, then multiplied by the number of units of year 2, and so on.

If the without project situation is considered constant, and is represented by the value of the unit variable for year zero, this value will be copied in years 1 and 2, for example, to represent the situation of units entering the project in year 3. The calculation could in fact be made by specifying a sufficient number of multiplications, additions and shifting, with the elementary functions of the above section.

But the calculation thus made is no longer correct if the without project situation is not projected as constant. In this case, two unit variables are necessary, one for the without project situation and one for the with project situation. The example below explains how the calculations should be made. The method of calculation is different depending on whether one wants results in total value or in incremental value (compared with the without project situation).

	Year							
	0	1	2	3	4	5	6	
Unit variable								
Yield with project	1	1,5	2	2,5	3	---->	year 20	
Yield without project	1	1,1	1,2	1,2	1,2	---->	year 20	
Phasing variable								
Number of ha (supplementary)	-	10	20	20	-			
Results - total values								
ha of year 1	10	15	20	25	30	30	30	
ha of year 2	20	22	30	40	50	60	60	
ha of year 3	20	22	24	30	40	50	60	
Total	50	59	74	95	120	140	150	
Results - Incremental Values								
ha of year 1	-	4	8	13	18	18	18	
ha of year 2	-	0	6	16	26	36	36	
ha of year 3	-	0	0	6	16	26	36	
Total	-	4	14	35	60	80	90	

For example, the incremental value of year 3 for the 20 ha entering the project in year 2 is equal to the with project yield in year (3 - 2 + 1), or 2, less the without project yield in year 3, or 1.2, multiplied by 20, or a value of 16.

The phasing function will therefore present different options:

- Constant without project situation, represented by year zero of the unit variable, or else non-constant without project situation, represented by a specific unit variable.

- Number of units in the phasing variable in total values or in incremental values (compared with year zero).

- Results in total values or in incremental values.

Finally, it would be useful to be able to specify the calculation, not on one variable at a time but on a block of variables, since it will frenquently happen, for example, that several parameters of the same crop (production, inputs) have to be aggregated by the same phasing variable (area).

c. Functions of Calculation of Rate of Return and Sensitivity Analysis

The programme can be provided with elementary functions, based on one variable alone, to calculate the present value and the rate of return (ROR). In the latter case, the calculation will be made automatically on the incremental values compared with year zero.

A single function will usefully combine these calculations with a sensitivity analysis: the "balance and switching values" function will make the following calculations:

- The algebraic sum of a number of variables, storing the results in a variable "balance". A weight coefficient should be specified for each component variable, for example, -1 to subtract. This would also lead to a first form of sensitivity analysis (specifying a coefficient equal to 1.2 means increasing the weight of the variable by 20% in the balance). But these coefficients should be used with caution if the values of year zero are not nil: The coefficient is applicable to the entire time series; to increase a whole variable by 20%, including year zero, could have some sense or not, depending on the circumstances; it is up to the analyst to understand what he is doing.

- The calculation of rate of return can be optional (it may be desired to use the function exclusively for adding a number of variables). It consists in calculating the ROR of the balance, and the present values and the switching values of each component variable, as set out in Chapter III (Section A.2). The present values correspond, like the switching values, to total values of variables (that is to say, values stored in the years from 1 to the end of the period), and a new variable, known as "without project balance", will correspond to the projection of the balance of year zero values over the period (sign changed: a benefit in without project situation is deducted from the with project value). It is important to understand that switching values will have a different meaning depending upon whether the stored values in the variables are total or incremental. Let us assume that the value of production of a cereal should be 100 without project and 150 with project. If the calculation is made with a variable containing the total values (100 in year zero, 150 in following years), a switching value of 20% will correspond to 20% of 150, therefore to a variation of 30. If a "value of incremental production" variable has been used (0 in year zero, 50 in following years), a rate of 20% will correspond to a variation of 10, or conversely a variation of 30 will correspond to a rate of 60%. In other words, interpretation of the switching value is not the same whether the variable represents total values or incremental ones.

d. Financial Functions

(i) Calculation of Debt Service

It is very useful to be able to specify the characteristics of a loan in a model, and to have available functions calculating the debt service. In particular, this will make it easy to decide, on the basis of a few trials, the conditions of credit which would suit any given investment programme, for example, by changing the interest rate, or the grace period, and by recalculating the model to see how the cash-flow balance reacts.

In practice, one function on an annual basis will certainly be enough. The necessary input parameters will be the interest rate, the grace period, and the number of years for repayment. A number of options will be useful:

- Interest paid, or not, during the grace period.

- Repayment in the form of constant total annual instalments or repayment of principal in constant annual instalments.

- Possible calculation, apart from debt service, of the interest component and the principal component in annual repayments, and of outstanding debt at year end.

The calculated loan values will be stored in a variable. It will be useful to specify whether the values should be considered as separate loans (with the same characteristics), or as a single loan. The first case will, for example, amount to a credit which is renewed each year; or will allow in the same variable the specification of a loan for an investment, and then its renewal after several years. The second case corresponds to a single loan, of which disbursements extend over several years, a frequent situation which arises in the financing of investments made over a long period. It is important in this case that the programme should not allow for the specification of a grace period which would imply a first repayment before the end of the disbursement.

(ii) Calculation of Depreciation

If it is intended to use the programme to calculate financial models of an accounting type, a function of depreciation calculation will be useful, able to calculate the amounts of depreciation in one variable, and the residual accounting value of the assets in another. Several methods of calculation can be envisaged (degressive depreciation), or the function can be limited to the calculation of a linear depreciation.

(iii) Interest on Overdraft

This function was conceived to make it possible to prepare financial analysis models (projections of cash-flow and balance) which balance automatically, assuming that cash-flow deficits are automatically financed by an overdraft (short-term credit) of which the agios are imputed to the following exercise. Assuming that the "cash-flow balance before losses and profits" and the "losses and profits before agios on overdraft" are stored in two variables, the function will calculate four new variables:

- Interest paid on overdraft.
- Balance of Profit and Loss Account (including interest on overdraft).
- Cash-flow balance.
- Cumulated cash-flow balance.

e. Functions to "create" Variables

When the initial variables of a set of data are defined, the values should be entered on the keyboard. The burden of work can be lightened by introducing a number of functions which will allow the specification of only a few values, the programme calculating the others:

(i) Automatic Repetition

Quite frequently, the values of a variable evolve over several years, after which the variable remains constant until the end of the period of analysis. It is important not to have to repeat this value. A convention should allow the programme to generate the end of the series (this could be the explicit specification of the number of years during which the data vary, or the repetition by default of the last value specified, or the use of a code such as the value PI in an interpreted system disposing of the function PI on the keyboard, etc.).

(ii) Cyclic Repetition

In the case of a forestry project being analysed over a long period (50 years, for example), and in which there are several fellings, it will be very useful to have a function allowing for automatic repetition of the series of 10 to 15 values corresponding to a production cycle.

(iii) Interpolation

Often an expert will be in a position to say, for example: the yield of this particular crop will rise from its present value to such and such a value in so many years. The intermediate values are interpolated, either in a linear way, or by an S curve (a sigmoid curve, also called logistic curve, corresponding to the observed rate of development of many phenomena linked to a process of training, for example, agricultural yields in the context of an extension programme).

(iv) Use of Percentages

If fertilizer consumption for one crop, for example, is specified at full development for year 6, it will be useful to be able to specify previous values in the form of percentages of final value (50% year 2, etc.).

(v) Exponential Compound Growth Function

One variable may have an initial value of 25, and increase by 5% per annum from year 2 until year 10, then remain constant.

(vi) Definition by Reference to Another Variable

The values of a variable X are defined by the value for a given year and a form of variation corresponding to another variable Y previously defined.

Example X = 10 in final value

	Year					
	0	1	2	3	4	20
Y	5	10	23	37	50	
X	1	2	4,6	7,4	10	

Thus, for example, one could extrapolate the consumption of an input X in accordance with the estimate of the yield in Y.

3. Tables

A project, with its different components and several farm models, will represent a considerable number of variables. Out of this whole, which can be visualized as a big table with several hundreds of lines, and as many columns as there are years (0 to 20 classically), it should be possible to easily extract lines and columns in order to specify different levels of table detail depending on the different elements of the analysis.

It is very important that the tables should be prepared in such a way that they can be directly photocopied into reports; the typing of figures in tables is in fact a long and unpleasant job, with a high risk of error, thus it would be ridiculous to have to retype tables simply because the programme prints figures and data in an incomprehensible way, or in a way which the user does not like.

D. Second Approach: Detailed Structuring of Data

1. General Conception

The general idea consists of organizing the data (the "variables" of the preceding approach) into specific data types which imply directly to the programme a number of standard calculations and output tables. Thus all the work of specification of calculations and tables is eliminated for the user.

The project is supposed to have a hierarchical structure. At the lowest level are the crops, defined by their yield and the needs in factors of production; the crops are combined, with investments if necessary, into farm models. The farm models are added at the level of sub-areas or components of the project, and/or the whole of the project area. At each level, it is possible to add other elements of costs or benefits (for example, infrastructure, extension service, etc.).

Such a programme will have the advantage of requesting the user to specify the data according to categories corresponding to a concrete reality. The types of data and the predefined tables will be a powerful element for standardization of the analysis methodology and the presentation of reports.

The degree of flexibility of the programme will depend directly on the variety of data envisaged at the conception of the programme, and the categories of calculations and tables that will have been predefined.

The two following sections present the list of types of data and results desirable to include in such a programme.

2. Types of Data

At each hierarchical level of the model, it is possible to pre-define all the possible categories of data, but in all cases an "open" category must be envisaged, that is to say, the possibility of specifying a line of values with a name (a "variable" of the type of the manipulation table in the first approach), or else two categories (costs and benefits) in order to avoid having to give the data an algebraical sign.

The principal categories of data could be the following:

- Prices: at this level the list of productions and inputs of the project could be defined, with their prices. A predefined category may be envisaged for "labour force", in which could be defined several types (family or wage-earning, skilled or unskilled, etc.). The prices could be economic or financial.

- Crops: the following categories are necessary if it is intended to cover all the cases that could arise: main production and by-products (yield and losses), investment costs, inputs, labour force, other production costs.

- Farm model: as well as the area occupied by each crop, it should be possible to specify at this level a number of costs and benefits that could be the subject of predefined categories: availability of labour (by level of skills), household consumption, work outside the farm, investments, operational costs (other than those directly attributable to the crops) and various other costs, taxes and charges, sources of finance (personel contribution, loans).

- Component of the project and/or total project: at this level, the distribution of holdings according to the farm models could be specified, and the rate of entry of these holdings into the project, as well as a number of cost or benefit variables applicable at this level (example: access roads, extension services, etc.).

The programme should make it possible to specify either one set of data, including year zero, corresponding to a without project situation projected as constant, or two sets of data, with and without project. It should also be possible to detail within the year consumption of factors of production, or some productions. A choice could be made between the possibility of defining sub-periods in the year and that of defining an input, for example, several times. That is to say, in a model on an exclusively annual basis it is always possible to define twelve categories (one per month) instead of one for work or irrigation water, for example.

3. Calculations Performed

Organization of the data in predefined specific categories means that the programme can automatically perform all the necessary standard calculations, and produce a set of tables sufficient to meet all needs.

The necessary calculations are defined by the nature of the output tables required. The following will be needed:

- Prices.

- Crop models:

    - production and consumption of inputs in quantity;

    - labour requirements;

    - crop budgets, with gross margin, and margin after remuneration of the family labour force (discounted against the cost of the wage-earning labour force).

- Farm models:

    - production and consumption of inputs in quantity;

    - labour requirements;

    - farm budget: this table will present the sum of costs and receipts, the benefit before financing, the sources of finance and the debt service, the benefit after financing. At this level the present value of the benefit can be calculated, and its rate of return (either before or after financing, as required).

- Sub-areas or components of the project, and the whole project:

    - basically the same tables will be found at this level. The data for production, consumption of inputs and labour, in quantity or in value, will be broken down either by crop (as if the area were one large farm) or by farm type.

The calculations - and the corresponding tables - will have to be done either in economic prices or in financial prices, and either for the with project situation, the without project situation, or the difference between the two.

E. Third Approach: General Structuring of Data

1. Fundamental Concepts: Commodity, Activity, Plan

The idea is to seek a middle way between the two approaches presented above, that is to say, a structuring of data which allows for maximum predefinition of calculations and output tables, but which limits the types of data, taking as a criterion of organization not the meaning of these data (example: work), but the type of calculation to be made: all information subject to the same type of calculation could be grouped under the same type.

This approach was initially discovered by the author in the conception of a programme called MADS (Multipurpose Agricultural Data System), a programme developed in the context of joint activities conducted by several countries on the use of computers in agriculture, within OECD (ref. 42). The MADS programme is described in Chapter V. Some of the MADS concepts and choices were abandoned, and other types of data introduced, when several programmes derived from MADS, also described in Chapter V, were being developed in FAO, Rome. The results of this work are described below.

It may be said that the calculations necessary to appraise an agricultural development project usually come down to the following problem: the calculation, at different future dates, of a number of costs and benefits. These costs and benefits often correspond to data in physical quantities multiplied by prices; it may be said in this case that they correspond either to consumption or to the production of certain commodities (such as, for example, production of grain, consumption of irrigation water or fertilizers, etc.). It will be noted that there is no fundamental difference between consumption and production, only a difference of algebraic sign. Also, even when costs and benefits correspond to data introduced directly in value into the model, it is always possible to represent them as corresponding to a certain quantity of an imaginary commodity with a price equal to the unit. It is therefore possible to represent all the costs and benefits of a project as the consumption or production of a certain commodity. Thus anything which has a price and is either consumed or produced by the project is called a "commodity": the "commodity" is defined by a name, a unit, and a price (or, more exactly, a time series of prices; one can also conceive of linking several sets of prices to the commodities: financial price, economic price, foreign exchange component etc.).

The project calculations amount to working out (for a crop, a farm model, a project component or the whole project) the consumption and production of commodities, and the consumption/production balance for commodities which are both consumed and produced. It is therefore necessary to see how these consumptions and productions will be specified in the models.

Anything in a project which consumes and/or produces commodities may be described as an "activity". The typical activity is the crop: 1 ha of sorghum, for example, which produces grain and straw and consumes labour, seed, etc. But the concept is far more general: dairy cow activity consumes feed and labour and produces milk (and meat when the animal is culled); the activity of building dams consumes labour and cement during construction, then produces electricity and irrigation water, etc.

An activity will be defined by a name, a unit, a list of commodities, with an indication for each commodity as to whether it is consumed or produced and also the corresponding level (in the form of a time series of values).

The level of production thus corresponds to the yield of a crop. The difference between consumption and production is exclusively the algebraic sign of the level. There is nothing to stop the same commodity from appearing more than once in the definition of an activity, whether in production or in consumption, with different levels corresponding, for example, one to production (yield) and the other to consumption (losses). Very often, there will be a labour commodity which is produced by a family activity - which also consumes part of the productions of the farm - and consumed by crop activities; or else a grass commodity produced by a pasture activity and consumed by a cow activity.

The different activities are combined linearly in a new structure of data called a plan. A plan will be, for example, a combination of crops and other activities in a farm model, or else the aggregation of farm models at project level. A plan is therefore a linear combination of activities or plans. It is defined by a name, a unit, a list of components (activities, plans, or other types of data defined below), and for each component the corresponding level (in the form of a time series of values), and also an indicator stating the calculation method to be adopted: annual or phased.

This last idea can best be explained through an example: that is, a rice commodity, a rice activity that produces rice, and an "irrigated area" plan which specifies the number of hectares of rice irrigated during the project.

The data would therefore be presented as follows:

	Years						
	0	1	2	3	4	5	6 ---> 20
Commodity							
Rice (tons)	150	150	150	150	150	150	150
Activity							
Rice (1 ha)							
Rice commodity	1,2	2,0	2,5	3,0	3,5	3,5	3,5
Plan							
Area							
Rice activity	-	100	200	300	-	-	-
(additional ha - phasing)							

This data means that the "area" plan includes the rice activity, at the level of 100 ha in year 1, 200 ha more in year 2, 200 ha more in year 3. The "rice activity" produces 1.2 t/ha before the project, and the yield as a result of the project rises from 2 to 3.5 t/ha in 4 years. The price of the rice is 150/t. There are two possible ways of calculating the production of the area:

i) If the yield depends on the year of the project, all that needs to be done is to multiply the value of the yield for a given year by the total level of activity of the same year (that is to say, the total area under the crop). In the above example, the production in year 3 would be 600 ha x 3 t/ha = 1 800 t. This method of calculation is called annual.

ii) If the yield depends on the age of the activity, that is to say, on the number of years which have passed since the entry of the activity into the project, the calculation must be made according to the phasing method described in Section C.2.2 of this chapter. In the above example, production in year 3 would be:

(300 ha x 2 t/ha) + (200 ha x 2.5 t/ha)
+ (100 ha x 3 t/ha) = 1.400 t.

This example illustrates several points:

- In the definition of a plan, it is necessary to specify the type of calculation needed, annual method or phasing method; the calculation will always be made in total values; in the case of a non-constant without project situation, it will be defined by another activity (rice without project) and a corresponding plan; the additional values will be calculated by the difference between the two plans (that is to say, a plan composed of the "project" plan at level 1 and the "without project" plan at level -1, with calculation by annual method).

- If the calculation is by the phasing method, it is necessary to state whether the level of components is expressed in a total number of units, or in an additional number of units compared with the previous period. The level will always be a total number of units in the case of calculation by the annual method.

- There is no difference between an annual crop and a perennial crop at the level of definition of an activity; the possible difference will be at the level of the method of calculation of a plan. The above example shows that the phasing method covers far more cases than those of perennial crops and other lasting activities.

- When the calculation of aggregation by phasing is done at several levels and the without project situation is not constant, it is important to have a clear idea of the mechanisms of the calculations in order to avoid the risk of error in the results.

The three concepts of commodity, activity and plan are extremely general and a wide variety of models can be expressed using them alone; however, some operations are difficult or even impossible to specify. It has therefore been necessary to allow for other types of data organization as well, which are described in the following section.

2. Refinements: Complementary Types of Data

This section describes three complementary types of data which have been introduced to meet specific operations: these are investments, credit operations, and transfer activities. Another method of organizing data - known as "aggregate" - will then be described, and the section will end with a discussion of the way in which details may be specified by sub-period in the year.

a. Investments

It is a current practice - often linked with financing mechanisms - to distinguish in project costs what is investment and what is operating cost, even if the distinction sometimes seems arbitrary. For example, operating costs in the initial period of a project could be considered as investments, as, for example, the deferred costs within the accounting plan for an enterprise. In the present context, the type "investment" has been conceived not only to make it easier to distinguish this category of costs in the tables of data and results, but also to provide a number of specific functions of calculation, corresponding to the usual way of treating investments. The type "investment" may be seen as being, in the hierarchical structure of the project, intermediate between the commodity and the activity; it will be specified as a component of a plan, just as an activity or another plan.

An investment will be defined by a name, a unit and a unit price - like a commodity. The number of units for each period (for example, the number of vehicles purchased each year, or the number of hectares developed), comparable to the level of a commodity in an activity, will be defined at the level of inclusion in a plan. If the investment is directly specified in value, there will be specified either a price equal to the unit, the values being introduced at the level of the inclusion in a plan, or a price corresponding to the values, the level of the investment in a plan being then 1. A number of other parameters will also be specified in the definition of the investment type, corresponding to the particular functions available, which are:

- Duration of life of the investment, allowing for the automatic calculation of its replacement, and residual value at end of life, expressed in percentage of the initial cost.

- Percentage of cost to be added to the total for physical contingencies (the normal technique to estimate the amount for contingencies to be added to an estimate).

- Percentage of the cost used to calculate the costs of maintenance of the investment, according to a method currently employed for certain types of investments (buildings, irrigation systems, roads, for example), and number of initial years after the realization of the investment during which this cost of maintenance is not applied.

b. Credit Operations

We find here again the need to dispose of a function for calculating debt service, as already detailed with regard to the "handling table" type of model (Section C.2.4.1).

A credit operation will be defined by a name, a unit, a series of values representing the loan amounts each year, and by the parameters necessary to define the operation: interest rate, grace period, number of years for repayment, type of repayment (constant or capital constant), interest paid or not during the grace period, series of values of the loan to be treated as a single loan or as so many separate loans. The series of values could be the result of calculations of a plan, that is to say an investment, or costs of production. The level of the credit operation in a plan corresponds to the part of these costs to be financed by a loan. For example, a credit operation specified on the basis of the category of "hydro-agricultural management" investment, and introduced in a plan at level 0.7, means that 70% of this investment will be financed by a loan.

c. Transfer Activities

Elements of cost or benefit of a project, which depend on the results of the calculation of a plan, cannot be defined directly with the concepts of activity and plan. The most obvious example is that of farm work; the quantity of wage-earning labour that will have to be used depends on the balance between the availability of a family labour force and the labour requirements of each activity (crop) of the model. The consumption of the commodity "wage-earning labour force" will depend on the production/consumption balance of the commodity "labour" in the plan "farm". This mechanism of calculation is applied in many cases: calculation of a subsidy based on fertilizer consumption, calculation of charges in proportion to the value of production, calculation of the "à la part" remuneration of the crew of a fishing vessel, purchase of feed concentrates depending on a fodder balance, etc.

The concept of transfer activity was developed by the author to deal with this type of situation. Say there are two commodities, one called "source commodity" (SC), and the other "transferred commodity" (TC). A transfer activity is an operation which, when it is introduced as component of a plan, is reflected in the algebraic addition to the consumption/production balance of the commodities in a plan as a "transferred commodity", at a level determined by the balance of another commodity (deficit or surplus, for the year under consideration, of the "source commodity") multiplied by a "correspondence coefficient" (CC).

When the TC represents, for example, compensation for an SC deficit, this SC deficit should be cancelled by the transfer activity; this mechanism is called "compensation transfer", or "transfer back". When the TC is added to the balance of the plan without influencing the SC values (calculation of a subsidy or a charge, for example), there is no transfer back.

The concept may be illustrated by two simple examples:

For a certain year of a certain plan, we have:

Labour consumed (by crops):	-150
Labour produced (by the family):	130
Balance	-20

If one introduces into the plan a transfer activity covering the TC "hired labour", based on the SC "labour", with a coefficient CC = -1 (that is to say, in a ratio 1 : 1, and shown as a cost) and with transfer back, the result will be:

	Labour consumed	-150
	Labour produced	130
Transfer	Hired labour	-20
	Labour (compensation)	20
Balance	Labour	0
	Hired labour	-20

Example without transfer back:

A plan produces grain. A tax equal, for example, to 20% of the value of the grain must be paid. A "charges" commodity will be defined, whose price is equal to that of the grain, and a transfer activity with grain as source and charges as TC, with a CC equal to -0.2:

	Grain produced	120
Transfer	Charges	-24
Balance	Grain	120
	Charges	-24
	Balance	96

In summary, a transfer activity is defined by:

- A source commodity, SC. The definition could be generalized by using as source, possibly, an investment or an aggregate (see below).

- A transferred commodity, TC.

- An indicator saying whether the transfer is based on an SC surplus or deficit, and if this should be considered in quantity or in value (this distinction is necessary to decide the meaning of the CC).

- An indicator saying whether or not there is to be a compensation operation (i.e. transfer back).

The TC level is equal to the absolute value of the deficit or the surplus, as the case may be, multiplied by the CC.

The correspondence coefficient can be specified in the form of the level of the transfer activity in a plan. This will therefore be a time series and not a single value (so that it could deal with the case, for example, of a subsidy whose rate would be degressive).

The transfer back of SC in the case of compensation is equal to the absolute value of the deficit if the operation is based on a deficit, or to the absolute value of the surplus multiplied by -1 if the operation is based on a surplus.

d. Aggregates

This is the name given to the specification of sub-totals, which will be calculated automatically at the time of plan processing if it has been defined within a set of data. Let us suppose, for example, that a model includes the commodities millet seed, sorghum seed and fertilizer; an aggregate "inputs" may be defined as equal to the sum of these three commodities; at the time of calculating a plan, those commodities present in the plan will be added together to calculate the aggregate. The components of an aggregate can be commodities, investments, credit operations, or other aggregates, and each component is attributed with a weighting coefficient (which gives the aggregate a more general character than that of a simple sub-total). The aggregate can be calculated at choice, either exclusively on the values, or on both the quantities and the values. A particularly important application of aggregates concerns profitability analysis, described later (Section 3).

e. Sub-periods in the Year

As set out in Section D in the case of detailed structuring of data, detailed analysis of time periods within the year can be treated in two ways:

i) In the interest of simplicity, the programme can be developed without any specific provision. As many commodities can be defined as periods are desired (for example, labour-January, labour-February, etc., for a monthly analysis), and the annual totals can be calculated using aggregates.

ii) It may be decided to "earmark" certain commodities as having to be defined by sub-periods. In this case, attempts can be made to simplify data input, for example, by specifying for one commodity the annual total, and a distribution in percentage between the different sub-periods of the year. The results of the calculation of a plan will include, as well as the values for each sub-period, the annual values for the sum of the deficits and for the sum of the surpluses (there is no compensation from one sub-period to the next). In the case of a transfer activity, these lines will serve as a basis for calculation. Example: for a given year:

	Labour consumed	-150
	Labour produced	130
	Balance (without transfer)	-20
By sub-periods:	Sum of deficits	30
	Sum of surpluses	10
Transfers:	Hired labour	-30
	Labour (compensation)	30
Balance:	Labour	10 (surplus)
	Hired labour	-30

3. Results

a. Standard Calculations

The calculations of a model are implied by the definition of a plan. A plan can be composed of commodities, activities, investments, credit operations, transfer activities, and other plans. The aggregates defined at the level of a data set will automatically be calculated for any plan of the model. According to the composition of the plan, the results, in the form of tables of time series of values (year by year), will include all or some of the following elements, either in quantity or in value:

Commodities:	- commodities consumed - commodities produced - commodities transferred - balances of commodities
Investments:	- investment and replacement - physical contingencies - maintenance of investments - residual value
Credit operations:	- loans - total debt service (for each loan) - interest component of debt service - principal component of debt service - outstanding loan at year end
Net value of plan	
Aggregates.	

The net value of the plan enables the present value and rate of return to be calculated.

All the above calculations can be done automatically when a plan is specified. The user will have to specify the results he wants printed: in quantity or in value, and, if year zero is used to represent the without project situation, total or incremental results compared with year zero.

b. Calculation of Rate of Return and Sensitivity Analysis

As with any budgetary simulation model, a sensitivity analysis can be made by changing the value of certain parameters and recalculating the whole model to see the impact on the results.

Another method is the calculation of the switching values of each of the variables composing a plan. At this level, it would be desirable to introduce two options:

- Detailed calculation based on the whole plan, that is to say, on each component (commodities, investments, loans) of switching values.

- Calculation on an aggregate: the rate of return and the present value will be calculated based on the value of the specified aggregate, and the switching values on each component of the aggregate (of which many will be other aggregates). The variables of a model may thus be regrouped at the level of detail required for the sensitivity analysis.

c. Importance of Data Organization to the Results

The flexibility of such a model is due to the fact that the nature of the plan calculation results is directly specified through the definition of commodities and other types of data. Depending on whether one or more commodities (possibly all at the same price) are used as different components of a plan, one or more lines of results will be obtained when the plan is calculated. Similarly, as mentioned above, the level of detail of the calculation of the switching values of the variables is fixed through definition of aggregates.

The meaning of a result, or the possibility of calculating such and such an element of a project, is directly linked to the way in which activities and plans are defined.

Experience shows that the commodities-activities-plans system enables a very great variety of situations to be described, sometimes at the price of a little imagination in the specification of data.

F. A Special Case: Livestock Projects

1. The Problem

When a ranch project is studied in Latin America, or a project for improvement of traditional animal husbandry in West Africa, it is difficult to treat the herd in the static way that one treats "activities" or "variables" in the models described above. The structure of the herd does indeed change according to precise rules, which correspond to breeding performance, to the rate of mortality, and to sales policy. The herd must be studied as one would study a human population, that is to say, on a demographic type model. It is only in simple cases, where the herd is small, the demographic performance stable, and the possible mortality compensated by purchase of new animals - in the cases, in other words, where the breeding aspect of the herd is secondary - that one can do without such a demographic model (in the case, for example, of European-style dairy animal raising). In developing countries the breeding of young animals is often of primary importance, and one of the major effects of any project will be an improvement in the performance of such a breeding herd. The normal method of evaluation of the effects of such a project is the use of a demographic model of the herd: the expected effects of project activities are "translated" through a technical judgement based, for example, on similar projects or on pilot programmes, into changes of the technical parameters that define the animal population concerned. A purely mechanical calculation will then translate these parameters into results: composition of the herd, mortality, production.

The following section describes the principles governing the design of this kind of model, and Section 3 sets out what are the other data that must be taken into account in a study of this type, such as the different productions (milk, manure, work, etc.), and particularly the overall size of the herd, in the light of feed supplies available or any other criteria.

2. The Dynamic Model of the Herd

The herd will be divided into age classes by sex. The structure of the herd will be completely defined by:

- Its initial composition.

- The demographic parameters: fertility rate, age on first calving, mortality by class of age.

- The parameters of utilization: purchases and sales of animals.

a. Structuring in Age Classes

Here again the base time will be the year, but some questions arise, linked in particular to the fact that the multiplication of age classes involves the need to specify a greater amount of data, which is not always available.

The level of detail will depend on the method of utilizing the herd, its level of performance, and the precision of available data. Ideally, a programme should allow the user to define age classes according to needs, either by aggregating a number of annual classes in one alone (for example: females over three years, males of 3 to 5 years), or by subdividing one class into two semesters, or even shorter periods (to be able to specify, for example, that heifers move into the breeding cows class at 2 1/2 years). The programme described in Chapter V was written for cattle herds, and the definition of age classes is fixed, which in practice has sometimes turned out to be a constraint. On the other hand, sufficient flexibility of definition of age classes means that one can imagine that the same programme could serve for different species (small ruminants, even pigs).

Another problem is knowing at what stage it is desirable to separate from the rest of the males those animals intended for breeding. One simple, but not always entirely satisfactory method of dealing with this problem consists in one of the following two options: (i) treat all the males, including those used for breeding, within the same categories, and (ii) create a "breeding males" category outside the herd, for which sales will be determined by a culling rate, and pruchases by the need to maintain a certain ratio between male and female breeders. One possibility - for the frequent cases where breeders are born in the herd, but raised by different methods as from a certain age - would be to specify, as from a certain age, two classes of males. This would also require the definition of a system of calculating the transfer of the animals between the two categories (depending on the need for adult breeding males, for the herd itself and possibly for sale).

b. Demographic Parameters

The parameters determining the evolution of the herd are rates of fertility and of mortality.

In practice it will be enough to consider one rate of fertility applying to all females of breeding age. The number of these females will depend on the age of first calving, that is to say, the age at which the females are transferred to the breeding group.

Mortality varies considerably according to age. It will therefore be necessary to specify several rates. Without going as far as a rate for each category of animals, four rates would be enough in the case of cattle: male calves, female calves (in many regions, the care given to calves varies according to the sex, and the rate of mortality could be very different), animals from 1 to 2 years and animals over two years. Here again, a general purpose programme could define on request the classes to which mortality rates will apply.

c. Purchase and Sales Parameters

The last piece of information needed for complete determination of the structure of the herd concerns purchases and sales of animals.

Purchases are usually expressed in numbers of animals, since they represent very few operations decided a priori (one exception - already pointed out - is that of male breeders), whereas sales are specified by annual percentages of numbers of each age class: the culling/sales rate. It will be necessary to specify a culling/sales rate year by year for each of the age (and sex) classes of the herd. These rates will reflect the method of management of the herd, and, to some extent, the marketing policy. The production of a herd will be different - even with equal fertility and mortality - if these rates are changed. In fact, one of the major interests of this type of model is that it can be used to test the effects of a change in method of exploitation of the herd, as shown by the example in Chapter VI (Section a.4).

d. Method of Calculation

When the different detailed parameters above are fixed, the structure of the herd is entirely determined. The data comes either from surveys made in the area, or from information acquired in similar areas, supported by information provided by experts working in the area, or even the livestock-raisers themselves. The composition of the herd could be calculated year by year by equations of the following type, assuming, for example, that the numbers are counted at the end of the year:

> Numbers in class 3-4 years (for example) in year t =
> number in class 2-3 years in year t - 1
> - mortality (that is to say, number X mortality rate)
> - sales (that is to say, number X culling rate)
> + purchases.

The deciding element in the size of the herd will of course be the number of female breeders. It will often happen that the size of the herd must be limited (see following section) and that the application of the parameters provided to define the model is reflected by continued growth of the herd. It will be wise for the programme to be equipped with a calculation mechanism which will automatically adjust the size of the herd by selling - when necessary - more females than the application of culling rates alone implies. The criteria could be a ceiling for the maximum number of breeding females, but less "direct" criteria are sometimes desirable, as discussed in the next section. A useful option would be to allow the user to specify what age class (or classes) the surplus females should be culled from for sale (should calves, heifers or cows old enough for breeding be sold?).

The demographic model can be shown in three tables:

- Projection of the herd structure.

- Mortality (by age class and overall rate).

- Sales (by age class and overall rate).

These results will serve to calculate the benefits linked to utilization of the herd, and also the costs, which are proportionate to the number of animals, as explained in the next section.

3. <u>Other Elements to be taken into Consideration</u>

To translate the results of the demographic model into a cost/benefit analysis, a number of additional data, detailed below, must be specified. Lastly, it would be desirable, in such a model, to be able to link the mechanism of stabilization of the size of the herd - described above - to factors such as, for example, the general availability of pastureland, or even to be able to specify in a more detailed way the feed needs/availabilities balance, which is one of the essential aspects of herd management.

a. <u>Cost/Benefit Data</u>

The results of the herd projection of the demographic model will enable a number of simple calculations (but burdensome to do manually) to be made through the specification of the corresponding data:

i) The specification of weights - every year, to take account of the improved performance of the animals under the effects of a project - and the prices per kilogramme for each class of animal will make it possible to calculate meat production, in quantity and in value (it will be expressed in live weight or in carcass weight according to data chosen).

ii) The model will also make it possible to specify the other productions. Some of these productions can be pre-specified, particularly milk, which will be specified by a yield and a price (always year by year). It will be desirable to specify the yield by lactation and not by year; changes in fertility rates will thus be directly taken into account. In all cases, there will have to be an "other productions" category, which could contain a number of variables which will be defined by:

- A name.

- The specification of the animal class or classes serving as a basis for calculation.

- The annual coefficients to be used (either a series, corresponding to the value per head, or two series, one corresponding to quantities per head and the other to unit prices).

One could thus, for example, specify the production of work by the non-breeding males aged three years or more, and the production of manure by those of the animals which are kept in stalls, etc.

iii) Investments must be specified separately, according to, for example, the modalities defined above in E.2.a, that is to say, specifying for each investment a name, a unit, a series of unit prices and the number of units per period, the duration of life, the percentage of physical contingencies, the percentage of maintenance costs.

iv) The other cost elements are to be divided into two categories: those which are determined in an exogenous way, which may be called fixed costs, and those which depend on the number of animals in one or several categories which may be considered as variable costs. The fixed costs will be determined by a name and a series of values, or possibly a series of prices and a series of quantities. The variable costs will be defined by a series of unit prices and by the indication of the animal class or classes to which these costs apply (in other words, the cost is equal to the unit price multiplied by the number of animals in the category concerned).

v) The above data, as well as the demographic model, will make it possible to compute all costs and benefits linked to the improvement and utilization of a herd, and therefore to calculate the net balance, its present value and rate of return.

b. Stabilization of the Size of the Herd and Feed

It has been shown above (paragraph 2.d) that the size of the herd should be limited by increasing sales of females, if necessary over the values specified by the culling rates. The different ways of limiting the size of the herd could be specified as follows:

i) The simplest is to specify directly the maximum number of breeding females (either in the form of a single value, or better still, in the form of a time series).

ii) One criterion could be the total size of the herd, expressed in animal units, that is to say, giving a specific weight to each animal class, this weight being a time series or a constant. This size will be compared with the specification of maximum possible number of animal units (corresponding, for example, to the maximum carrying capacity of a ranch or a grazing region), and translated by the programme into a need to reduce the number of breeding females through sales.

iii) It could also be useful to include in such a programme the specification of feed needs of the herd, and to use as a factor of stabilization of the herd size a datum (exogenous) of feed availability. The problem here is that feed needs are not expressed simply in a single unit. It is necessary to simplify matters to remain at a level compatible with the precision of the data. The suggestions presented below have not - as far as the author knows - been used in an actual programme of the type of those described in Chapter V (the development of a programme based on such specifications is to take place during the year 1984). It will therefore be necessary to verify through experience if they will meet the realities of demand for this type of analysis. Nutritional needs could perhaps be defined in two or three categories, which could be, according to the user's preference, either categories of concrete feeds (for example, fodder, concentrate), or categories of nutrients (for example, energy, proteins and minerals). One of the categories would be chosen as a factor limiting the size of the herd, and its availability specified in an exogenous way, the model calculating the needs for other categories (corresponding to costs). Depending on the case, a feed-nutrient correspondence would have to be established:

- First case: The animals' needs are expressed (by age class and by year) in fodder and concentrate per head (which is current practice in some countries). The availability of fodder is considered as a limiting factor. The model would stabilize the size of the herd in the light of this availability (exogenous variable), and would add to the costs of the project those of the purchase of concentrate, depending on the quantities calculated.

- Second case: The animals' needs are expressed in energy, proteins and "minerals" (an abstract unit, for the sake of simplicity, representing a typical mixture of the principal minerals necessary in the area). There are then two possibilities to express the availabilities: if they are expressed in energies, proteins and minerals, the result is the same as in the first case. If they are expressed in quantities of feed, the programme should:

  - Specify for each feed the nutritional value in each of the nutrients (energy, proteins and minerals).

  - Specify the quantities of one or more feeds being detemined in an exogenous way, to serve as a constraint on the size of the herd.

  - Calculate the quantities of other feeds which will be charged as costs to the model. The feeds will be taken into account one after the other, in the order specified by the user.

iv) Lastly, an option could usefully be envisaged whereby the calculations of needs and supplies of feed specified above would be possible without this being used as a factor of control of the herd size, but only as a means of semi-detailed calculation of the feed costs, the size of the herd being controlled then by one of the other mechanisms existing in the model. Such a model would make it possible to study by successive approximations the possible balance between the different factors of management of the herd, in a way both simple and closer to the producer's decision making process than, for example, a model (classic for this type of application) of linear programming. Reference will be made to this subject in Section C.2 of chapter VI, which describes programmes of this type intended for use as agricultural extension tools in France.

G. Advantages and Limitations - Towards an Integrated System

1. Respective Advantages and Limitations of the Systems Proposed

Apart from the case of animal husbandry, which can only be treated in a specific way in the context of models including mechanisms for calculating a population, let us see what might be the advantages and limitations of the three approaches proposed above: the table manipulation, the detailed structuring, and the general structuring of data.

The obvious advantage of the first approach, the table manipulation, is its great conceptual simplicity, which enables the user to understand immediately the structure of his data and the nature of the calculations made. This advantage is perfectly illustrated by the astonishing success - in other fields of application - of the VISICALC programme and the many similar programmes which followed it. This type of programme allows for definition from a terminal of lines and columns to be shown in a table, and also the calculations that can be performed with the data from two cells of the table, or between lines, or between columns; the availability of such programmes has greatly increased the use of micro-computers in small enterprises (VISICALC is produced by the American firm VISICORP). The price to be paid is the need to specify in detail all the calculation operations, which requires on the one hand a certain amount of work, and on the other, some organization of the way in which variables are defined. Nevertheless, the real use of programmes of this type has shown that it is quite possible to construct models of several hundred, even several thousand variables, thus considerably reducing the amount of work and improving the quality of analysis in all cases, compared with traditional manual work.

It is also possible to imagine ways of lightening the work load by constituting a library of typical models, which could be copied and adapted to each particular case, as explained in the next chapter (MANIP programme).

Another advantage of this first approach is that, because of its simplicity, it can be implemented by programmes of relatively small size, and therefore can function on small machines (micro-computers). Of course, such an advantage is only relative, first because computers of considerable capacity can be found throughout the world (at least in the capital cities), and secondly, because the capacity of micro- and mini-computers continues to grow at a rapid pace.

A programme based on the second approach - the detailed structuring of data - does not require the user to provide detailed specification of the calculations to be made, ands this is a big advantage. The disadvantage is that the flexibility and general application of such a programme depend directly on the number of predefined data types, calculations and specific tables pre-programmed. This will result in lengthy programmes, therefore requiring a priori big systems, and very "ramified" from the point of view of the user: in the case of a menu system, there will be a large number of menu levels and many options or categories at each level, and in the case of a command language, the user will need to know many keywords and commands. Such a system will be particularly suitable in all cases (and there are many) which can fit into a standard mould of analysis.

The third approach - the generalized structuring of data - corresponds, as mentioned above, to the search for a synthesis between the two previous approaches, associating a sufficiently general structuring of data with predefined typical calculations. Such an approach has proved very powerful, but practical use has shown that the abstract concepts of commodity, activity and plan are confusing to the inexperienced user, who has difficulty in understanding the importance of the link between the way these data structures should be defined and the results he hopes to obtain from the model. On the contrary, after the experience of one or two projects, the model becomes very simple to use because of the very limited number of data types.

2. Functional Specifications of an Integrated System of Programmes

The ideal tool for a user involves in a great variety of projects would consist of the programmes corresponding to the three approaches described above, and a dynamic model of the herd, and ideally, providing the choice between a menu system and a command language. The problem - apart from the implications in terms of programming work and material resources - is to specify the practical modalities of communication among such programmes, that is to say, the exchange of data or results.

Two approaches may be envisaged, which will both be illustrated by the programmes described in the following chapter:

a. The Approach of Local Variables and External Variables

Each module of the system (table manipulation, detailed or general structuring of data, livestock model) is treated as an autonomous model, but with the following possibilities:

- The data of the model can be specified normally - they are then "local" to the model - or specified by reference to an "external" variable. An external variable is a data structure which can be used - through the appropriate references - by any module in the system.

- The results of the calculation of the model can remain internal to the module, that is to say local, or can be transformed into external variables, that is to say, be used to fuel the data base common to all the modules.

The complete system will therefore be composed, on the one hand, of a number of autonomous modules and, on the other hand, of a communal data base with which each of the modules can communicate.

b. The Approach of Hierarchical Organization of Programmes

Each module is an independent programme, with its own files of data and results, and communication among programmes is limited, in the following way:

- Programmes of the data structuring type and livestock model type are considered, because of their relative lack of flexibility, as useful for making only that part of the analysis which could be qualified as standard, composed of predefind data structure and predefined calculations.

- Any analysis outside the above framework will be made by a programme of the "table manipulation" type, which provides maximum flexibility.

- There is no reason, from this point of view, to allow the transfer of information from the table manipulation programme to other programmes. It is enough that programmes in the pre-handling table stages have a function enabling them to transcribe the information (in principle, all or part of their results) in a set of files accessible to the handling table programme.

This second approach, less general than the first, has the advantage of being simpler to use, and also simpler to implement. It is not certain, in the present state of use of existing programmes, that the relative limitations it has are really a constraint in current practice for preparing projects.

The actual programmes corresponding to the different concepts described in this chapter are presented in the next chapter.

V. IMPLEMENTATION

The first part of this chapter describes a number of programmes already in existence or being developed (of which several are being developed by the author), classified according to the approaches presented in the previous chapter, and whose functional specifications are compared with the ideal model.

The bibliography in Annex 7 (Section II.a) gives a list of the works on programming and other handbooks used by the author for his work. The works of Wirth (ref. 26 and 31) and Welsh and McKeag (ref. 30) were particularly useful.

The second part describes a number of problems related to the development of the use of these programmes.

A. State of Development of Programmes

1. First Approach: Table Manipulation

a. CBDISPLAY (World Bank)

This programme was developed in 1979 by the Department of Agriculture and Rural Development in the World Bank (ref. 38), and is written in FORTRAN. It allows for the entry into the system of a number of value series (variables) and the making of calculations of profitability (the name CBDISPLAY means COST-BENEFIT-DISPLAY). It is an interactive programme using a system of command language: when the user is connected - through a terminal - to the system on which the programme is installed, he gives the command, RUN * CBDISPLAY; he receives the prompt: "Put in the number of periods (or run a GET command)". The GET command leads to data retrieval from an existing file. The specification of a number of periods (in principle, years; the maximum number is 50; a special version of the programme accepts up to 200 periods) means that the user can work with a new set of data. The programme, once it has run a command, displays the message: "OK, what's next ?" and waits for a new command. The programme works on data organized in variables (called "streams" in the programme handbook). A variable is composed of a name and a time series of values, for the number of periods chosen.

The name of a variable is composed of a letter prefix, a point, and a suffix composed of 1 to 10 characters (letters or figures). Four prefixes are allowed, corresponding to predefined types of variables:

C : cost variable
B : benefits
N : net benefits
W : weighting coefficients

The programme enables variables to be put into the system, allows for a certain number of variable manipulations to be made, and for rate of return calculations and sensitivity analysis to be performed and produces some graphs on a plotter. The maximum number of variables that can be handled is 1,300. The data are put in by typing a command on the form: <name of variable> = <list of values>.

Examples: C.1 = 25.7, 18.9, 12.7, 8.3, 5, 20 * 0
or C.1 = 25.7, 18.9, 12.7, 8.3, 5, 0

The symbol * is an indicator of repetition. If the list includes less data than there are periods, the last value is repeated automatically. The decimal sign is the full stop, and the comma is used as a separator.

An editor enables a typing error to be corrected before putting in a line, without retyping the whole line, and a variable already defined can be changed by the command:

<Name of the variable> (specification of period) = <list of values>

(The specification of period is a list of periods or intervals).

A variable may be erased (DELETE <name of variable>), the list of variables printed (LIST STREAMS), and all or part of the defined variables printed (the command PRINT allows for specification of the list of variables and the periods to be printed).

A data set can be stored for future use, at any time, with the command SAVE UNDER <name of reference>.

The name of reference may include up to 64 characters, and is used with the command GET to retrieve a data set previously stored.

It is possible, with CBDISPLAY, to manipulate the data to some extent and calculate returns. The series of relevant commands can be stored in the form of procedures, which can later be executed (commands CREATE and PROCESS). Up to 63 procedures can be defined.

Possible manipulations of the variables are as follows:

- A variable can be created as a result of the combination of variables, lagging operations, or arithmetical operations between variables, as illustrated by the examples below:

```
COMPUTE N.TOTAL1 = C.1, C.4, C.8, B.1, B.5, B.7, B.9
C B.2 = B.1 * 1.1, B.3 * 1.2 (multiplication by constants)
C B.2 = B.1 LAGGED 2 YEARS
C B.LATE = B.BASE LAG 4
CALCULATE B.3 = B.1 * W.1 + B.2 (constants not allowed)
```

- There is also a command which creates a group of new variables based on an existing group:

```
                                 UP                 (percent)
GENERATE  <List of variables>    DOWN        k      (percent)
                                 LAGGED             (years)
```

The new variables will automatically be created with the name of the former ones, plus the suffix U, D or L and a number of two digits indicating the change (the variable B.1 reduced by 20 percent thus becomes B.1D20).

- Calculations of present values (PV) and rate of return (RR) are done with the aid of the following commands:

OCC = <discount rate> which fixes the rate (opportunity cost of capital)

PRINT PV <list of variables> <list of rates>

PRINT IRR <list of variables>

The parameters are optional: the last value used in a previous command is utilized by default.

- Sensitivity analysis: three commands are available:

  - The command SWITCHING VALUES (or SWITCH) calculates for a list of variables, and one or more discount rates, the switching values as defined in Chapters III (Section A.2) and IV (Section C.2.c).

  - The command GROUPED SENSITIVITY ANALYSIS tests the sensitivity of a project or a project component, defined as the sum of a number of variables, to changes (in percentage) of components. The test can cover either present value or rate of return. Example:

    GROUPED SENSITIVITY ANALYSIS B.1, B.2, C.1, C.2, C.3, with B.2 DOWN 10 PERCENT, WITH C.1 UP 20 PERCENT C.2 UP 10 PERCENT or possibly:

    GROUPED B.1, B.2, C.1, C.2, C.3 W B.2 - 10, W C.1 + 20 C.2 + 10

    Each WITH corresponds to a sensitivity analysis.

  - The command PAIRED leads to a sensitivity analysis between a pair of variables, one representing costs (incremental), and the other benefits (incremental) of a project, as in the example below:

    PAIRED IRR B.1

    will give the following result on the variables B.1 and C.1 (for any benefit variable there must necessarily be a corresponding cost variable already existing, with the same suffix):

Ok, What's next?
PAIRED SENSITIVITY ANALYSIS B.1,C.1

PRESENT VALUES OF NET STREAMS AT A DISCOUNT RATE OF 12.0%

	B.1	UP 10%	UP 20%	UP 50%	DOWN 10%	DOWN 20%	DOWN 50%	LAG 1 YEAR	LAG 2 YEARS	LAG 3 YEARS
C.1	38612.5	48758.5	58904.4	89342.2	28466.6	18320.7	-12117.1	26631.2	15933.6	6382.1
UP 10%	32327.9	42473.8	52619.7	83057.5	22181.9	12036.0	-18401.8	20346.5	9648.9	97.4
UP 20%	26043.2	36189.1	46335.0	76772.8	15897.3	5751.3	-24686.4	14061.8	3364.2	-6187.2
UP 50%	7189.2	17335.1	27481.0	57918.8	-2956.8	-13102.7	-43540.5	-4792.2	-15489.8	-25041.2
DOWN 10%	44897.2	55043.1	65189.1	95626.8	34751.3	24605.3	-5832.4	32915.9	22218.2	12666.8
DOWN 20%	51181.9	61327.8	71473.7	101911.5	41035.9	30890.0	452.2	39200.5	28502.9	18951.5
DOWN 50%	70035.9	80181.8	90327.7	120765.5	59890.0	49744.0	19306.3	58054.6	47356.9	37805.5
LAG 1 YEAR	-	-	-	-	-	-	-	33473.3	22775.7	13224.2
LAG 2 YEARS	-	-	-	-	-	-	-	-	28884.7	19333.3
LAG 3 YEARS	-	-	-	-	-	-	-	-	-	24787.7

INTERNAL RATES OF RETURN OF NET STREAMS

	B.1	UP 10%	UP 20%	UP 50%	DOWN 10%	DOWN 20%	DOWN 50%	LAG 1 YEAR	LAG 2 YEARS	LAG 3 YEARS
C.1	21.251	23.451	25.590	31.728	18.977	16.610	8.575	17.682	15.154	13.203
UP 10%	19.187	21.251	23.253	28.983	17.049	14.815	7.148	16.063	13.797	12.017
UP 20%	17.411	19.362	21.251	26.639	15.384	13.259	5.886	14.644	12.592	10.957
UP 50%	13.259	14.967	16.610	21.251	11.468	9.573	2.785	11.227	9.638	8.322
DOWN 10%	23.691	26.058	28.364	34.998	21.251	18.719	10.218	19.555	16.704	14.545
DOWN 20%	26.639	29.215	31.728	38.975	23.990	21.251	12.151	21.763	18.506	16.088
DOWN 50%	41.310	44.976	48.565	58.941	37.557	33.701	21.251	32.002	26.519	22.767
LAG 1 YEAR	-	-	-	-	-	-	-	21.175	17.570	15.006
LAG 2 YEARS	-	-	-	-	-	-	-	-	21.081	17.434
LAG 3 YEARS	-	-	-	-	-	-	-	-	-	20.964

Ok, What's next?
STOP

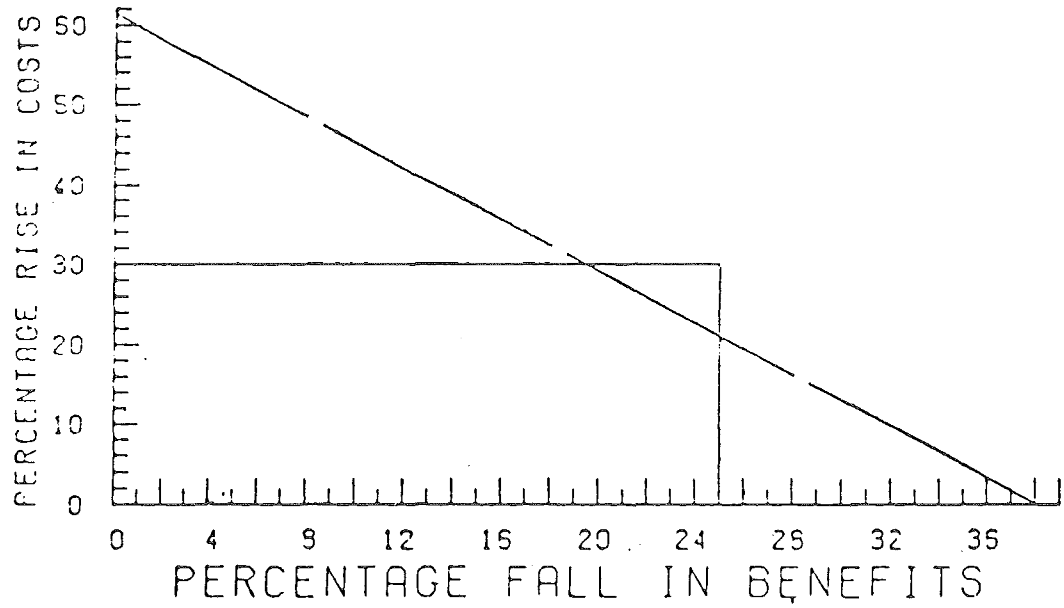
*** GOOD BYE, HAVE A NICE DAY.

- Graphic functions: Two are available. One plots the present value of a list of variables in accordance with the discount rate (command PLOT). The other plots, for a pair of variables representing, one the costs and the other the benefits of a project, the locus of combined changes (in percentage) of each of the two variables which result in a zero value of the present value of the project (for a given discount rate). The graph compares the results with a "window" corresponding to the maximum increase of costs and to the maximum reduction of benefits estimated possible by the analyst. The command LOCUS specifies this function, of which one example is provided below. Several additional commands lead to specification of the graph scale, title, etc.

The CBDISPLAY programme has been the first interactive project analysis programme to be widely used within the World Bank. The functions of data handling are too limited to do much more than the analysis of profitability which comes towards the end of the calculation of models, but CBDISPLAY is the immediate precursor of the COMPASS system, described below, in which it is incorporated.

MALAYSIA

NORTHWEST SELANGOR INTEGRATED AGRICULTURAL DEVELOPMENT PROJE



LEGEND

= MAXIMUM EXPECTED FALL IN BENEFITS IS 25.0%
= MAXIMUM EXPECTED RISE IN COSTS IS 30.0%

= TARGET YIELD IS 0.0%
BENEFIT SWITCHING POINT IS -39.1%
COST SWITCHING POINT IS 61.4%

b. <u>MANIP (FAO)</u>

A first version of this programme, called PDMP (Project Data Manipulation Package), was developed by the author in the FAO Investment Centre in 1979, on a Hewlett-Packard office computer (model HP9845B), using a very powerful BASIC language, which was, however, difficult to transpose to other systems. A new version was written in Pascal in 1981. (1) The technical specifications of the information system used (an ND-100 Norsk Data mini-computer) and of the Pascal compiler are given in Annex 1. The Users' Guide is reproduced in Annex 2. Two versions for IBM computer (VM/CMS system) are also available. A programme derived from the PDMP programme has also been developed - under the name MANIP - by the European Investment Bank (Luxembourg), in the language APL.

The functions of MANIP are similar to those of the CBDISPLAY programme described above, that is to say:

- Putting into store a number of variables (up to 1,500), a variable being defined by a name (up to 25 characters including spaces), and a series of values on a predefined number of years (maximum 50, plus year zero).

- Carrying out a number of operations with these variables, with the possibility of correction and table printing.

There are three major differences between MANIP and CBDISPLAY:

- MANIP is an interactive programme based on a menu system and messages displayed on the terminal screen, whereas CBDISPLAY is a programme based on a command language (a discussion of approaches is presented in Chapter IV, Section B).

- The MANIP variables include a year zero to represent a situation without project when it is constant. There is only one type of variable, whereas there are four in CBDISPLAY.

- There are many more calculation functions in MANIP.

The general organization of the programme (from the user's point of view) is set out below, and also indications on each of the main functions available:

(i) <u>General Organization</u>

The programme is organized in five modules, corresponding to the available functions, which are:

- Initial module: used once - and once only - at the beginning of each session, this module either creates a new data set, by specifying the name of the project, the name of the data set, the currency, the duration of the project, and possibly a number of variables - or

---

(1) Initial transposition was done by B. Samuelsson, finalization and later developments by the author.

provides access to a data set created at a previous session. A data set uses several files (six), whose names are formed automatically by the programme, based on a generic name (from 1 to 14 characters) given by the user.

- The calculation module (data processing).

- The printing module (data listing).

- The correction module (data editing).

- A function for printing - on the screen or on paper - the list of variables defined in a data set, with their code number. This code (and not the name) will be used as reference to the variables throughout the programme.

Once the initial module has been executed, the principal menu of the programme is displayed, allowing the user to select and execute any one of the other four modules, in any order and as often as necessary, until selection (in the same principal menu) of the "end of session" option is made.

(ii) Calculation Module

With this module a number of operations can be made on the variables, starting from the menu of available functions. Before reaching the menu of calculation functions, an initial menu enables the user to select the mode of operation, either manual or programmed. The options are:

Code = 1	Manual operation
Code = 2	Defining a sequence of operations (procedure)
Code = 3	Listing a predefined sequence of operations
Code = 4	Editing a predefined sequence of operations
Code = 5	Running a predefined sequence of operations
Code = 6	Printing the list of predefined sequences
Code = 7	End of processing

A sequence may include up to 50 operations. But an operation can consist in a call to another predefined sequence; it is thus possible to structure operations in logical blocks, and to execute many of them automatically (whatever the number of sequences, there is in the present version of the programme a maximum limit of 1,000 operations that can be defined in a data set).

There are 23 functions available. They are the following:

1. Display/print X
2. Rate of return on X
3. Incremental values over year zero
4. Cumulative values of X
5. Add X over A years
6. Advance X by A years
7. Delay X by A years
8. Interest on overdraft
9. AX + B
10. AXY + B
11. AX/Y + B
12. AX + BY + C
13. If X>0, AX + BY + C
14. If X<0, AX + BY + C
15. If X and Y>=0, AX + BY + C
16. Carry over if AX + BY + C<0
17. Depreciation on X
18. Debt service on X
19. Phasing a block of variables
20. Balance and switching values
21. Read from/write to another data set
22. Run a predefined sequence of operations
23. Print a table using a predefined format
24. End of processing

Most of the above-mentioned functions correspond to those described in detail in Chapter IV, Section C.2. In the above menu, X and Y correspond to code numbers of variables, and A, B and C to parameters. When a function is selected, the programme calls the necessary data. It also requests for the variable where the results should be stored. If the code number specified corresponds to a variable *not yet defined*, the user must enter the name of the new variable.

The comments below complete the description of Chapter IV.

- Function 1: displays the values of a variable without having to use the module for printing the table; it will, for example, be used as the last operation in a calculation sequence of cash-flow projections for a farm model, to see the final result of the calculations before deciding to print the tables or modify the data to test other values.

- Function 2: The calculation of the rate of return is made automatically on incremental values of the variable compared with year zero.

- Functions 3-7, 9-12 and 19: These functions can be specified in one operation to apply on blocks of variables, which, through a judicious organization, leads to a reduction in the number of operations to be specified. If, for example, it is intended to construct a model containing five crops, it will be desirable to introduce the five yields and the five variables containing the areas in consecutive variables. If the yields are in variables 1 to 5 and the areas in variables 6 to 10, one running of function 10 (with X=1, Y=6, A=1, B=0, on a block of 5) will make it possible to calculate the production of each crop (calculation using the annual mode).

- Function 17: Calculates only constant depreciation.

- Function 18: Debt service. The calculation of the interest component and the outstanding amount (at year end, after payment of the corresponding annual instalment) is optional. The loans are assumed to be disbursed at the beginning of the year and reimbursed at the end of the year.

- Function 20: It is possible to specify a list containing up to 50 variables in one operation. This is therefore the maximum of detail for the calculation of switching values.

- Function 21: This function enables the user to transfer a complete block of variables (names and values) from one data set to another. In the destination data set, the copied variables can be either inserted from a determined position, or copied after the last defined variable. This function means that a model can also - if necessary - be shared among several sets of files.

- Function 22: This function does not appear on the menu for manual operations, but can be used to call a predefined sequence of operations from another sequence. If the call is made in the middle of the first sequence, the second is executed, then the control comes back to the original sequence, which then continues.

- Function 23: In the manual mode, this function is not shown on the menu. It allows for the printing of tables corresponding, for example, to the results of calculations, to be included in a sequence.

- Function 24: This function returns control to the previous menu, and also terminates the list of operations in a sequence currently being definied.

Note: The graphic function corresponding to the command LOCUS of the programme CBDISPLAY, which figures in the first version of MANIP (PDMP programme) has not been maintained in the later versions, since experience has shown that it was not used.

(iii) Printing Module

It is possible to produce tables from a data set by selecting the variables and the years to be printed in a given table, which also receives a title and optionally a number. An option also enables the user to calculate and include the total by lines, corresponding for each variable to the interval of the years printed (that is, if the years printed are 1, 2, 3 and 10, the total will correspond to all the years from 1 to 10). When all the variables are constant for several consecutive years in the life of a project, the programme will print only one column (example: years 15-20; this only happens if all the years are requested).

Apart from the option allowing the user, without other specification, to print the whole data set, the format of a table must first of all be defined. This definition of a table can itself be printed, and also corrected, and of course run, as many times as necessary.

The programme allows for selection of the type of printout desired: anything for printout - tables, results of certain functions of the calculation module, list of variables, definition of tables, definition of sequence of operations - can be directed by the user either to the terminal screen or to a printer, or to a file on magnetic disk on the system.

The format of the tables (that is, the number of columns per line) will be decided by the programme according to the values of the variables to be printed, the number of decimal digits specified by the user, and the output unit chosen: 80 characters per line for the screen, 132 or 227 depending on the printer. When a disc file is used, it can be read later by the text-processing software available on the same system, which means it can be handled and printed like any other text (for example, inserting sub-titles or replacing decimal points by commas for a report in French or

Spanish). A table thus produced can also be easily inserted into the text of a report typed using the system.

(iv) Correction Module

This module is for changing the titles of a data set, the name or the values of a variable, and for adding or deleting variables.

To add variables, the user has the same functions available as for the definition of variables in the initial module, which correspond to the different methods of data generation set out in Chapter IV, Section C.2.e, that it is to say, automatic repetition, cyclical repetition, linear or sigmoid interpolation, use of percentages, exponential variation and definition by reference to another variable.

When one variable is deleted, the others are not renumbered (unlike what happens with the PDMP version). The code number of the variables does not change, the existing definitions of tables and sequences of operation remain valid. In practice, the possibility of deleting variables from a data set is of very limited interest, and the absence of this function from the programme (which would allow for a number of simplifications in the programming) would not really be a limitation for the user.

(v) Use of Predefined Sequences or Formats

When the user requests the definition of a sequence of operations or the format of a table, he must choose between two options:

- new definition;

- copying an existing definition.

An existing definition can be copied either from the same data set, or from another one. When such a definition is copied, all the code numbers of the variables can be modified by adding to them - or taking away - a constant factor, called displacement of address. This mechanism allows for the copying of a definition by adjusting it automatically to a new block of variables (in which the variables must of course have been created in the right order). It will, however, be noted that only the code numbers of the variables are changed. If a sequence of operations includes calls to other sequences or references of output tables, these operations are not changed and will perhaps need to be corrected "manually".

The above mechanism considerably reduces the work required in the detailed specification of operations of calculating a project. It can be used in two ways:

- It is possible to constitute a library of "standard files" corresponding to typical applications; for any given application, a set of files contains variables, with nil values, and definitions of sequences and tables. An example of this technique is shown in Appendix 3 of Annex 2.

- When a project composed of several farm models is studied, the types of analysis and presentation desired will in general be the same for all the models. The variables, the sequences of operations and the tables corresponding to one model can be defined and copied as often as necessary to construct the next models. The nine models shown in the credit project mentioned (ref. 48) were prepared in this way.

(vi) Example of Use

The following example - shown in detail in Annex 4 - shows how MANIP can be used to construct a model.

Let us suppose that the following information is available for a farm model (in the form of "variables", that is to say, series of annual data):

1) Total value of production
2) Other income
3) Investments
4) Cost of production less labour
5) Cost of wage-earning labour force
6) Value of family labour force
7) Household consumption.

The intention is to study the methods of financing this farm by long-term credit to finance the investments, and short-term credit to finance production costs (less labour force and wage-earning labour force).

The lines of a summary table may be constructed step by step in the following way:

1) Add the variables 1 and 2 (operation 12 in the menu, with X = 1, Y = 2, A = 1, B = 1, C = 0) to create the variable 8 = total income.

2) Using operation 12 or operation 20 (balance), add the variables 4 (costs of production) and 5 (wage-earning labour force), applying to each of them a weighting coefficient corresponding to the calculation of interest on a short-term loan: a rate of 15% per annum and a loan of six months will be translated by, for example, a coefficient of 0.075. Operation 20, by which more than two variables (up to 50) can be added together, is preferable here, since it will be easier to correct the model by adding other variables or changing the coefficients. Result: variable 9 = interest on seasonal credit.

3) Using operation 20, the sum of the variables 3, 4, 5 and 6 can be calculated to create the variable 10, or total expenditure (before financing).

4) The difference between variables 8 (total income) and 10 (total expenditure) is calculated (operation 12, AX + BY + C, with X = 8, Y = 10, A = 1, B = -1, C = 0), to give the variable 11: net benefit before financing.

5) And so on ...

The complete example presented in Annex 4 (in which the variables do not have the same code numbers, since they were created in a different order) then shows how the user may choose to calculate the amount of a long-term loan as a percentage of investment costs, the difference being financed by a contribution by the farmer; how the debt service can be calculated in constant terms; how to apply to it a deflation factor (to reflect the "actual" weight of the reimbursements in a model expressed in a constant monetary unit); and how the benefit after financing is adjusted by the values of household consumption and family work to obtain a projection of monetary income.

c. Pico - MANIP: Programmes on Pocket Computers

In view of the growing power of programmable calculators, it is possible to envisage the development of computing programmes, certainly with lower performance than those mentioned above, but able to render meaningful services to teams working in the field, far from the facilities available at their organizations headquarters.

From the MANIP programme, which is very modular, essential functions can be extracted which can be put into operation on such machines.

By doing this, the author developed a set of four programmes functioning with a Hewlett-Packard calculator of the HP 41 CV type, equipped with a reader of magnetic cards and a printer. The programmes - presented in detail in Annex 3 - work on a data structure identical to MANIP: the variable. For obvious reasons of memory capacity, the names of the variables are limited to six characters. Beyond the life of the project, it is necessary to specify the "maximum period of development", that is to say, the number of years during which the data continue to change. Only these years will be stored in the machine. A variable can be either a data series supplied to a programme, or a series of results. The variables can be stored on magnetic cards, which allows them to be passed from one programme to another. It would be possible, by modifying the programmes slightly, to use also the digital recorder with cassettes now available for this calculator.

The programmes are organized, as with MANIP, according to a system of menus and questions. The main menu for each programme is based on the selection of keyboard strokes to which the corresponding functions are allocated, and each module of the programme follows in a consenquential way with a system of questions and answers.

The four programmes composing the system are as follows:

(i) ROR-PV (for "rate of return - present values")

The programme enables the user, after having specified the period of analysis, the maximum period of development, and the discount rate:

- To enter, one after the other, a number of variables, whether using the keyboard or from magnetic cards. The variable which has just been entered can be multiplied by a constant, visualized or printed, corrected, and stored on magnetic card.

- On the number of variables entered, to calculate and print (or to display on the screen):

    - The balance (algebraic sum of the variables), which can be stored on magnetic card.

    - The incremental balance (compared with year zero).

    - The rate of return of the incremental balance.

    - For each variable, for the balance and for the projection of the balance in year zero (situation without project), the present values and the switching values (defined in Chapter IV, Section C.2.c).

- To correct the data (change to be made to such and such a variable in such and such a year), so as to make the above calculations all over again.

For a maximum development period of 10 years, the maximum number of variables is 40. The number drops to 30 if the development period is longer than 20 years.

(ii) PHASE

This programme allows phasing mode calculation to be performed as described in Chapter IV, Section C.2.b. It works on one unit variable, for which two data series must be specified: with and without project. The period of analysis is limited to 20 years (plus year zero), and the phasing variable is limited to ten years (which in practice is more than enough).

(iii) DEBT

This programme calculates the service of a specific loan in a variable, according to the description in Chapter IV, Section C.2.d. (the only difference being that the stored values in the variable "loans" are automatically treated as a single loan; a test prevents the specification of a grace period, which would be translated by a first reimbursement before the end of the disbursements).

(iv) TS (Time Series)

This programme can generate variables, which will later be transferred to other programmes using magnetic cards, depending on the functions of cyclical repetition, linear or sigmoid interpolation, exponential growth, and use of percentages (see Chapter IV, Section C.2.e).

2. Second Approach: Detailed Structuring of Data

a. Predecessors

The detailed structuring of data being the most immediate approach (it consists in transcribing what the users do by hand in a programme), the first programmes developed for agricultural project analysis - both at the World Bank and in the FAO Investment Centre - followed this approach. These different programmes, now practically abandoned, were a major contribution to the development of the concepts set out in the previous chapter.

The APAS (Agricultural Projects Appraisal System) programme was developed by the World Bank in 1977-1978. It was based on a hierarchical structure (crops - models - project), but had a number of major disadvantages, as a result of which it was used very little. The main characteristic of the programme was that it functioned by batch processing, with the aid of punch cards. The project analyst had to transcribe his data on forms, observing a restrictive codification, send these forms to the punching centre, retrieve the punched cards, and submit them to the calculating centre; the results were only available after a fairly long time, say several hours. The programme had in fact been written by "traditional" programmers, with insufficient relations with future users. This was evident even in the specifications of the programme: the calculating mechanism of lagged series (phasing) did not figure in the initial version. As it was added later - at the express request of users - the result was a method of data specification, for this calculation, particularly impractical and likely to lead to errors. (Since this experience, programmes for project analysis have been planned in the World Bank by users, and no longer by the computer centre!).

In the FAO Investment Centre, two programmes may be mentioned:

- The ROR-Cash-flow programme (1) was the first one available. Although it is very limited in its possibilities, it was and still is very popular, since it meets a specific demand. The first part of the programme enables three lines of data, corresponding to investments, operating costs and revenues of a project, to be entered. The data can be printed and corrected; the situation without project is represented by the year zero, and it is possible to specify a certain evolution in this situation over a definite number of years (growth or reduction at a certain rate). The programme can calculate present value and rate of return of the project, and can make a number of sensitivity analyses; an example is given in Annex 2 (Appendix 3, Table Annex 4). The second part enables the user to specify, after the three lines of data mentioned, several credit operations, and also lines of other costs or benefits, and to produce a standardized table of cash-flow projections (for farm models).

- The FARMOD programme (2) enabled the user to specify a list of crops and a list of inputs, to define for each crop - for each year - the yield and consumption of each input, to define a cropping pattern, and other costs and benefits not proportionate to the crops, and to calculate total costs and benefits of the model (in quantity or value). The programme - interactive and commanded by a menu system - had the major defect of excessive rigidity: the list of inputs and crops, once specified, could no longer be changed; and initial data had to be entered altogether, without interruption, which proved to be a considerable disadvantage in practice. Also, the programme was conceived to treat one model only (project or farm model), not allowing for the specification of several models which could then be aggregated. The programme is no longer used.

The above two programmes (ROR-Cash-flow and FARMOD) were written in HP-Basic, for the desk top computer HP-9845B.

b. <u>The COMPASS System (World Bank)</u>

The Computerized Project Analysis Support System - COMPASS - is a system of programmes developed since 1980 by the World Bank, under the guidance of G. Temple. The system is based on a "core", which is to some extent an improved version of CBDISPLAY, with the addition of a number of specialized modules, some of which are still being developed.

---

(1) The return analysis part was done by the author, the cash-flow projections part by F. Vita.

(2) Developed by F. Vita, with the author's participation in development of specifications.

The system therefore aims to become in due course an integrated system comparable to that proposed in Chapter IV, Section G.2, with the major restriction that, until now, the system has been conceived exclusively around a command language and the menu approach has not been used.

Three parts of the system are now operational:

- The "core", which controls the files and the terminals (ref. 41) and by which operations of the "table manipulation" type can be carried out (see description of CBDISPLAY), with most of the functions available in the MANIP programme (phasing and debt service in particular).

- The COSTAB module (ref. 40), in service since 1981 in the World Bank, prepares highly detailed tables of project costs, based on elements defined by their unit cost, their quantities, and the category to which they belong. The tables produced follow the standardized format which the World Bank wants to bring into general use in its appraisal reports.

- The FARMOD (FARM MODEL PROCESSOR - ref. 39) module, to which the finishing touches were being put at the beginning of 1984, allows the user to define crops, to combine them in farm models, and to combine the latter in project components, or sub-areas, and at project level. It puts into practice the concepts set out in Chapter IV, Section D, making them even more flexible: The "crop" and "farm" levels can change their names (activity and enterprise) to define more general actions (similar to the activities and plans of the generalized model).

In the case of both COSTAB and FARMOD, the user defines, with the aid of the editor which forms part of the system, a procedure that will later be treated by the appropriate module. The procedure contains instructions concerning the construction of the model, and local data; it can also refer to external data, that is to say, common to the whole COMPASS system, in the context, of course, of a specific data set. (1) In the same way it is possible, in the procedure, to specify that certain results must be transferred to COMPASS, to be used later by other modules.

The treatment of a procedure by the relevant processor is done in two phases:

- A compilation phase, which analyses the text of the procedure to detect errors, and then, if there are no errors, translates the procedure into an intermediate code.

- A running phase, which runs the procedure in intermediate code and produces the requested reports; a compiled procedure can of course be re-run without further compilation. The use of external variables is a means of changing some data of the model without the need to recompile the relevant procedure.

---

(1) See Chapter IV, Section G.2, concepts of local and external variables.

There is an instruction (READ FROM) which leads to the combination of an existing procedure with a new procedure. It is therefore possible to construct the models in stages which results in a complete project, in the form of procedures which call each other: for example, a procedure for prices, one for each crop model, one for each farm model, etc.

The most serious disadvantage of COMPASS - and it remains to be seen whether in practice it will really prevent expansion of its use - is that it is a very large programme (more than 100,000 lines of code in FORTRAN, excluding the comments and instructions for use, for COMPASS, COSTAB and FARMOD), which therefore needs powerful systems, and of which the transfer from one system to another, and then the maintenance, are likely to be difficult.

As well as the modules mentioned above, at least two others are envisaged for the future, one for financial analyses (FINAL) and one for animal husbandry projects (based on a dynamic herd model).

3. Third Approach: General Structuring of Data

a. The MADS Programme (OECD)

In the context of Technical Cooperation programmes of OECD (Organization for Economic Cooperation and Development - Paris), a joint activity covering the use of computers in agriculture was carried out by several European countries in 1978, following the development by the same organization of a handbook on irrigation project analysis. The proposed programme (1) known as MADS (Multipurpose Agricultural Data System), was based on the concepts of commodity, activity and plan (as set out in Chapter IV, Section E.2.a. - ref. 42).

Compared with the concepts described in Chapter IV, which were developed from the MADS programme, the most marked differences are as follows:

- The programme was conceived to be used for batch processing; the user interface is done by punch cards.

- The types of data are limited to the commodity (which can be simple or composite), the activity and the plan. A fourth type, called resource, is (like the composite commodity) a linear combination of commodities. Whereas the composite commodity is intended to reduce the number of coefficients to be written into the definition of an activity, the resource is used to define an aggregate of commodities which is considered as a whole, and on which it is desired that a number of calculations should be made. The notion of resource in fact proved difficult to understand and gave rise to many discussions within the OECD working party (at meetings at which the author was present), the more so since use of a composite commodity as component in the definition of a resource is not equivalent to the use of components of this commodity.

---

(1) The programme was conceived by J.M. Boussard, working in INRA (National Institute for Agricultural Research, Paris) as an OECD consultant.

- Time is treated in an extremely detailed and essentially static way, both on the level of specification of data and on that of format of output tables. The consumptions and productions of commodities by an activity are in fact defined in a static way (a given situation and not a time series of values), but specifying by dates (therefore to the nearest day) the period during which this production or this consumption occurs. The programme distinguishes between annual activities, for which production or consumption coefficients are assumed constant, and perennial activities, which define the ensemble of production and consumption which can be different from one year to another (example: cultivation of fruit trees); perennial activities are divided into "sections" and each is defined in a way similar to an annual activity. In the same way, at the level of definition of plans, the way in which variations in the plan composition during its duration must be specified is very cumbersome. The components of a plan (activities or other plans) enter into a plan in a given year, with a given level and a given age. A component remains in the plan with this level, either until the end of the plan, or until the year when this same component again comes into the plan at a different level; in other words, the component in the plan must be specified as often as its level changes.

- On the level of the data processing implementation, the programme has been conceived - because the specifications were prepared at a time when available systems did not have the capacities they have today - to function with a very limited central memory capacity. This is reflected in an excessive complexity of implementation, using a great many magnetic disc files (9 permanent files and 17 temporary files). The result is that processing becomes extremely expensive if the work is on a large system, and excessively slow if it is on a small system without any hard disc. The system is in fact based on the creation of a master file containing the data, and every command (for a table corresponding to determined calculations) involves a complete exploration of the data base to run the relevant calculations.

The MADS project was abandoned before the programme had been completely developed. The major merit of the operation has been, as already mentioned, the development of fundamental concepts of commodity, activity and plan, which were taken up again in the programmes presented below (of which some retain, for this reason, the original name of MADS).

b. The MADS II Programme (FAO)

This programme was developed jointly by the Investment Centre and the Development Policy Studies and Training Service (ESPT), FAO, (1) using the BASIC language of an HP-9845B computer (see specifications in Annex 1). The programme borrows from the above-mentioned MADS programme the fundamental concepts of commodity, activity and plan, but has essential differences compared with MADS (ref. 43):

(1) Specifications developed by the author and by Carlo Cappi. Initial programme of the calculation and output tables by C. Cappi. Input and correction of data by F. Vita. Finalization by the author.

- The programme is entirely interactive, using a menu system based on the keyboard and screen of the (monostation) computer used.

- The basic structure of the data over time is the annual time series. Year zero is included, and could be used to represent the situation without project if the latter could be considered constant (bearing in mind that this assumes that not only the technical coefficients but also the prices should be projected as constant). A more detailed analysis could also be specified for a limited number of commodities, for which consumption or production could be specified by sub-periods in the year (the sub-periods, maximum 24, are defined in the same unique way for all the commodities).

- The notion of annual activity and perennial activity was abandoned; all activities are treated in the same way; the mode of calculation corresponding to the nature of the activity (annual or phasing) is specified when the plans are defined.

- The type investment has been introduced.

- The plans are of three types: plan of activities, plan of investments, or plan of plans.

- The type resource has been abandoned. When a plan is called, a list of standard calculations is made, and the user selects the results he wants printed (the type resource, in MADS, was used to specify the breakdown of the results of a calculation requested by specific commands).

The calculations made automatically by the programme for a given plan are the following:

- Consumption, production and balance of each commodity. If the plan includes investments, calculation of investments, their maintenance cost, their replacement and the provision cost for physical contingencies.

- The results are in quantity and in value; they could be in total values or in incremental values in relation to year zero.

- If certain commodities have been specified by sub-periods, the programme calculates the quantities produced and/or consumed per annum, per sub-period, and the annual totals of surpluses and deficits: the deficit in any given year is the sum of the production/consumption balances for the sub-periods when this balance is negative, and similarly the surplus is the sum of positive balances. For example, if the product is family work, the annual deficit represents the demand for family labour not employed in a slack period.

- The rate of return of a plan can, at will, be calculated on the overall balance (on incremental values compared with year zero).

The structure of results is basically the same as that of the variables in the MANIP programme described above, that is to say, annual time series of values, with a name.

The MADS II programme includes a module with which it is possible, when the calculations of a plan have been made, to put all or part of the results into data files compatible with the PDMP programme (the first version of MANIP, on the same computer as the MADS II programme) which means that downstream of the programme, all the calculations which are not included in MADS II, like, for example, the sensitivity analysis by calculation of switching values, can be performed using PDMP.

The programme is organized, from the user's point of view, in seven modules:

START: Opens a session, either for new data, or by reading data already stored. The data can come from a file named by the user(created by the STORE module below), or can be found in the working files associated with the programme (in the first case, the data are copied in the working files).

INPUT: Defines new components of the data base (commodities, activities, investments, plans), and defines sub-periods and corresponding commodities.

EDIT: Modifies the values of already defined components. The input of time series has functions of automatic repetition when the value remains constant, of linear interpolation and exponential growth. It is not possible to suppress a component of the data base, since the programme has no internal arrangement that can, for example, check that the commodity to be suppressed has also been used to define an activity and to suppress it automatically from the definition of that activity.

PRINTOUT: Prints the data base, completely or selectively, and also a catalogue of the data structure showing the list of components and certain indicators (commodity consumed or produced, activity calculated annually or with phasing in a plan, etc.).

STORE: Transfers the entire data base - as at that moment - scattered in working files, to a permanent file whose name is chosen by the user.

CALCULATION: Makes all the standard calculations on the chosen plan, and if required, prints the results and/or transfers some of them to a file allowing for "à la carte" calculations (using the PDMP programme).

DATA MANIPULATION: Accesses data lines created by the preceding model, to effect all the operations described in A.1.b (MANIP-PDMP programme).

The programme has limitations as regards the size of the data base that can be manipulated. If the duration of the project (period of analysis) is 20 years, the limits are:

- 44 commodities;

- 20 investments;

- 20 components in a plan, and 20 commodities in an activity;

- 10 commodities specifiable per sub-period (12 sub-periods).

Despite these limitations, and the absence of some components that could usefully form part of such a programme (transfer activities, credit operations), the programme has proved very powerful, provided the user has completely assimilated the concepts of commodity, activity and plan, and properly understood the direct relation between the definition of these project components and the precise nature of the calculations to be made and the result tables.

c. <u>The DASI Programme (FAO)</u>

This programme (whose name is an abbreviated form of "Data Analysis and Simulation Programme") was developed by the Development Policy Studies and Training Service (ESPT) in FAO. (1) This is a programme derived from MADS II, described above, written in FORTRAN and of which two versions exist, one for the IBM 370 computer, and the other for the Apple II microcomputer. DASI is in fact composed of seven separate programmes, corresponding to the logical organization of the programme, which are called by name by the user (see below, and ref. 44).

The major differences between MADS II and DASI are as follows:

- The data structures are the same: commodities, activities, investments, plans. The sub-periods within the year have been suppressed; it is enough, where necessary, to define a commodity as often as sub-periods are required (example: irrigation water in the dry season and in the rainy season). It is possible to specify as components of the same plan the activities, investments, other plans, even single commodities; there is therefore only one type of plan.

- An option allows the possibility of including or excluding year zero in the time series, to be used, if necessary, to represent the situation without project when it is assumed constant.

- The variety of "à la carte" calculations that can be made after the calculations of a plan, is limited to the definition of aggregates (weighted sum of result variables, that is to say, commodities consumed, or produced, or investments), to the definition of the balance of a project as composed of the sum of a number of single variables or aggregates, and to the possibility of calculating for each of these components the present value and switching values and of making sensitivity tests calculating present value and rate of return of the project when some variables increase or fall by given percentages.

---

(1) By C. Cappi.

- The method of entering data to create a data set corresponding to a project is not based on a menu system, but on the preparation of a text (with a programme of text editing) in which the data will be specified according to certain rules as to their format and their order, and also through the use of a number of key words in the text (such as COMMODITY, ACTIVITY, PLAN). Here again we find the "modelization language" approach described in Chapter IV, Section B.3. Two files of this type are necessary:

  - The first defines the data base, and contains five sections: headings, commodities, investments, activities, plans.

  - The second defines the optional calculations which can be made after the standard calculations of a plan, with three sections: aggregates, components for calculation of rate of return and switching values, and sensitivity analysis by variations in percentages.

The organization of the system in seven programmes is as follows:

1) CREATE: The programme reads the data from the first text file, and transforms it into a set of data files for its own future reference. To correct the data, it is only necessary to change the text file and repeat the CREATE programme.

2) PRTDATA: Prints the data base.

3) QUANTY: For a plan specified by the user (in response to messages displayed by the programme), this programme calculates the quantities of commodities consumed or produced by the plan, and the investments.

4) PRTQUA: Prints results in quantities calculated by the QUANTY programme.

5) COMVAL: Multiplies the quantities previously calculated by prices, to calculate the values.

6) PRTVAL: Prints the costs and benefits calculated by the COMVAL programme.

7) AGGREG: Using the specifications of the second text file, calculates the aggregates, and makes analyses of rate of return and sensitivity.

Unlike MADS II, the constraints in the programme as to the size of the problem which can be treated (linked to the size of the available memory) are expressed globally by two parameters only, the duration of the maximum life of the project and total maximum number of elements (commodities, activities, investments and plans) that can be defined.

d. The MADS III Programme

This programme was developed by the author (1) in FAO, Rome, on an ND-100 computer in the programming language Pascal (see specifications of the system in Annex 1). The general specifications are based on concepts outlined in Chapter IV, Section E and are based on the experience acquired in preparing the MADS, MADS II and DASI programmes described above. The user's guide is reproduced in Annex 5.

(i) General Characteristics

The MADS III programme takes up the fundamental concepts of MADS, that is to say, the data types commodity, activity and plan. It also includes additional types described in Chapter IV, that is to say, the investment, credit operation, transfer activity and aggregate.

The basis of calculation of a credit operation can be, among the results of a plan, either a commodity or an investment, or an aggregate of commodities or investments.

An aggregate may contain as components commodities, investments, credit operations, or other aggregates. In the case of credit operations, the result at the level of the aggregate will be the net financing (amount of the loan less debt service). For the purposes of simplification, the following restrictions were introduced in the definition of transfers: a transfer may be defined only on the basis of quantities (not values); in case of compensation (transfer back), the "source" can only be a commodity; in the absence of compensation, the source can also be an investment; the use of aggregates as source was excluded.

The sub-periods in the year were suppressed, as in the DASI programme, since they could be replaced by definitions of commodities corresponding to each sub-period. Year zero is optional in a data set.

The approach of the text file for entering the initial data was kept, as in the DASI programme. The programme - once the text file is created - remains entirely interactive through the medium of menus and questions.

The results of the standard calculation of a plan can, as in the case of MADS II, be transferred to data files which can be used for making supplementary calculations or producing tables with the MANIP programme.

(ii) Organization of the Programme

From the user's point of view, the programme is composed of five modules:

- The initial module, which is run once at the beginning of each new session: by entering a generic file name, the user can either start working on the basis of an existing data set, or prepare files for a new one.

---

(1) In collaboration with S. Katz, who did a large percentage of the actual programming.

- The module for creating a data base, which analyses a text file (previously prepared) and uses it to create the project data base.

- The module for printing the data base.

- The calculation module.

- The module for correcting the data base.

(iii) Creation of a Data Base

This module is automatically called by the initial module when a new set of files is created. It is also called through the main menu when the user wants to replace his old data base with new data, and process a new text file (called source file).

The module calls the name of the source file to be used, and processes this source file to create a data base containing the definition of commodities, activities, plans, etc., of the project. The data base can be corrected, if necessary, in two ways:

- Certain limited corrections are possible in an interactive way, using the relevant module of the programme (see below).

- Whatever correction is necessary, it is always possible to "come out of" the MADS III programme, to correct the source text (using the text editor programme of the system), and to call MADS III again, and to then run the module of data base creation with the modified source file. This method is the only possible one for corrections not allowed in the programme itself, and particularly when it is intended to modify the structure of the data base by adding or taking away components (commodities, activities, plans, etc.).

When the source file is being created, it is necessary to follow a certain format for the data. In order to limit errors, it is possible to copy a standard text from the system which already contains the "skeleton, and indications of how to enter the data. This file can then be modified (through entering data to create the source file for the user (see example in Annex 5).

At the time of processing the source file, the module for creating the data base will carry out a number of checks on the possibility of "abnormal" specification data at the time of definition of aggregates, credit operations, and transfer activities:

- An aggregate can, depending on the option chosen, be calculated on the basis of data expressed in quantities and in values, or on values alone. When an aggregate is based on quantities, it is not valid for it to include as a component an aggregate based solely on values. The credit operations, consequently, can only be components of aggregates based on values.

- A loan being destined a priori for financing costs, the basis of calculation of a credit operation - if this is an aggregate - should only include commodities or investments, and no other types.

(iv) Printing the Data Base

This module can produce a "catalogue" of the data base, that is to say, a list of components with some of their characteristics, and can print all or part of the project data, in a format suitable for direct insertion in a report.

(v) Calculation Module

Calculations are made altogether on quantities of commodities and investments for all the plans defined in the data base, and the corresponding results are stored in a file. These results will have to be recalculated each time the module creating the data base is run again, or when the data concerning the levels of components of an activity or a plan are modified by the module for correcting the data base (described below).

The other calculations are only made for a given plan, selected by the user. They concern results in value, credit operations, aggregates, and the net value of the plan.

This calculation mechanism leads to correct results even when the commodities whose prices change with time are the subject of a calculation of aggregation following the phasing mode: only quantities are aggregated in this way, the multiplication by price series being done at the level of the plan for which results have been requested.

The module includes five sub-modules:

- Calculation of quantities (called automatically when necessary).

- Complementary calculations for a given plan.

- Printing of results.

- Rate of return calculations.

- Transfer of results to MANIP.

The nature of the calculations made, and the rate of return and sensitivity analysis calculations correspond to the specifications developed in Chapter IV, Section E.3, with the following details as regards the credit operations:

- When the basis of calculation of the amount of the loan is a commodity, the values corresponding to costs (the purpose of the loan being to finance these costs) will alone be taken into account. If the commodity if exclusively consumed, the values will be used directly. If the commodity is consumed, produced and transferred, only the years for which the balance is negative, and for which therefore finance is needed, will be taken into account.

- When the basis of calculation is an investment, only the cost of the initial investment and the provision for physical contingencies will be taken into account. By convention, the other costs calculated by the investment operation (see below) will not be taken into account in the calculation of the need for financing.

- When a credit operation is defined in a plan (for example, a farm model) which is itself aggregated in another plan (for example, a project) using the phasing mode, the calculation of aggregation can be done directly on the results of the credit operation only if the price of the commodities or the investments serving as a basis for calculation of the loan are constant; in the opposite case, the calculation would have to be made differently, and this possibility does not figure in the programme. The consequence is that the credit operations will be ignored at the time of calculating the aggregation of the components of a plan if the following conditions are met:

  - There is at least one credit operation which should be aggregated using the phasing mode.

  - This credit operation is based on commodities/investments of which at least one has a price which is not constant.

It should be remembered that the results, in the form of tables of time series in quantities and in value (year by year), will include the following items:

- Commodities consumed.
- Commodities produced.
- Commodities transferred.
- Balance of commodities.
- Investments and their replacement.
- Physical contingencies on investments.
- Maintenance of investments.
- Residual value of investments.
- Loans.
- Debt service (for each loan, and total).
- Interest and principal components of debt service (in option).
- Outstanding amount (for each loan).
- Aggregates.
- Net value of the plan (balance).

The calculation of profitability (rate of return and present value, and present values and switching values of each component) will be made as required, either on a plan or on an aggregate.

Once a plan has been calculated, the other four modules can be run in any order. Internal tests prevent errors. It is, for example, only possible to request at the opening of a new session the print-out of results corresponding to the last plan calculated at a preceding session, if in the interval there has been no change in content of the data base.

The results of a plan can be accumulated, in whole or in part, in files of the type of those of the MANIP programme, created from the same generic names as those of the MADS files now being used: if, for example, the user is working with MADS with a generic file name such as SENEGAL-1, the same name will be given in reply to questions on the MANIP programme when the latter programme is used to make supplementary calculations.

(vi) Correction of Data Base

This module can, without coming out of the MADS III programme, modify values or characteristics of the components of the data base, without having to recreate it entirely. It is thus possible to change quite easily, through a menu system, the value of a yield or the rate of interest on a loan, and to recalculate the results of a plan. This gives all the necessary flexibility to the analyst for adjusting the model and testing the sensitivity to changes in the data.

(vii) Example of Use

The simplified example below is intended to illustrate the different concepts of MADS III.

The first section of data, in the source file, contains five lines corresponding to the name of the project, the name of the data set, the name of the monetary unit, the base year - 0 or 1 - and the duration of the project, and the discount rate.

Example:

```
Mali - Irrigation project
Financial analysis
CFAF
1  20
12.0
```

The following stage is the identification of the variables corresponding to the different types of data.

Let us assume that the analysis concerns an area to be developed for irrigation. The area is at present cultivated by recession flood rice, and could after development be cultivated partly with erect rice, followed by maize over one-third of the area, and partly with floating rice. The costs of production are expressed in inputs per ha (in value) and in days of work.

The costs of the project include the following items: investment, maintenance costs, costs of pumping for the full parameter, and costs of extension services.

The following commodities will first have to be defined: rice, maize, inputs, work, pumping and extension services.

The commodities section of the source file will look like this:

```
COMMODITIES
Rice                    tons
A  1   180000
Maize                   tons
A  1   230000
Inputs                  FCFA 1,000
A  1     1000
Labour                  man-day
A  1      700
Pumping                 m3
A  1        5
Extension services      FCFA millions
A  1  1000000
ENDATA
```

Each commodity is defined by a name, a unit, and a price series. The letter A indicates that the following figure represents the number of values supplied (the last value is repeated until the year 20), and the following are the values themselves.

Each crop will then be described in the form of an activity:

```
ACTIVITIES
Irrigated rice                  Ha
Rice                            P
A   4    2   2.5   3    3.5
Inputs                          C
A   4   35  40     45   52
Labour                          C
A   1        118
END
Floating rice                   Ha
Rice                            P
A   3   1     1.5   2
Inputs                          C
A   3        40     50   60
Work                            C
A   1        98
END
Rice without project
Rice                            P
A   1          0.5
Inputs                          C
A   1        11.4
Labour                          C
A   1        54
END
Maize (project)                 Ha
Maize                           P
A   3          1     1.5     2
Inputs                          C
A   3        20    30      35
Labour                          C
A   1       118
END
ENDATA
```

The lettre C or P, opposite the components of an activity, indicates whether the commodity is consumed or produced.

The following section is that of investments:

```
INVESTMENTS
Land development    FCFA millions
A     1     1000000
50    3     5     10     0
ENDATA
```

The investment will be given in value (in the definition of the plan), and the data of the third line correspond to its duration (50 years), to the time between the investment and the first maintenance costs (3 years), to the maintenance cost (5% of the investment cost each year), and to the percentage to be applied to the cost by way of provision for physical contingencies (10%). The last figure (zero in the example) is the residual value of the investment (in percentage, at the end of its useful life fixed by the first parameter).

The following section, that of aggregates, allows the user to define, for example, the total production of cereals (rice plus maize). The aggregate perimeter will serve to calculate an economic rate of return that excludes the costs of labour and credit (it is assumed, in this simplified example, that the other prices are the same for economic and financial analyses).

```
AGGREGATES
Cereals               tons
QUANT
Rice                  1
Maize                 1
END
PERIMETER             U
VALUES
Rice                  1
Maize                 1
Land development      1
Inputs                1
Extension services    1
Pumping               1
END
ENDATA
```

The mechanism of transfer activity can be used to calculate the costs of pumping, if it is estimated that they are proportionate to production. Assuming that the pumping is only necessary for rice, and that it takes 1,200 m3 per ton of erect rice and 2,000 m3 per ton of floating rice, at a cost of FCAF 5 per m3, the following transfer activity could be defined:

```
TRANSFERS
Pumping water              m3
SURPWITHOUT
Rice                       Pumping
ENDATA
```

The quantity of pumping water will be calculated on the basis of rice production, without transfer back. The quantity of the commodity "pumping" will depend on the specification of this transfer in a plan. Since the choice has been made to use the same commodity for two crops whose water requirements differ, it will be necessary to specify two separate plans. An alternative would have been to define two commodities (floating rice and erect rice) and an aggregate to obtain total rice production. The following section describes credit operations. Let us suppose that inputs are financed partly by seasonal credit:

```
LOANS
Seasonal credit               FCFA
0       1       13.5
PAID                          EQUAL.INST
Inputs                        SEPARATE
ENDATA
```

The three figures in the second line indicate the duration of the grace period, the number of annual repayments, and the rate of interest. The values 0 and 1 correspond to an annual loan. The following line indicates that the interest is paid during the grace period and the reimbursements are made in constant annual payments (the specifications are necessary but have no effect in the case of annual credit). Inputs is the name of the commodity used to calculate the amount of the loan, and SEPARATE means that annual values should be considered as the basis of several different loans. The sections "plans" completes the data set:

```
PLANS
Erect rice                     Area
Irrigated rice                 PT
A    4    0     100     250    450
Pumping water                  TR
A    1    -1200
END

Lowland rice                   Area
Floating rice                  PT
A    4    0     75      150    250
Pumping water                  TR
A    1    -2000
END

Project                        Area
Erect rice                     A
A    1    1
Lowland rice                   A
A    1    1
Maize                          PT
A    5    0     20      60      120     150
Seasonal credit                A
A    6    0.60     0.60     0.50     0.40     0.30     0.0
Rice without project           A
A    1    -200
Extension services             A
A    1    -4
Land development               A
A    6    396.6    486.4    177.7    64.7    39.8    0
END
ENDATA
```

The plans for erect rice and lowland rice were necessary, as mentioned above, to calculate needs in pumping water (the "levels" -1,200 and -2,000 are negative to indicate a cost). It will be noted that identifiers should be unique: we have therefore used "lowland rice" for the plan, "floating rice" for the activity and "rice" for the commodity. The parameter PT means that the calculation has been made according to the phasing method and that levels of activity (the areas of each crop) are in a total number of units.

The project plan includes other plans (erect rice and lowland rice), two activities (maize and rice without project, which will be deducted since its level is negative), an investment (land development, specified in value, and in FCFA millions), a credit operation (the seasonal credit will cover 60% of the cost of inputs, decreasing later and being cancelled as from year 6), and a commodity (extension services, where one line of costs can be directly specified).

The calculation of the project plan will show rice and maize productions, and the total of cereals, as well as the different costs. All the results will be in incremental value compared with the situation without project, and the balance of the plan will be used for the rate of return calculation. Below is a copy of the source file. The file is constituted by inserting data into a predefined standard file, which also provides instructions on how the data should be presented. Appendix 3 of Annex 5 also shows the same example, but with names in French, as well as the corresponding tables of data and results produced by the MADS III programme.

COPY OF THE SOURCE FILE

```
                                Mali - Irrigation project
                                Financial analysis
                                FCFA
                                1  20
                                12.0
********************************************
COMMODITIES
Rice                                tons
A  1 180000
Maize                               tons
A  1  230000
Inputs                              FCFA 1,000
A  1    1000
Labour                              man-days
A  1      700
Pumping                             m3
A  1        5
Extension services                  millions
A  1  1000000
ENDATA
**********************************************
ACTIVITIES
Irrigated rice                        ha
Rice                                P
A   4    2   2.5   3    3.5
Inputs                              C
A   4   35   40   45    52
Labour                              C
A   1        118
END

Floating rice                       ha
Rice                                P
A   3   1    1.5   2
Inputs                              C
A   3        40    50    60
Labour                               C
A   1        98
END

Rice without project                ha
Rice                                P
A   1        0.5
Inputs                              C
A   1       11.4
Labour                              C
A   1       54
END

Maize (project)                     ha
Maize                               P
A   3        1     1.5    2
Inputs                              C
A   3       20     30     35
Labour                              C
A   1      118
```

```
END
ENDATA
************************************
INVESTMENTS
Land development                millions
A     1     1000000
50    3     5     10     0
ENDATA
*************************************
AGGREGATES
Cereals                        tons
QUANT
Rice                             1
Maize                            1
END
PERIMETRE                      U
VALUES
Rice                           1
Maize                          1
Land development               1
Inputs                         1
Extension services             1
Pumping                        1
END
ENDATA
*************************************
TRANSFERS
Pumping water                  m3
SURPWITHOUT
Rice                           Pumping
ENDATA
*************************************
LOANS
Seasonal credit                FCFA
0       1      13.5
PAID                           EQUAL.INST
Inputs                         SEPARATE
ENDATA
*************************************
PLANS

Erect rice                      Area
Irrigated rice                  PT
A     4     0     100     250    450
Pumping water                   TR
A     1     -1200
END

Lowland rice                    Area
Floating rice                   PT
A     4     0     75      150     250
Pumping water                   TR
A     1     -2000
END

Project                         Area
Erect rice                      A
A     1     1
Lowland rice                    A
```

```
A     1     1
Maize (project)                 PT
A     5     0     20     60     120     150
Seasonal credit                A
A     6     0.60     0.60     0.50     0.40     0.30     0.0
Rice without project            A
A     1     -200
Extension services              A
A     1     -4
Land development                A
A     6     396.6     486.4     177.7     64.7     39.8     0
END
ENDATA
******************************************
******************************************
FORMAT FOR INPUT :
================

Headings :
  1st line : Project name (max. 50 char.)
  2nd line : Data set name (max. 30 char.)
  3rd line : Currency name (max. 10 char.)
  4th line : Base year (0 or 1) and project LIFE (max. 50 years)
  5th line : Opportunity cost of capital (in percentage)

Data :
set TAB to 30
FOR ALL TYPES :
 1st line : NAME (up to 25 char.) - TAB - UNIT (up to 10 char.)

COMMODITIES : 2nd line : Time series of PRICES

ACTIVITIES : For each commodity in the activity : 2 lines
            1st line : NAME - TAB - TYPE
            2nd line : Time series of COEFFICIENTS
             END on a line at the end of the list of a given commodity
  Possible types : C (for consumed), P (for produced)

INVESTMENTS : 2rd line : Time series of unit prices
              3nd line : 5 numbers : INV.LIFE, TIME LAG FOR MAINTENANCE,
                % for MAINTENANCE COSTS, % of CONTINGENCIES, % for RESIDUAL
                VALUE

AGGREGATES : 2 nd line : VALUES or QUANT
             1 line for each item in the aggregate :  NAME - TAB - WEIGHT
               Item can be commodity, investment, aggregate or loan.
               Number of items is limited to 30 per aggregate.
             END on a line at the end of the list of a given aggregate

TRANSFERS : 2st line : TYPE
            3nd line : NAME of SOURCE - TAB - NAME of TRANSFER COMM.
  Possible types : DEFBACK, DEFWITHOUT, SURPBACK, SURPWITHOUT
  If type is DEFWITHOUT or SURPWITHOUT, source can be a commodity or an
    investment.
  If type is DEFBACK or SURPBACK, source MUST be a commodity.

LOANS  : 2nd line : 3 numbers : GRACE PERIOD, REPAYMENT PERIOD, INTEREST
           RATE%
         3rd line : PAID/NOTPAID - TAB - EQUAL.INST/CONST.CAPIT
```

4th line : Name of item with LOAN AMOUNTS - TAB - SINGLE/SEPERATE
Item can be commodity, investment, aggregate of commodities
and/or investments.

PLANS : For each item in the plan : 2 lines
1st line : NAME - TAB - TYPE
2nd line : Time series of COEFFICIENTS
END on a line at the end of the list for a given plan
Possible types : A (for annual calc),
PT (for phasing calc - nbr of units is total)
PI (for phasing calc - nbr of units is incremental)
TR if item is a transfer
PT and PI can be used only for activities and plans.
Possible items : anything but an aggregate.

TIME SERIES
===========
On the same line :
- Type : A(ctual), L(inear interpolation), C(ompound rate)
- Numbers, depending on type :
A : 1st number : year when values get constant - next : list of values
Number of values MUST be equal to - 1st number if BASE=1
- 1st number + 1 if BASE=0
L, C : 1st number : Value in base year (0 or 1)
2nd and 3rd numbers : First and last year of change
4th number : if L : Value in last years
if C : Compound rate in % (negative if decrease)
To separate numbers, use only SPACEs (i.e. blank characters)
(***)
****

The table below includes some of the results of the calculation of the plan "project". It has been prepared by transferring these results to the MANIP programme. Note that MANIP has added suffixes to the data names, to allow for precise identification (e.g. RPCST corresponds to Replacement Cost - see Annex 5, p. 16).

The second table presents rate of return and sensitivity analysis based on the aggregate PERIMETRE.

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

RESULTATS DU PLAN PROJET

9- 1-1984

Year	0	1	2	3	4	5	6	7	8-19	20
Riz-COPQ	0.0	-100.0	175.0	637.5	1337.5	1650.0	1875.0	1975.0	1975.0	1975.0
Riz-COPV	0.0	-18000000.0	31500000.0	114750000.1	240749999.9	297000000.1	337500000.0	355500000.0	355500000.0	355500000.0
Mais-COPQ	0.0	0.0	20.0	70.0	160.0	240.0	285.0	300.0	300.0	300.0
Mais-COPV	0.0	0.0	4600000.0	16100000.0	36800000.0	55200000.0	65550000.0	69000000.0	69000000.0	69000000.0
Intrants-COCQ	0.0	-2280.0	4620.0	15120.0	30570.0	36170.0	39820.0	41370.0	41370.0	41370.0
Intrants-COCV	0.0	-2280000.0	4620000.0	15120000.0	30570000.0	36170000.0	39820000.0	41370000.0	41370000.0	41370000.0
Travail-COCQ	0.0	-10800.0	10710.0	40480.0	80960.0	84500.0	84500.0	84500.0	84500.0	84500.0
Travail-COCV	0.0	-7560000.0	7497000.0	28336000.0	56672000.0	59150000.0	59150000.0	59150000.0	59150000.0	59150000.0
Pompage-COTQ	0.0	0.0	-390000.0	-1035000.0	-2015000.0	-2460000.0	-2770000.0	-2890000.0	-2890000.0	-2890000.0
Pompage-COTV	0.0	0.0	-1950000.0	-5175000.0	-10075000.0	-12300000.0	-13850000.0	-14450000.0	-14450000.0	-14450000.0
Encadrement-COTQ	0.0	-4.0	-4.0	-4.0	-4.0	4.0	-4.0	-4.0	-4.0	-4.0
Encadrement-COTV	0.0	-4000000.0	-4000000.0	-4000000.0	-4000000.0	-4000000.0	-4000000.0	-4000000.0	-4000000.0	-4000000.0
Amenagement-QUANT	0.0	396.6	486.4	177.7	64.7	39.8	0.0	0.0	0.0	0.0
Amenagement-VALUE	0.0	396600000.0	486399999.8	177699999.9	64700000.0	39800000.0	0.0	0.0	0.0	0.0
Amenagement-RPCST	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Amenagement-CONTG	0.0	39660000.0	48640000.0	17770000.0	6470000.0	3980000.0	0.0	0.0	0.0	0.0
Amenagement-MAINT	0.0	0.0	0.0	0.0	19830000.0	44150000.0	53035000.0	56270000.0	58260000.0	58260000.0
Amenagement-RSVAL	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	723022001.4
Cereales-AGGRQ	0.0	-100.0	195.0	707.5	1497.5	1890.0	2160.0	2275.0	2275.0	2275.0
Cereales-AGGRV	0.0	-18000000.0	36100000.0	130850000.0	277550000.0	352200000.0	403050000.0	424500000.2	424500000.2	424500000.2
PERIMETRE-AGGRV	0.0	-455980000.0	-509509999.8	-88914999.9	141905000.1	211800000.3	292345000.3	308410000.3	306420000.3	1029442001.5
Credit de campagne-VALUE	0.0	0.0	2772000.0	7560000.0	12228000.0	10851000.0	0.0	0.0	0.0	0.0
Credit de campagne-DBTSV	0.0	0.0	3146220.0	8580600.0	13878780.0	12315885.0	0.0	0.0	0.0	0.0
Credit de campagne-INTCM	0.0	0.0	374220.0	1020600.0	1650780.0	1464885.0	0.0	0.0	0.0	0.0
Credit de campagne-CAPCM	0.0	0.0	2772000.0	7560000.0	12228000.0	10851000.0	0.0	0.0	0.0	0.0
Credit de campagne-OUTLN	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Projet-PLNBL	0.0	-448420000.2	-517381219.7	-118271599.9	83582220.3	151185115.4	233195000.4	249260000.4	247270000.4	970292001.5

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

RESULTS OF AGGREGATES FOR PLAN Projet (VALUES)

9- 1-1984 Year	1	2	3	4	5	6	7	8-19	20
PERIMETRE U	-455900000.0	-509509999.9	-88914999.9	141905000.1	211800000.3	292345000.3	308410000.3	306420000.3	1029442001.5

Rate of return = 19.2

Switching values for opportunity cost of capital = 12.0 %

Current AGGREGATE being analysed is PERIMETRE

		PRESENT VALUES	SWITCHING VALUES PER CENT
ITEM	Riz	1777013616.07	-33.0
ITEM	Mais	334749954.04	-175.3
ITEM	Amenagement	-1209560992.66	48.5
ITEM	Intrants	-211457067.82	277.5
ITEM	Encadrement	-29877774.30	1963.7
ITEM	Pompage	-74160651.09	791.1
WITHOUT	PROJECT BALANCE	0.00	- - -
	NET BALANCE	586707085.18	

4. The Case of Livestock - the LIVMOD Programme

The programme shown below, the LIVMOD programme, was devised by the author for the needs of the FAO Investment Centre. A programme corresponding to the specifications which were developed in Chapter IV, Section F, is also being developed in Rome (LIVMOD II).

This programme is based on a dynamic model of the herd, in which the composition of the herd, mortalities and sales are determined year by year in the light of the initial composition and the different parameters provided by the user. A second part of the programme introduces data on investment, costs, and unit value of production, calculating the net value of the project. The calculation can be made twice, for the situation without project and the situation with project, and the economic analysis made on the difference.

a. The Demographic Model

The herd is divided into 11 age classes, which are fixed: males and females of 0 to 1 year, 1 to 2 years, 2 to 3 years, males of 3-4 years, males of 4-5 years, females over 3 years, males over 5 years, breeding males. The category of breeding males is a simplified method of dealing with cases where breeders are of different types. The numbers are calculated in the model by the application of sales rates and a fixed ratio to the number of breeding females, specified by the user, which results in purchases if necessary. If the male/female ratio has been fixed at zero, this category will be ignored and will not show in the tables.

As well as the initial composition of the herd, the user should specify:

- The duration of the project (maximum 20 years).

- The maximum number of breeding females at which the herd should be stabilized, and/or the maximum size of the herd in animal units: this latter factor will only be important if it represents a limitation compared with the previous one. Animal units are calculated in a simplified way by putting weighting coefficients (given by the user) on classes 0-1 year and 1-2 years.

- The males/breeding females ratio (see above).

- The demographic parameters of the herd, year by year, which are fertility rates, mortality rates (4 rates: male calves, female calves, animals of 1-2 years, and the rest of the herd), age of first calving (which means that females of 1-2 years and 2-3 years could possibly be included in female breeders), age of surplus females on sale (when the herd is stabilized - see discussion in Chapter IV, Section F.2.d), percentage of breeding females used for milk production.

- The parameters of exploitation of the herd: culling/sale rates for each class (in percentage) and purchasing programme (in number of animals).

It is possible - at the beginning of the programme - to choose between two options:

- The situation without project is presumed stable; it can be represented by stored values in year zero of the time series.

- If the situation without project is not stable, it will be necessary to define two data sets, and to calculate two projections. In this case the programme allows for the simplification of data input by defining one of the situations in relation to the other (the data of one situation will be copied in the other, to be corrected later, using the normal module for data correction).

This first part of the programme, therefore, enables a herd model to be calculated (numbers, mortalities, sales). After initial data input (from the keyboard or from a file previously created) a menu allows for choice among the following operations:

- Correcting the data on the herd.

- Printing the data on the herd.

- Calculating the model.

- Printing the herd projections.

- Proceeding to economic analysis.

The stabilization of the herd is effected in accordance with the limitation of maximum size mentioned above.

b. <u>Economic Analysis</u>

This second part of the programme defines a number of costs and benefits, in order to calculate the profitability of a project based on a modification of herd performances through a number of interventions.

The model defines the following categories of costs/benefits:

- Investments: for each category (up to 8 can be defined), the cost must be specified year by year, and the duration for calculating the replacement.

- Fixed costs: these are a series of costs, with a name, specified year by year.

- Variable costs: for each category, the user specifies a unit cost per head (year by year), and the class or classes of animals to which these costs apply.

- Purchase and sale of animals: the purchase prices are assumed constant. The sale prices, per head, can vary in time to reflect, for example, changes in unit weight or quality.

- Other income: a number of categories can be defined by specifying - as for the variable costs - a unit value per head and the classes of animals generating this production (examples: milk, manure, traction).

The programme will compile all these data, using the results of the demographic model, to calculate the series of costs and benefits, including change in the herd value at the end of the period.

The analysis may be made by reference to year zero, or by the difference between two situations. In this case the model calculates not only the rate of return of the project, but also the present values of situations with and without project, and the difference expressed in value and in percentage.

The model can test very quickly the impact of changes in hypotheses concerning the data, whether they be, for example, prices, herd performances, or culling policy, as is shown in the example of use presented below and detailed in Annex 6. But this model is only suitable for the analysis of purely livestock-raising models (ranch or development in a region of transhumant livestock-raising, for example), and does not treat satisfactorily the case of combined cropping and livestock-raising farms, where crops provide part of the animal's feed.

c.) <u>Example of Use</u>

During a study on the livestock sector in Mali (ref. 46), the author used the LIVMOD programme to test the effects of certain proposals for development policy. The northern areas of the country, with a Sahelian climate, are farmed by transhumant livestock raisers, whereas the south is occupied by settled farmers. The so-called livestock stratification policy consists in specializing the Sahelian zone in breeding, that is to say, production of young animals, which are raised in the zones further south where fodder resources are more plentiful.

The model was used to test the effects of such a livestock policy on the level of income of a Sahelian livestock-raiser, in the following way (see Annex 6):

By using the data provided by a study made by the Institute of Rural Economics, Mali, a model of 1,000 head corresponding to a transhumant herd in the dry area was defined (composition by age classes, culling and mortality rates, animals prices and milk production value - see Annex 6, Table 1).

These data, assumed constant, served to project the composition of the herd over 20 years, and also the sales value (Annex 6, Tables 2 and 3). The size of the herd was automatically stabilized by the model at 1,100 animal units, or 25% more than the present size, this figure corresponding to an estimate of the maximum carrying capacity of the grazing area used by this herd.

The stratification policy was tested in the following way: in relation to the projection described above (called "without project"), the culling rate of males was gradually changed, so as to obtain in due course a herd in which all the males would be sold between the ages of 1 and 2 years. The effect on the structure of the herd (Annex 6, Tables 4 and 9) was an increase in the number of breeding females and the off-take rate.

The second parameter tested was the rate of change. The mechanization of the calculation made it possible to test three hypotheses rapidly (change in rates of sale of males over three, six or ten years).

In the third place, it was assumed that the specialization in a breeding herd would involve (as is the case in other parts of the world) an increase in the price of young males, which was introduced into the simulation (increase of one-third over five years).

A study of the different results of the model (see Annex 6) enabled us to see that the herd management policy proposed was insufficiently attractive for the stock-raiser, which confirmed what certain specialists had already expressed in qualitative terms. The income of the herd increases initially, when the surplus males are sold, but drops afterwards during a transition period, and is stabilized again at a considerably higher level (18% or 27% more, according to the hypothesis of prices) than the reference situation. The weight of the period of transition means that the increase in revenue, discounted over 20 years at 12%, is only 8 to 14% depending on the hypothesis, a very low figure compared with the fluctuation in the level of production.

B. Problems Involved in Expanding the Use of Programmes

1. Compatibility and Transferability

Apart from the use of programmes by the organizations that develop them (for example FAO and the World Bank), it is obviously desirable for their use to be more widespread, in particular at the level of countries that have to prepare and then implement agricultural development activities whose study could benefit from the use of such tools.

Since there is very little standardization of computer systems, it is often difficult to transfer a programme from one piece of equipment to another. Thus the programmes using BASIC language of the HP-9845 system are not, in practice, usable on another system. The World Bank COMPASS/COSTAB/FARMOD system was developed on a Burrough computer in FORTRAN IV language. Although this language is considered very standard, the development of a version for IBM computer (VM-CMS operating system), which is underway, requires many months of work by the programmer.

The DASI programme was originally undertaken in order to have available a FORTRAN version of MADS II. Two versions exist, one for IBM (VM-CMS) and the other for the Apple II micro-computer.

The Pascal language was chosen by the author for new programme developments, because this language is also relatively standard, and far more flexible to use than FORTRAN. Highly structured programmes easy to read and change can be written with it, which should enable the various users to make the programme evolve by themselves if they want to. As well as the original version on the NORSK-DATA computer, a version of the MANIP programme has been developed for IBM main-frame computers (see references of the compiler used in Annex 1).

As well as the possibility of installing the same programmes on different machines, which could be done with a greater or lesser amount of work depending on the machines, it would also be a practical advantage if data files could be exchanged among users, in particular between a country where a team is working on the study of a project, and the headquarters of an external organization providing assistance (financing institutions, FAO). Such exchanges have not yet been made, and in practice would probably only be feasible if the same equipment was available in both places.

The use of diskettes (or magnetic tapes) is, here again, blocked because there is little standardization of equipment (even when supplied by the same company). The solution involves the use of certain "ad hoc" standards and the creation of software for converting data files from a given machine to the format chosen as a means of exchange.

2. Ease of Use - Training of Users

The degree of use of programmes will often depend on the organizing of training sessions for potential users. The more complicated the programme is, the more training will be required.

For a conceptually very simple programme, with a menus system like MANIP, a demonstration of one hour is enough for relatively simple use. The operation of more sophisticated models will need a little more practice, and will be greatly facilitated by the existence of predefined models that can be used as starting material.

A reasonable mastery of programmes such as COMPASS, on the other hand, requires three days' training, followed by regular practice with the programme.

A programme of the MADS type can be used very easily once the fundamental concepts of data organization have been properly understood; experience shows that these concepts can be confusing at first for many people, and that here again it is useful to organize training sessions, based on case studies.

There is always a tendency, in the development of such tools, to underestimate the importance of the time required to prepare good handbooks and teaching material (case studies) for training users. The problem is complicated by linguistic factors, when it is a question of developing French or Spanish versions, for example, of software and documentation originally developed in English.

3. Use in Rome and in Various Countries

The diffusion of the programmes described in this work is still in its very early stages.

The DASI and MANIP programmes were installed in January 1983 in Chile, and were used for the first time - to the author's knowledge - in the preparation of a large-scale project to be financed by the World Bank. The MANIP programme and its "help" manual were translated into Spanish (in Chile) and also installed in Colombia.

The DASI programme (IBM or Apple II versions) was sent, upon request, to several countries (Algeria, Tanzania, Panama, Mexico and Argentina) but information on their use is not available.

The PDMP and MADS II programmes are used by CNEA (Centre national d'études agricoles, Tunis) which has the same Hewlett-Packard equipment as that of the FAO Investment Centre.

The COMPASS system, whose COSTAB module is now used for most of the agricultural projects evaluated by the World Bank, has been installed recently at FAO in Rome. Wider diffusion will only be possible, in view of the size of the system, after an extended period of use on a limited number of sites.

The programmes available in Rome (MANIP in particular) are now used in the preparation of a high percentage of projects. There is, however, a shortage of trained analysts, and not enough terminals available. All too often, therefore, the programmes are only used for part of the calculations (calculations of rate of return and sensitivity, phasing calculations); because of this, all the potential benefits have not yet materialized.

VI. CONCLUSIONS. ACTUAL USE OF TOOLS, BENEFITS, LIMITATIONS AND FUTURE POSSIBILITIES

A. Benefits

Experience has shown that the benefits to be expected from the use of computing tools, as presented in Chapter III, are in practice materializing. This is illustrated by the few examples below, drawn from the author's experience in the FAO Investment Centre.

1. Refining Techniques

Two examples, already mentioned, illustrate the role that could be played by a calculation programme as "vehicle" of a certain technology:

- The use of switching values as a special tool for analysing the sensitivity of the profitability of a model is widespread in the World Bank, thanks initially to the CBDISPLAY programme, and then to its successor COMPASS. In the Investment Centre in Rome, the same role is played by the MANIP programme. The use of DASI, then MANIP, in Chile to prepare the project on which the author worked had the same result.

- The problems of taking inflation into account in cash-flow projections, of the treatment to be applied to the farm capital, the family work, or household consumption in farm models, are treated in different ways according to circumstances, that is to say, according to the type of farm and the economic context. Such a diversity of treatment in fact corresponds to underlying and non-explicit hypotheses (for example, excluding the family work means that a nil economic cost is attributed to it). To clarify the discussion, and suggest a more homogeneous presentation of the models, a standard example was prepared using the MANIP programme (see V.A.2.b and Annex 4), and also instructions for using the corresponding data files to define new models: the calculation programme thus becomes a simple and effective means of spreading a certain method of work (since the predefined sequence means a saving of work for the MANIP user, it will be in his interest to use it, even if he must adjust it to his specific case).

2. Saving Time for the Project Analyst

The usefulness of freeing the analyst from the physical burden of calculation lies mainly in the fact that the time thus saved will usually allow him to improve the quality of his work, because he will have more time to devote to problems raised by the project itself.

In some cases, when, for example, the time allocated for the preparation of a project is particularly short, the availability of a calculating tool can make all the difference and make it possible to achieve results that would not otherwise have been possible.

In 1982 a team from the Investment Centre prepared an irrigation project in Morocco (ref. 49) which included the study of seven areas (practically seven "projects" in one!), needing the development of three to four farm models for each area. The number of production activities in each model (crops and livestock-raising) varied from 15 to 20, given the very diversified nature of the agriculture concerned. It was possible to finish the report in less than three months, which was considered indispensable in the light of the various constraints of the calendar; it is clear that the schedule could not have been met, and the quality of work would have suffered, if a programme like MANIP had not been available.

3. Testing Technical Assumptions

When calculations are no longer a constraint, it becomes possible to test various technical assumptions, that is to say, to see the effect of such and such a change in the values of a parameter on the results of a model.

Livestock-raising is a particularly good illustration, since in the case of a herd demographic model it is particularly difficult to form an opinion on the results without making all necessary calculations, and these calculations are particularly burdensome. In such conditions, there are some types of analyses which will only be made if a computer model is available. For example, the author had the opportunity to participate, in 1979, in a study of the livestock sector in Mali (ref.46). One of the elements of the development strategy proposed was the "stratification" of cattle-raising, that is to say, the specialization of extensive pasture lands in the Sahelian area of the country for the maintenance of breeding herds for the production of calves, and the raising and fattening of animals born in the north in the more southern areas, where fodder is more plentiful. The simulation of a Sahelian herd, thanks to the LIVMOD programme, showed that the impact of such specialization on the income from the herd was such that the stock-raiser would certainly need other incentives in order to convince him. The income in fact after increasing in the initial period of transformation of the herd management (sales of surplus males), dropped to a level below that of the starting situation, before increasing again; even if the balance over a long period is finally positive, the scale of fluctuations would be a major problem (see Chapter V, A.4.c and Annex 6 for more details).

4. Validating Data and Exploring Alternatives

The role of consistency of models in facilitating data validation by checking the credibility of the obtained results has already been mentioned several times.

We have already quoted the case of a forestry project in Nepal, where the model used (thanks to informatics) proved not only the validity of the assumptions (by testing the sensitivity of the model) but also showed certain unforeeen effects of the project (indirect long-term effect of proposed forestry management on the natural forest in the area).

The case of livestock-raising projects was also particularly illustrative. The following example is taken from a report on the preparation of a livestock development project in eastern Senegal (ref. 50) written by the author. A herd model was prepared for each of the two areas of the project, based on a wealth of data provided by the monitoring component of an ongoing project in the area (a second phase had to be prepared). In this part of the world, attention is usually concentrated on meat production (because it is marketed and provides food for the big towns). The monitoring cell of the project had also concentrated its observations - apart from those on the structure of the herd, fertility and mortalities - on the growth and weight of the animals. The quantitative model prepared with the LIVMOD programme clearly showed that milk production (consumed in the home) had just as much weight in total income from the herd as the sale of animals. This was confirmed by discussions with the livestock-raisers, which emphasized the importance of milk in the utilization of the herd. The availability of a quantitative model also gave far more weight to recommendations for reorienting the project (in particular as to the parameters to be taken into account by monitoring activities!).

The use of a computing tool also, by facilitating calculation, makes exploration of alternatives easier. At the time of a study of extension plans for a given project, the formulation of many farm budgets will improve the definition of the kind of activities to be encouraged. It will also be possible to quantify all the implications of such suggestions, which sometimes leads to adjustments: a suggestion which appeared possible a priori will be judged, for example, too optimistic when all the implications in staff or supplies/logistic and/or maketing have been made explicit.

In the case of a development project in Mali based on the creation of village associations (ref. 51 - project prepared using the MADS II programme), the team responsible for the study had to define types of farm according to their technological level (traditional, improved traditional, and larger-scale improved), and to define two possible developments for each type. The phasing of farm entry into the project was at two levels: gradual entry of the families of a given village into the programme, and participation of an increasing number of villages in the project. The possibility of mechanizing calculation enabled the team to refine the definition of farm types and to test several development rates. At the level of economic analysis, where problems arose as to reference prices to be used, it was also easy to test several calculation assumptions of economic prices, and to present them in the form of a sensitivity analysis.

B. Difficulties and Limitations

The advantages of using the computing tools described should not lead us to forget that these tools also have their limitations and can themselves create specific difficulties, as shown below.

1. Risks of Error

It is easy when entering the data to make a mistake and enter erroneous values. The user may not verify his data with the necessary care, and errors can therefore easily slip into the models. The fact that a programme is powerful, and that the intermediate stages of calculation are not visible, will often make the verification of the result validity very difficult.

The need to print and check carefully all data used in defining a model cannot be over-emphasized.

2. Excessive Variety of Assumptions

When the burden of calculation is no longer a major problem, some analysts who are not very sure of their basic data have difficulty in making up their minds, and lose much time in continuously adjusting their data, or making sensitivity analyses with an excessive number of variants (here we come to the problem mentioned in the next paragraph).

Nothing is easier than to make a programme produce pages and pages of tables! In the end, the analysis is obscured rather than clarified.

3. Excessively Detailed Analysis

Because the constraint represented by the burden of calculations - when done manually - is no longer a factor, project analysts are launching out into the construction of more detailed models than they used to prepare, multiplying at will the number of crops concerned, the categories of inputs, the types of farm.

At best, an excess of detail will represent an additional burden in the presentation of the project. This will not help the process of decision-making, which was in principle the purpose for which the report was prepared.

At worst, if the data used is not founded on a solid statistical base, the result will be deceptive and will give the impression of a profound knowledge of a situation, without it being possible to form any opinion on the real degree of precision of the analysis presented.

4. Psychological Aspects

The use of data processing to prepare projects of the type that concerns us here is still relatively limited. The computer remains for many people a mysterious, almost magical instrument. There is a danger that the results will be considered with less critical judgement because they "come out of a computer", even if this reaction is often unconscious. There is a tendency to think that "the computer does not make mistakes", and too often it is forgotten that the computer is used by people who can themselves make mistakes, specify incorrect data, make errors in defining a model, or that the programme itself may be insufficiently tested and may in certain cases give erroneous results.

The best remedy is often a little common sense in looking at the results.

C. Future Prospects

1. Increased Availability of Material

The almost general availability of large computers in the capitals of developing countries, and the regular fall in costs of micro-computers, accompanied by an increase in their power, means that the introduction of the programmes described in Chapter V in developing countries becomes more and more possible. A micro-computer with a 16-bit processor and a hard disc of 10 Mo, for example, costs ten to fifteen thousand dollars on the American market (including a printer and basic software), and could easily run the MANIP or MADS III programmes. (1) The World Bank's COMPASS system needs a

---

(1) Work is in progress at FAO Investment Centre to adapt these programmes to an IBM-PC/XT micro computer.

larger capacity in the system (virtual memory of 8 Mo), but in a very few years there will be on the market micro-computers equipped with a virtual memory management system at a very modest cost.

The availability of equipment is therefore less and less of a constraint. The problem remains, first, the final development of software and the creation of versions adapted to different machines, and also versions in Spanish or French (both for messages sent by the programmes and for the documentation), and secondly, the organization of the installation of software in different countries, which must be accompanied by a minimum of training for all users.

The installation of such tools at the level of teams planning and preparing projects has two important consequences in developing countries:

- The programme of calculation, as seen above, is the vehicle of a certain methodology of analysis, and also contributes to a certain standardization in the presentation of the results.

- The availability of the calculating tool increases the possibility of participation of these countries in preparing projects.

A recent experience of the author clearly illustrates this phenomenon: a project for agricultural credit and other services was being prepared in Chile, to be financed with the participation of the World Bank. The work was being done by the Planning Office of the Ministry of Agriculture (ODEPA), with the assistance of FAO/World Bank Cooperative Programme missions. It had been envisaged that the writing of the preparation report would be finalized in Rome, in particular because of the time required. Making the DASI and MANIP programmes available to ODEPA achieved a substantial time-saving in the preparation of the fifteen farm models that were required by the project, also facilitating adjustments of hypotheses in the course of the work. This was a major element in the decision to finalize the preparation report in Chile itself, which was obviously far more desirable and ensured far greater involvement of the Government in the project.

2. From Preparation to Execution

When data processing is introduced in the preparation of projects, it quickly becomes obvious that it also has a role to play in the implementation of these same projects.

The most conventional applications concern management and monitoring of projects, and projects are increasingly envisaging - when this has not already happened - the mechanization of these tasks.

The use of simulation models of the type described in this work is also developing in the context of agricultural extension programmes. In many developed countries, models of farm budgetary simulation are being used more and more as tools of advanced management, particularly in the case of combined crop/cattle-raising farms (one may mention, for example, the programme EXPLORE of the Institute of Rural Management and Economics in France, or the package of medium-term management programmes for dairy farms developed at INRA-Grignon by J.M. Attonaty's team). Before reaching this stage, however, the use of software of the MANIP or MADS type could be introduced in many agricultural extension programmes, allowing for the regular updating - as prices change or technology evolves - of farm models serving as a basis for extension activities.

Such a proposal was introduced in the formulation of the project on which the author has been working in Chile, mentioned above. The project consists in an agricultural extension programme intended to assist medium-sized or small farms, which would create needs for additional financing by way of credit lines (which represent the most important cost of the project). This programme - and the other activities of the Ministry of Agriculture - should be supported by the establishment of an information system called "agricultural information system", through the creation of data processing centres at national and regional level by the Ministry to manage the equipment which will support the establishment of a number of information systems (agricultural statistics, information on prices and markets, yearbook of suppliers of inputs and services, agricultural meteorology, etc.), and also the mechanization of administrative tasks (in particular, the keeping of current accounts for the credit programme). It is also envisaged, in the context of the extension programme, to introduce farm registers with a certain number of farmers. The integration of the data from these registers, with all the information to be supplied by the system, will need an additional stage of analysis. This stage has been defined in the form of a sub-system called "agro-economic analysis", which will in particular integrate information to present it in the form of budgets for crops and farm models of a type that could be developed for the preparation of projects - with the computing tools presented in this work. The model, which could be adjusted regularly on the basis of new information, thus becomes a component of the extension programmes, and a synthetic means of presenting the information which is the subject of the extension.

3. Before Preparation

Part of the data necessary for the preparation of projects comes either from monitoring of ongoing projects, or from surveys made specially for the purpose.

The use of informatics in these two fields is still in its early stages in developing countries. Mechanization of processing monitor data has already been mentioned above. The introduction of computer systems at the level of processing surveys is also capable of considerable development. The standard situation at present is one in which processing of survey data is centralized at a computer centre. Data input is usually very cumbersome, and validation tests tend to eliminate a high percentage of questionnaires. When the analysis is finally presented to the user of the study, it becomes very difficult to use the data again for a subsidiary analysis - for example, because the consultant who did the work is the only person capable of understanding how the data is organized, and he has gone off to do another survey somewhere else. This kind of situation, now frequent, will disappear when the use of the micro-computer (and this is already happening) will make it possible to enter and validate data during the survey itself, to obtain very rapidly results and to give back to the user not only a report of the analysis, but also a data base easily usable for later analyses. This requires the availability of suitable and easy-to-use computer software. Such software packages already exist or are in the course of development (like the FARMAP (1) system of FAO), but their use needs to be more widely spread.

---

(1) FARMAP - Farm Analysis Package - User's Manual by J. Dixon - FAO - January 1983.

SPECIFICATIONS OF COMPUTER SYSTEMS USED

1. Hewlett-Packard Desktop Computer

- type: 9845 B
- core memory (available to user): 192 K
- language: extended BASIC (includes procedures with local variables)
- disk system: two 8 inches floppy drives (2 x 250 K) and one hard disk drive (10 Mb fixed + 10 Mb removable).

2. NORSK-DATA Mini-Computer

- type: NORD-100
- memory: virtual memory system; each task may use up to 128 Kb for programme plus 128 Kb for data
- language: Pascal, with the following extensions over N. Wirth original definition:
    - CONNECT and DISCONNECT procedures, allowing to dynamically link a file variable with an external file or device
    - GETRAND and PUTRAND procedures, allowing random disk file access.

3. IBM Computers

- type: 4341 or 3031
- operating system: VM-CMS
- compiler used: Pascal P4, version 1.17 for VM/CMS
  by Department of Computing and Control
  Imperial College
  London, England

  The following extensions over the N. Wirth standard have been used:

    - CPCOMMAND and CMSCOMMAND to access the operating system from a programme
    - CONNECT procedure (similar to NORSK-Pascal)
    - PUTRECORD and GETRECORD procedures, allowing random disk file access.

ANNEX 2

MANIP

PROJECT DATA MANIPULATION PACKAGE

USER'S GUIDE

(February 1985)

Contents:

Appendices:

INTRODUCTORY COURSE - MANIP

Manipulation Package

I. General Description

1. The Manip programme allows the user to perform the following functions:

i) to enter data into memory;

ii) to perform various operations on it; to subsequently

iii) edit, and

iv) print the stored data in tabular form, and

v) to exit from the programme.

2. The programme is organized into five modules. Each module consists of the many options a user has in performing the above-mentioned functions. Each module is described briefly below; a full description of the data processing module is given in Appendix 1.

II. General Organization and Procedure

A. Start Module

3. The Start Module is used once - and once only - at the beginning of each session. This module either creates a new data set, by specifying the name of the project, the name of the type of data to be processed, the currency, the life of the project, and possibly a number of variables - or provides access to a data set created at a previous session. A data set uses six files, whose names are formed automatically by the programme, based on a file name (from 1 to 14 characters) given by the user.

4. A data set consists of the following components:

Project Name	= e.g.	India Reservoir Development Project or Senegal Artisanal Fisheries Project etc.
Data Set Name	= e.g.	Cash Flow Projection, or Phased Project Cost, or Economic Analysis.
Currency Name	= e.g.	US$, 000 US$ etc.
Project Life	=	"Project Life" as used here is indicative of the period for which computations are intended e.g. 7 years for project cost, 20 years for economic analysis, 10 years for financial analysis etc.

Variable	=	Variables are defined by their name (Yield 1, Investment cost, Paddy, etc.) and values given to a time series related to that variable. These values can be input manually by using one of the features for defining variables or be the result of computations defined by the user. The number of years of the times series of each variable corresponds to the project period defined above.

Alternative Methods for Inputting Variable Values

Code = 1	Actual Values
Code = 2	Linear Interpolation
Code = 3	Percentages
Code = 4	Compound Rate
Code = 5	S - Curve
Code = 6	By reference

5. Actual values. The user enters on a year by year basis the actual values of the variable.

6. Linear Interpolation. Calculates variable values on a linear basis: the user must input the value for Year 0, the first and last years of the period during which the values change and the final value.

7. Percentages. The user allocates percentages of a base value to particular years.

8. Compound Rate. Applies a compound rate to the inital value, over a given interval of years. However, only one compound rate can be applied.

9. S-Curve. Calculates variable values for a Sigmoid curve. The user must input values for the first and last year of the period concerned and then select one of three forms of S-Curve.

10. By Reference. The user must input values for a base year and enter the number of the "reference" variable. Variable values are then calculated with the same percentage increases or decreases as the "reference" variable.

11. Once the initial module has been executed, the principal menu of the programme is displayed, allowing the user to select and execute any one of the other four modules, in any order as often as necessary, until selection (in the same principal menu) of the "end of job" option is made. If the user is unsure about how the functions of MANIP work, he can refer to the User's Guide (Code 5, Principal Menu).

Principal Menu

Code =	1	Data Processing
Code =	2	Data Editing (correcting headings or variables)
Code =	3	Data Listing (producing tables of variables)
Code =	4	Print/Display a List of Variable Names
Code =	5	Display MANIP User's Guide
Code =	6	Exit MANIP

B. Calculation Module - Data Processing

12. This module processes the data contained in the variables, through any of the arithmetical operations available. New variables can also be created as a result of the data processing operation. Processing of variables can be achieved by either manual processing or programmed processing. The difference is that in the case of manual processing all programming steps would have to be redefined if the computation had to be repeated, while in programmed processing, computing steps are defined once and can be run again, or edited by the user.

13. After selection of the manual or programmed mode, the user is provided with a set of 17 arithmetical and 6 support functions, which can be combined for the effective processing and manipulation of his data set:

Menu of Functions of Processing Menu

1. Display/print X
2. Rate of return on X
3. Incremental values over year zero
4. Cumulative values of X
5. Add X over A years
6. Delay X by A years
7. Advance X by A years
8. Interest on overdraft
9. AX + B
10. AXY + B
11. AX/Y + B
12. AX + BY + C
13. If X>0, AX + BY + C
14. If X<0, AX + BY + C
15. If X and Y>=0, AX + BY + C
16. Carry over if AX + BY + C<0
17. Depreciation on X
18. Debt service on X
19. Phasing a block of variables
20. Balance and switching values
21. Read from/write to another another data set
22. Run another sequence of operations
23. Print a table using a predefined format
24. End of processing
25. Print/display a list of variable names
26. Data editing (correcting headings or variables)

14. In the above menu, X and Y correspond to code numbers of variables, and A, B and C to parameters. When a function is selected, the programme calls the necessary data. It also requests the variable where the results should be stored. If the number specified corresponds to a variable not yet defined, the user must enter the name of the new variable.

15. These functions are described in detail in Appendix 1.

16. The programmed mode is achieved by defining a sequence. A sequence may include up to 50 operations. An operation can also consist of calling another predefined sequence. It is possible to structure operations in logical blocks, and to execute many of them automatically (whatever the

number of sequences, there is a maximum of 1,000 operations which can be pre-defined in a data set). Once a sequence has been defined, it can be edited, listed or run.

Menu of Data Processing Module

Code	=	1	Manual operation
Code	=	2	Define a sequence of operations
Code	=	3	Print/display the definition of a sequence
Code	=	4	Edit a pre-defined sequence of operations
Code	=	5	Run a pre-defined sequence of operations
Code	=	6	Print/list names of pre-defined sequences
Code	=	7	End of processing.

Once the data processing is completed, the user can leave the calculation mode by using Code 7 - End of processing. He will then return to the principal menu.

C. Editing

17. This module allows the user to edit the title of the data set, the name, or the values of a variable. It also allows one to add variables or to delete variables, and to change the life of the project.

Menu of Data Editing Module

Code	1	=	Headings
Code	2	=	Variable Name
Code	3	=	Variable Values
Code	4	=	Delete a Variable
Code	5	=	Add New Variables
Code	6	=	Change project life
Code	7	=	No more Editing

18. When one variable is deleted, the others are not re-numbered, thus the existing definitions of tables and sequences of operations remain valid.

19. Editing of variable values (Code 3) and adding of new variables (Code 5) lead the user to a new set of options, which allows him to manipulate or define the data of the variable directly or to use arithmetical functions for this purpose. (For details see II A, Start Module).

20. After the editing procedure is completed, the user can leave the editing module (Code 7), and return to the Principal Menu.

D. Data Listing - Printing Module

21. The user has the possibility of preparing tables on the basis of the data contained in his data set. These tables can be either a complete or partial listing of the variables in the order of their occurrence in the data set or a selective listing of the variables in an order chosen by the user. Four types of printers can be used:

- the terminal of the user (CRT);
- the line printer (132 characters per line);
- the H-P printer (227 characters per line);
- the Output file (internal word processor); 1/

1/ The "output file" or "text" file is a work file created by the programme having the same name as the data set e.g. (DATA SET NAME)-W:TEXT. If during a particular run of the programme the user specifies the output file as the printer, the output file accumulates the requested information. At the beginning of each run of the programme the output file is cleared. If the user would like to retain the information printed to the output file, he has the following options:

To obtain a hard copy of information written to output file the user can:

(1) EXIT from programme
@ COPY-FILE
DESTINATION FILE: H-P (name of a printer)
SOURCE FILE: name of Data file

If the user would like to retain the contents of the output file in a permanent text file, he can write the contents of the output file to another text file:

(1) EXIT from programme
@ PED (name of Data file)
Once the output file appears press W (write)
STORE DOCUMENT: "name of new file in inverted commas"
CR

(2) or COPY-FILE to another text file:
Exit from programme
@ COPY-FILE
DESTINATION FILE "new file name in inverted commas"
SOURCE FILE: name of Data file

In this way the file will be stored as a text file for future editing.

22. The Data Listing Mode provides the following options:

Data Listing Menu

Code	=	1	Standard output of all/block of data
Code	=	2	Produce a table using a previously defined format
Code	=	3	Define a table format
Code	=	4	Print/display a previously defined format
Code	=	5	Edit a previously defined format
Code	=	6	Print list/names of defined formats
Code	=	7	End of output operations

23. The standard output (Code 1) prints all or a selected block of variables and is normally used to verify the content of variables, mainly during the data processing phase.

24. If headings and names of variables are adequate, the selective listing could be used to prepare final tables for reports and annexes. Lists prepared by the MANIP package have all variable names printed in capital letters, and standard headings. However, if the MANIP output is not adequate these lists could be edited, i.e. footnotes added, headings improved, variable names redefined, if the lists were printed into the output file from where they could be converted into a text file and edited with normal word-processing procedures (see footnote 1/).

25. The creation of tables (selective listings) - Code 3 provides the following possibilities for defining the tables:

- table sub-title;

- table number;

- number of years, or years to be printed;

- scaling of data - use of constant factors (thousand, million etc.) to define the size of values to be printed.

- totals per line (values added over the number of years to be printed);

- number of decimals;

- blank lines in a table;

- printer to be used.

26. Once a table has been created, it can be edited and any of the above parameters modified. After the tables have been created and the editing process is completed, the user can either return to the Principal Menu without printing the tables, by using Code 7, or print the tables by using Code 2. In the second case, the tables will be printed on the printer defined in the table format.

27. The user has the option of printing each table individually by using Mode 2, or printing a set of pre-defined tables in a sequence. For this he would have to use function 23 in a new or pre-defined sequence in the processing module and then run the sequence.

III. Use of Pre-defined Sequences or Formats

28. The user has the option of creating new sequences or table formats or of copying existing sequences or formats.

29. In the case of new sequences or table formats, the user needs to define all the functions and parameters he deems relevant for his sequence or table format. This process is time consuming, as he must frequently check the accuracy of computations and the adequacy of the presentation of a table format.

30. Once a sequence or a table format has been defined, and the user wishes to apply the same operations and table formats to other data in the same or another data set 1/ he can copy the sequence or table format, thus creating a new sequence or table format. When such a sequence or table format is copied, all the code numbers of the variables in that sequence/table format can be modified by a postive or negative constant factor, (displacement factor). This allows the use of the same sequence/table format for another block of variables, the order of which must however, correspond to the order of the variable set.

31. This mechanism considerably reduces the work required in the detailed specification of operations needed for the calculations of a project. It can be used in two ways:

i) It is possible to constitute a library of "standard files" corresponding to typical applications; for any given application, a set of files could contain variables, with nil values, and definitions of sequences and tables.

ii) When a project composed of several farm models is under study, the types of analysis and presentation desired will, in general, be the same for all the models. The variables, the sequences of operations and the tables corresponding to one model can be defined and copied as often as necessary to construct the other models.

IV. Example

Construction of a Model

32. The following example shows how MANIP can be used to construct a model.

33. Let us suppose that the following information is available for a farm model (in the form of "variables", that is to say, series of annual data):

1/ Only possible if in the same user space.

1 = Total value of production
2 = Other income
3 = Investments
4 = Cost of production less labour
5 = Cost of wage-earning labour force
6 = Value of family labour force
7 = Household consumption

34. The intention is to study the methods of financing this farm by long-term credit to finance the investments, and short-term credit to finance production costs (less labour force and wage-earning labour force).

35. The lines of a summary table may be constructed step by step in the following way:

- Step 1: Add the variables 1 and 2 (operation 12 in the menu, with X=1, Y=2, A=1, B=1, C=0) to create the variable 8 = total income.

- Step 2: Using operation 12 or operation 20 (balance), add the variables 4 (costs of production) and 5 (wage-earning labour force), applying to each of them a weighting coefficient corresponding to the calculation of interest on a short-term loan: a rate of 15% per annum and a loan of six months will be translated by, for example, a coefficient of 0.075. Operation 20, by which more than two variables (up to 50) can be added together, is preferable here, since it will be easier to correct the model by adding other variables or changing the coefficients. Result: variable 9 = interest on seasonal credit.

- Step 3: Using operation 20, the sum of variables 3, 4, 5 and 6 can be calculated to create the variable 10, or total expenditure (before financing).

- Step 4: The difference between variables 8 (total income) and 10 (total expenditure) is calculated (operation 12, AX + BY + C, with X = 8, Y = 10, A = 1, B = -1, C = 0), to give the variable 11: net benefit before financing.

- Step 5: And so on ...

36. The complete example presented in Annex 4 (in which the variables do not have the same code numbers, since they were created in a different order) then shows how the user may choose to calculate the amount of a long-term loan as a percentage of investment costs, the difference being financed by a contribution by the farmers; how the debt service can be calculated in constant terms; how to apply to it a deflation factor (to reflect the "actual" weight of the reimbursements in a model expressed in a constant monetary unit); and how the benefit after financing is adjusted by the values of household consumption and family work to obtain a projection of monetary income.

V. Practical Use - Some Advice

A. Typical Problems and Errors in Using MANIP

37. The MANIP package permits one to create and manipulate data by using pre-defined sequences. This manipulation of data must follow established mathematical rules, otherwise the programme would be automatically interrupted or calculate incorrect results.

38. Typical mistakes that occur when using the MANIP package are:

- assignment of values to the wrong variables;
- reassignment of the results of data manipulations to the same variable which served as a basis for the computation;
- use of variables in pre-defined formats which do not correspond to those used in the computing process;
- incorrect logical order of operations within predefined sequences;
- creation of circular sequences i.e. within a sequence calling an operation that runs the same sequence;
- accidental overwriting of a given data set by other data when transferring data frome one data set to another.

39. Most of these errors will occur especially at the beginner's level but will be avoided after some experience with the MANIP package. Some of them will, however, occur even with the experienced MANIP user, and the objective should be to reduce them to save time.

40. The best way to reduce the occurrence of errors, and the subsequent time-consuming process of identifying and correcting these errors is a clear organization of data and sequences. The MANIP package allows organization of variables in blocks as shown in Section V B. Clear organization of operations and sequences depends, however, entirely on the extent to which the user has analyzed his problem and defined his objective.

41. To reduce the time spent on identifying errors, some of the most frequent causes are listed below:

1. Internal Rate of Return

The screen shows "no roots" or internal rate of return, < 0 or > 200. This might correspond to the reality, but could also be the result of some errors in your data:

Possible Causes

(a) The weights in function 20 (Balance and switching value), where the IRR is being calculated are wrong, i.e. all of the weights are positive or negative.

Solution: Check operation and correct weights.

(b) The dimension used for computing revenues, investments or operating costs are not homogeneous, e.g. revenues are expressed in single currency units while investment and operating cost are expressed in thousands or millions.

Solution:

- Change values in variable, or
- Adjust weight in function 20 to take account of different dimensions, or
- Introduce operation in sequence that would reduce or increase dimension of variable (by multiplying it with a constant factor corresponding to dimension).

(c) The build-up of yields or revenues has been started in the wrong year.

Solution: Check and correct variables

(d) One of the variables used in function 20 contains unrealistically high or low values (refer to b)

2. Variable Values

After computations, variables assume unrealistically high or low values

Possible Causes

(a) In a predefined sequence, the results of a computation are assigned to the same variable, that has been used as base variable for that particular computation

Examples:

i) Multiply a variable with a constant:

```
      X          A    B        Y
20 Revenues X 0.01 + 0 =  20 Revenues
```

ii) Multiply variables with each other:

```
      X         Y       A     B      Z
20 Yields  X 21 Area X 0.01 + 0 = 20 Yields
```

iii) Divide variables by each other:

```
      X          A        Y          B          Z
20 Revenues X 0.01 / 21 Price  + 1.00  = 20 Revenues
```

Solution: The above computations are perfectly acceptable, as long as the base variables are interim variables that are newly computed with each run of the sequence. In this case no harm can be done.

(b) In a particular sequence, a variable could be the result of computations of a previous sequence. If only this particular sequence is run, without the previous sequence which gives the variable its value, the results could be wrong.

Solution: Run the whole set of sequences and the error will be automatically corrected.

(c) Although values were assigned to a variable, these values have disappeared without clear reason:

i. By accident an operation assigns a zero value to the variable.

ii. The delay function has been applied and the result assigned to the same variable. After a series of runs, the variable value will disappear.

iii. Data are being read from another data file, but the value of that other data file is zero. Through the transfer of data procedure, the value copied from the other file will overwrite the value contained in the file being operated.

Solution: In these three cases the wrong operation has to be traced and deleted and the variable in question has to be corrected.

3. The Programme is Interrupted and the Following Message Appears

Possible Causes

(a) Arithmetic overflow in address ....

The computation contains an invalid arithmetical operation, most likely caused by the corruption of the values of a variable in the data file.

Solution:

- Exit and re-enter your MANIP file.
- List your variables on the screen or the printer until the programme breaks off again. Identify the variable at this point.
- Once the "bad" variable has been identified, edit this variable by transferring the values of any variable from a different file using Function 21 in the data processing menu.
- Identify the sequence and operation where the "bad" variable has been calculated and correct the operation if there is a mathematical error.
- If you cannot identify the operation where the "bad" variable is being created, contact Steve for advise.

(b) D2 error in address.........

This error occurs when the user is trying to create a variable, and his user space is exhausted. Once this error has occurred extreme caution is required as your data base and sequences could be destroyed. The following procedure is recommended.

Step 1: Press ESC: @ sign appears

Step 2: Type "US-ST" and give name of your programme area; i.e. 1 or A1, 2 or 2a etc. The terminal will than show whether the user space in question is indeed used up. This can be seen by the statement:

Available pages 200
Used pages 200

If these two numbers are identical there is no space left and the user should contact the person responsible for supervision of project computer space in his service, who will arrange for more space to be allocated.

Step 3: After the user space is expanded the following procedure has to be followed. The user re-enters his data set - calls the manual operation in the processing module. There he will read from another data set any given variable and "write" into the variable that has turned "bad". After this procedure he may return to the edit module and delete or correct this particular variable.

Step 4: After this procedure is completed the programme can again be operated.

Warning: The "bad" variable cannot be deleted, unless this procedure is followed.

(c) D3 error in address.........

This error occurs when the user tries to create an operation in a sequence when the user space in his programme area is exhausted. Once this error has occurred extreme caution is required as otherwise your data base and your sequences may be destroyed. Once the D3 error has occurred, it is impossible to use the same data base again. If the data base and the sequences are to be salvaged the following procedure has to be followed.

Step 1: as above

Step 2: as above

Step 3: After the user space has been expanded the following procedure has to be followed. The bad operation will be identified by calling each operation in the sequence commencing with the last. The operation at which the programme aborts is the "bad" one. After this has been done, the user creates a new set of files using a different generic name.

Step 4: Through the manual operation in the Data Processing Module, the user copies all variables from the old to the new file using identical numbers.

Step 5: All operations in all sequences of the old file are then copied except the operation identified as "bad".

Step 6: After all operations except the "bad" one have been copied, new operations can be created.

Step 7: All the table formats are copied from the old file.

Step 8: After all the variables, sequences and table formats have been copied, the old file should be deleted so as not to occupy computer space.

To avoid the problems mentioned above, check regularly that there is enough space in your user work area (User-Statistics Command).

B. Proposals for More Efficient Data Organization

Creation of Variables

42. Efficient data organization can be best brought about if related variables are arranged in blocks. Some of these variables, however, are created as a result of processing of data. While it is possible to foresee what these variables may be, using the standard procedure of entering variables at the start of the programme may be time consuming as names and dummy values must be entered. It is also possible as with the case of large projects, that all variables necessary may be difficult to determine at the start. The procedures described below show how variables may be created as quickly as possible without having to enter values and names:

Step 1: Creation of data base.

Step 2: When the programme asks for number of variables to be entered, type in 1. Give this variable the name you wish to assign to your first variable.

Step 3: Enter Data Processing Module and choose Manual Operation (code 1). From the menu choose function 9 - Z = AX + B. Give the number of your variable, assign to A and B the values 0, and have the results assigned to variable 2.

Step 4: As variable 2 was previously not created, the computer will ask for the name of the new variable. Type the name you wish to give to variable 2, or type in a "dash" if you are uncertain of the name.

Step 5: Repeat procedure described in step 3, give the number of your first variable, i.e. 1, give the number of variables, i.e. now 2, and assign the value 0 to A and B. The results should be in variables 3 and 4. Repeat the procedure in step 4. Now already 4 variables are created.

Step 6: Repeat the procedure for the creation of the next 4, 8, 16, 32 etc., variables. As in each case the block of variables covered by the operation will be larger, the number of new variables will also be larger.

Step 7: Enter the Data Editing Module and assign names and values to the variables, as required.

Advantage. Since only variable names are being asked for, the user can leave individual variables unused, and organise his variables in an efficient manner. This will facilitate at a later stage duplication of variables, identification of errors, etc.

43. Variables can also be created by transferring them from one file to another, by using function 21 in the Data Processing Module. This function can be effectively used, for the creation of identical variables in other data sets, for which only the values differ. For this purpose, the user could create a file containing the model data base, for example 50 variables. Once these 50 variables are named, the entire variable block can be transferred to file 2, by copying them into variables 1 to 50. Then, the same variables could be read back into file 1, and inserted into variables 51, 101, 151 etc. In this way, 150 to 200 variables can be created within a relatively short time. After this procedure is completed, the values for each individual variable can be entered.

44. The procedure may be followed if different data sets with identical data structure, i.e. variable names, need to be processed. It would then be possible to copy an existing sequence with a given "Displacement factor" thus saving time.

VI. Advantages and Limitations of MANIP in DDC Project Work

45. The use of MANIP in DDC Project Work is useful and cost effective in the following situations:

A. Advantages

i) Experts and technicians are not certain about assumptions. In situations where the technical experts are not certain about parameters to be used in agricultural models, the preparation of various financial and/or economic models would be very time consuming. Also, if a review of a project leads to modification of parameters, often the entire financial and economic cost, as well as project cost need to be recalculated.

Using MANIP, the project cost, financial and economic analysis can be prepared even if technical parameters are not final. Once a decision has been made, a final computation could be carried out. A further revision of these parameters during project reviews would result in a simple editing of the data base in the computer, and the up-dated analysis would be available within a short time.

ii) Assumptions need to be tested for their financial feasibility. In situations where technical or financial assumptions need to be tested, i.e. the same computations repeated many times, with different parameters, the MANIP programme gives the user the advantage of altering the parameters concerned and "rerunning" the programme. Possible applications would be to identify minimum prices and test the effects on financial or economic internal rates of returns, or the effect of different loan schedules on cashflows and financial positions of farms or enterprises.

iii) Identical models need to be calculated for different data. If the same financial or economic models need to be prepared for different but similiar situations, e.g., different technical parameters for three different models which are prepared for four project regions, the possibility of preparing one model and using it in similar data sets can result in considerable gain of time, and thus higher efficiency in project preparation.

iv) Complex and time consuming computations can be done automatically. Certain computations are relatively simple but also extremely time consuming. Predefined models or computing functions can considerably reduce the time required for these computations and thus provide more time for substantive project preparation work. Typical computations of these nature are:

- calculation of internal rate of return;
- calculation of loan programmes;
- phasing of variables;
- financial analysis and statements;
- project cost with price escalation;
- economic analysis using Standard Conversion Factors or /Export Premiums;
- sensitivity and risk analysis.

Any of these computations can be done by hand, but most are very time consuming. Consequently, the analyst might forego some analyses altogether, due to time constraints. Well organized MANIP packages can considerably increase the efficiency of analysis and thus improve project analysis.

v) Numerous and similarly designed tables need to be printed. Economic and financial analysis require the production of detailed financial cashflows and economic cost benefit analysis. Secretarial capacity is scarce, and the preparation of project reports always require the typing of complex tables. The MANIP programme permits tables to be transferred to the word processing area and edited there. For example, one to two days are usually required for the typing of a large table. Using a combination of MANIP and word processing, the same table could be completed within 2 to 3 hours.

B. Limitations

i) Key to the efficient and effective use of MANIP is the preparation of clear and well organized variables and a schedule of operations. If this is not done, the likely result would be a loss of time in tracing and correcting errors. Simple corrections which would normally require minor alterations in manual calculations may take much longer in the MANIP programme. However, if the user does not carry out the changes in the MANIP programme and decides to recalculate by hand, he has lost all the time previously invested.

ii) Formulation of calculation sequences requires time if a prepared package or solution is not available. Complex computations need to be checked step by step to ascertain that the desired calculations are executed as intended. Manual calculation of tables may be done faster than with the MANIP package, at least for the first set of tables. The advantage of MANIP is demonstrated only if the parameters for these tables are modified. Manual recalculation requires the same amount of time as for the previous table, while in the MANIP package very little time is required to edit the data base and to re-run pre-defined sequences.

iii) The use of MANIP (and other software packages) will remain unattractive if the present ratio of terminals to users is not improved and if printers are not conveniently located.

MANIP

PROJECT DATA MANIPULATION PACKAGE

USER'S GUIDE

(February 1985)

Description of Functions Available in Data Processing Menu

Description: This Appendix describes all functions available in the Data Processing Menu of the MANIP programme.

Contents: Description of Functions Available in Data Processing Menu.

MENU OF FUNCTIONS OF PROCESSING MENU

1. Display/print X
2. Rate of return on X
3. Incremental values over year zero
4. Cumulative values of X
5. Add X over A years
6. Delay X by A years
7. Advance X by A years
8. Interest on overdraft
9. AX + B
10. AXY + B
11. AX/Y + B
12. AX + BY + C
13. If X>0, AX + BY + C
14. If X<0, AX + BY + C
15. If X and Y>=0, AX + BY + C
16. Carry over if AX + BY + C<0
17. Depreciation on X
18. Debt service on X
19. Phasing a block of variables
20. Balance and switching values
21. Read from/write to another data set
22. Run another sequence of operations
23. Print a table using a predefined format
24. End of processing
25. Print/display a list of variable names
26. Data editing (correcting headings or variables)

FUNCTION = 1 Display/Print X

Table 1

Year	0	1	2	3-10
X =	0	10	20	30

Comment: Displays the values of a variable on the terminal screen or the other printers without having to use the print module. It could, for example, be used as the last operation in a calculation sequence to see the values in the variable during the course of computations. It would thus be possible to see whether computations are executed as intended, without having to refer to the print or data editing module.

FUNCTION = 2 Rate of Return on X

Table 2

Year	0-1	2	3	4	5-10
X =	0	-100	-200	100	200

Rate of return = 44.7

Comment: The calculation of the rate of return is made automatically on incremental values of the variable over year zero. The result is printed on the terminal screen or the other printers.

FUNCTION = 3 Incremental Values Over Year 0

Table 3

Year	0	1-10
X =	10	20
Z =	0	10

Comment: Self-explanatory.

FUNCTION = 4 Cumulative Values of X
(Option = Total Cumulative Values)

Table 4 a

Year	0	1	2	3	4	5	6	7	8	9	10
X =	0	10	20	30	30	30	30	30	30	30	30
Z =	0	10	30	60	90	120	150	180	210	240	270

FUNCTION = 4 Cumulative Values of X
(Option = Incremental Cumulative Values)

Table 4 b

Year	0	1	2	3	4	5	6	7	8	9	10
X =	10	20	20	20	20	20	20	20	20	20	20
Z =	0	10	20	30	40	50	60	70	80	90	100

Comment: Self-explanatory.

FUNCTION = 5 Add X Over A Years

Parameter a = 5

Table 5

Year	0	1	2	3	4	5-8	9	10
X =	0	10	20	30	30	30	30	30
Z =	0	0	0	0	90	0	150	30

Comment: Self-explanatory.

FUNCTION = 6 Delay X by A Years

Parameter a = 1

Table 6

Year	0-1	2	3	4	5-10
X =	0	10	20	30	30
Z =	0	0	10	20	30

Comment: Useful in sensitivity analysis. The value in year zero is not affected by function. The value of year 0 is copied into the year(s) by which X is delayed.

FUNCTION = 7 Advance X by A Years

Parameter a = 1

Table 7

Year	0	1	2	3	4	5-10
X =	0	0	10	20	30	40
Z =	0	10	20	30	40	40

Comment: Useful in sensitivity analysis. The value in year zero is not affected by this function. The value of the last year of X is copied into the year(s) which remain without value(s) after X is advanced.

FUNCTION = 8 Interest on overdraft
Parameter a = 13%

Table 8 a

Year	1	2	3	4	5	10
CASH-FL EXCL. PR/LOSS	100	-300	250	175	175	175
PR/LOSS EXCL. INT/OD	50	50	65	45	45	45
INTEREST ON OVERDRAFT	0	0	13	0	0	0
PR/LOSS INCL. INT/OD	50	50	52	45	45	45
CASH-FLOW BALANCE	150	-250	302	220	220	220
CUMUL. CASH-FLOW BAL.	150	-100	202	422	642	1742

Table 8 b

Year	1	2	3	4	5	10
CASH-FL EXCL. PR/LOSS	100	-150	250	175	175	175
PR/LOSS EXCL. INT/OD	50	50	65	45	45	45
INTEREST ON OVERDRAFT	0	0	0	0	0	0
PR/LOSS INCL. INT/OD	50	50	65	45	45	45
CASH-FLOW BALANCE	150	-100	315	220	220	220
CUMUL. CASH-FLOW BAL.	150	50	365	585	805	1905

Comment: Used in financial models. Cash flow deficits are assumed to be financed by short-term credit, interest on which is charged to the following year. Assuming that "cash flow balance excluding profit/loss" and "profit and loss before interest on overdraft" are stored in two variables Y and X, the function will calculate four new variables.

Z = Interest on overdraft;
Z+1 = Profit/loss exluding interest on overdraft;
Z+2 = Cash flow balance;
Z+3 = Cash flow cumulative balance.

Cash flow deficits in a given year are automatically balanced by short term credit (overdraft), and the interest is charged to the following year. Thus financial statements can be automatically balanced.

FUNCTION = 9 AX + B

Parameter a = 0.1
Parameter b = 0

Table 9

Year	0	1	2	3-10
X =	0	10	20	30
Z =	0	1	2	3

Comment: Useful for calculating cost or revenues by multiplying physical quantities (units, tons, etc.) with a constant unit price a.

FUNCTION = 10 AXY + B

Parameter a = 0.1
Parameter b = 0

Table 10

Year	0	1	2	3	4-10
X =	0	10	20	30	30
Y =	0	10	20	30	40
Z =	0	10	40	90	120

Comment: Useful for calculating values that are the product of two different time-rows, e.g. area and yields per hectare. Parameter a could represent a unit price or determine the scale of the result.

FUNCTION = 11 AX/Y + B

Parameter a = 2
Parameter b = 0

Table 11

Year	0	1	2	3	4-10
X =	0	10	20	30	30
Y =	0	10	20	30	40
Z =	0	2	2	2	2

Comment: Self-explanatory. See also Function 10.

FUNCTION = 12 AX + BY + C

Parameter a = 2
Parameter b = 2
Parameter c = 0

Table 12

Year	0	1	2	3	4	5-10
X =	0	10	20	30	30	30
Y =	0	0	10	20	30	40
Z =	0	20	60	100	120	140

Comment: Especially useful if series of corresponding pairs of variables are to be added, e.g. Crop 1 to Crop 10 in Crop Model A and Crop 1 to Crop 10 in Crop Model 3. Parameters a and b could stand for the number of hectares of each of the two crop models.

FUNCTION = 13 If X > 0, AX + BY + C

Parameter a = 1
Parameter b = 1
Parameter c = 0

Table 13

Year	0	1	2	3	4-10
X =	0	-10	20	-30	40
Y =	0	10	20	-30	40
Z =	0	0	40	0	80

Comment: Permits selection of positive values and to effect computations only with these selected values.

FUNCTION = 14 If X < 0, AX + BY + C

Parameter a = -1
Parameter b = 0
Parameter c = 0

Table 14

Year	0	1	2	3	4	5-10
X =	0	0	-100	-200	100	200
Y =	0	10	20	30	40	40
Z =	0	0	100	200	0	0

Comment: Permits selection of negative variable values and to effect computations only with these selected values: e.g. if X is the balance of labour use, the function could be used to calculate hired labour for the years with labour deficit (X is in man-days, and parameters a = -1, b = 0 and c = 0). The result in variable Z will be the hired labour in man-days.

FUNCTION = 15 If X and Y >= 0, AX + BY + C

Parameter a = 1
Parameter b = 1
Parameter c = 0

Table 15

Year	0	1	2	3	4-10
X =	0	-10	20	-30	30
Y =	0	10	20	-30	40
Z =	0	0	40	0	70

Comment: Permits selection of positive values from two variables, on condition that values in the two variables assume positive values in the same year. If the selection should be for negative values, reversal of the sign beforehand would be required (with Function 9).

FUNCTION = 16 Carry over if AX + BY + C < 0

Parameter a = 1
Parameter b = -1
Parameter c = 0

Table 16

Year	0	1	2-10
X =	10	16	20
Y =	12	12	12
Z =	0	2	8

Comment: To carry over results from one year to the next year if the sum of two variables is negative. For example, if X is consumption and Y production of a crop, the function will calculate the shortfall if the result is positive, and carry over the difference to the next year, if the result is negative.

FUNCTION = 17 Depreciation on X

Useful life = 10 years

Table 17

Year	0	1	2	3	4	5	6	7	8	9	10
ASSET VALUE	0	1,000	0	0	0	0	0	0	0	0	0
DEP.VALUES	0	100	100	100	100	100	100	100	100	100	100
CURRENT VALUES	0	900	800	700	600	500	400	300	200	100	0

Comment: Calculates straight-line depreciation on asset values stored in a variable, as well as current book value (at end of year, after depreciation) for a given life of the asset.

FUNCTION = 18 Debt Service on X

The loan function allows the choice of the following options:

(i) Annual amounts to be treated as separate or one loan
(ii) Interest rate
(iii) Grace period
(iv) Repayment period
(v) Repayment as equal instalments (inclusive of interest) or equal capital instalments

An updated form of this function is under preparation which will allow different grace periods for repayment of principal and interest. The loans are assumed to be disbursed at the beginning of the year and reimbursed at the end of the year. If the loan is disbursed over several years, the length of the grace period (if applicable) must cover the disbursement period so that repayment commences after the last loan has been disbursed.

Table 18 a

Interest rate = 10%
Grace period = 2 years - interest paid during grace period
Repayment period = 5 years - equal capital instalments

Year	0	1	2	3	4	5	6	7	8-10
LOAN	0	2000	0	0	0	0	0	0	0
TOTAL DEBT SERVICE	0	200	200	600	560	520	480	440	0
INT. COMP.	0	200	200	200	160	120	80	40	0
OUTSTANDING LOAN	0	2000	2000	1600	1200	800	400	0	0

Table 18 b

Interest rate = 10%
Grace Period = 2 years - interest paid during grace period
Repayment period = 5 years - equal instalments

Year	0	1	2	3	4	5	6	7	8-20
LOAN	0	2000	0	0	0	0	0	0	0
TOTAL DEBT SERVICE	0	200	200	528	528	528	528	528	0
INT. COMP.	0	200	200	200	167	131	92	48	0
OUTSTANDING LOAN	0	2000	2000	1672	1312	916	480	0	0

Table 18 c

Interest rate = 10%
Grace period = 2 years - interest NOT paid during grace period
Repayment period = 5 years - equal capital instalments
Annual amounts treated as separate loans

Year	0	1	2	3	4	5	6	7	8	9-10
LOAN	0	2000	1000	0	0	0	0	0	0	0
TOTAL DEBT SERVICE	0	0	0	726	1041	968	895	823	266	0
INT. COMP.	0	0	0	242	315	242	169	97	24	0
OUTSTANDING LOAN	0	2200	3520	3146	2420	1694	968	242	0	0

Table 18 d

Interest rate = 10%
Grace period = 2 years - interest NOT paid during grace period
Repayment period = 5 years - equal instalments
Annual amounts treated as separate loans

Year	0	1	2	3	4	5	6	7	8	9-10
LOAN	0	2000	1000	0	0	0	0	0	0	0
TOTAL DEBT SERVICE	0	0	0	638	958	958	958	958	319	0
INT. COMP.	0	0	0	242	323	260	190	113	29	0
OUTSTANDING LOAN	0	2200	3520	3234	2599	1902	1134	290	0	0

Comment: The calculation of the interest component and the outstanding amount (at year end, after payment of the corresponding annual instalment) is optional.

FUNCTION = 19 Phasing a Block of Variables

Table 19 a

PHASING - CONSTANT WITHOUT PROJECT SITUATION

Year	0	1	2	3	4	5	6-10
YIELD WITH PROJECT	1.0	1.5	2.0	2.5	3.0	3.0	3.0
NO. OF HA (SUPPLEMENTARY)	0.0	10.0	20.0	20.0	0.0	0.0	0.0
HA OF YEAR 1	10.0	15.0	20.0	25.0	30.0	30.0	30.0
HA OF YEAR 2	20.0	20.0	30.0	40.0	50.0	60.0	60.0
HA OF YEAR 3	20.0	20.0	20.0	30.0	40.0	50.0	60.0
RESULTS - CONSTANT W/O	50.0	55.0	70.0	95.0	120.0	140.0	150.0

Table 19 b

PHASING - NON-CONSTANT WITHOUT PROJECT SITUATION

Year	0	1	2	3	4	5	6-10
YIELD WITH PROJECT	1.0	1.5	2.0	2.5	3.0	3.0	3.0
YIELD WITHOUT PROJECT	1.0	1.1	1.2	1.2	1.2	1.2	1.2
NO. OF HA (SUPPLEMENTARY)	0.0	10.0	20.0	20.0	0.0	0.0	0.0
TOT. YEAR 1 HA	10.0	15.0	20.0	25.0	30.0	30.0	30.0
TOT. YEAR 2 HA	20.0	22.0	30.0	40.0	50.0	60.0	60.0
TOT. YEAR 3 HA	20.0	22.0	24.0	30.0	40.0	50.0	60.0
RESULTS - TOTAL VALUES	50.0	59.0	74.0	95.0	120.0	140.0	150.0
INCR-YR 1 HA	0.0	4.0	8.0	13.0	18.0	18.0	18.0
INCR-YR 2 HA	0.0	0.0	6.0	16.0	26.0	36.0	36.0
INCR-YR 3 HA	0.0	0.0	0.0	6.0	16.0	26.0	36.0
RESULTS - INCREMENTAL VAL	0.0	4.0	14.0	35.0	60.0	80.0	90.0

Comment: The function calculates aggregated values of a time series (unit variable) phased in by a certain number of units each year (phasing variable) over a certain number of years (from year 1) defined by the user.

The function has two options:

- Without project situation is constant, represented by values in year zero of the unit variable (see table 19 a).
- Without project situation is not constant, two unit variables are necessary, one for with project and one for without project situations. On option, the results of the calculation will be either total values or incremental values (over without project situation (see table 19 b).

The number of units stored in the phasing variable can be either total or incremental. If they are incremental (i.e. incremental over the previous year), any value stored in year zero will be ignored.

Note: The function calculates final results but for demonstration purposes, production corresponding to each group of hectares has been calculated.

FUNCTION = 20 Balance and Switching Values

Table 20 a

Simple Balance

Year	0	1	2	3	4	5-10
X =	10	20	20	20	20	20
X =	0	10	20	30	30	30
Y =	0	0	10	20	30	40
BALANCE	10	30	30	30	20	10

Table 20 b

Balance and Switching Values

Year	0	1	2	3	4-10
GROSS REVENUES	8000	11000	13000	16000	18000
INVESTMENT COST	0	20000	0	0	0
OPERATING COST	4000	5000	6000	6000	6000
BALANCE	4000	-14000	7000	10000	12000

Rate of return = 32.0

Switching values for opportunity cost of capital = 10.0 %

	PRESENT VALUES	SWITCHING VALUES PER CENT
PROJECT GROSS REVENUES	98603.71	-20.2
PROJECT INVESTMENT COST	-18181.82	109.4
PROJECT OPERATING COST	-35958.31	55.3
WITHOUT PROJECT BALANCE	-24578.27	80.9
NET BALANCE	19885.31	

Comment: It is possible to specify a list containing up to 50 variables in one operation. Each variable will be multiplied with a constant factor, usually 1 or -1, but other factors are possible. The factors will not modify the value stored in the variable, but be reflected only in the results. This function is useful for many computations: to calculate totals, contingencies, economic values and sensitivity analysis. The internal rate of return is calculated on the incremental values of the balance. Without project balance is the projection over the full period of the value in year 0 balance.

FUNCTION = 21 Read From/Write to Another Data Set

This function enables the user to transfer variables (names and values) from one data set to another. In the destination data set, the copied variables can be either inserted into a pre-defined position, or be appended after the last defined variable. This function is useful, for instance in a project with several project areas, each one having several components and requiring individual economic/financial analysis. A data set could be created for each project area and the results of computation from each of the files could be transferred to a central file for calculation of aggregate economic analysis for the entire project.

This function would be useful in the following situations:

a) Results of different computations, (project cost, crop projections, farm models) which would be stored in different data sets, and merged for final economic analysis.

b) One data set with a set of pre-defined variables and sequences would be used to process sets of identical variables with different values stored in different data sets. Useful for calculating farm models with different variable values, thus saving time and computer space.

c) For creation of identical variable groups within one data set. The first group of the variables would be created in the first data set, and then transferred to a second data set. From these the same block of variables can be tranferred to the first data set to predefined positions. This saves considerable time in creation of variable names. (Note: Copying a block of variables within a data set is not advisable, as it is considered a potentially risky operation).

FUNCTION = 22 Run Another Sequence of Operations

This function enables the user to run several sequences of operations with one command. This can be done by inserting the number of a predefined sequence within a sequence of operations. After the called sequence has been executed the calling sequence will continue execution of defined operations.

FUNCTION = 23 Print a Table Using a Predefined Format

In the manual mode, this function is not shown on the menu. It allows for the printing of tables corresponding, for example, to the results of calculations, to be included in a sequence.

FUNCTION = 24 End of Processing

This function returns control the the Data Processing Menu. It also terminates the creation of new operations in a sequence currently being defined.

FUNCTION = 25 Print/Display a List of Variable Names

This option allows the user to print or display the names of a block of variables or all variables for example when editing or defining a sequence of operations. Once the list has been printed or displayed, the menu of functions reappears on the screen, and the user can continue the sequence.

FUNCTION = 26 Data Editing (correcting headings or variables)

This option gives the user access to the data editing module where headings, variable names and values can be altered. Variables can also be added or deleted and the life of the project changed. After the user has carried out the editing required, the menu of functions reappears on the screen and the user can continue defining or editing a sequence, or performing operations (in 'manual' mode).

MANIP

PROJECT DATA MANIPULATION PACKAGE

USER'S GUIDE

(February 1985)

Sample Runs of MANIP

Description: This appendix gives the "screen" picture of the interaction between user and the MANIP programme.

Contents: Page

START WITH PREVIOUSLY STORED DATA

```
@MANIP
             ******    M A N I P   ******

          PROJECT DATA MANIPULATION PACKAGE
 VERSION 7.1 -  COPYRIGHT FAO INVESTMENT CENTER - ROME

Do you want a general description of the program?  (Y or N)
N
Do you want to use data which have previously been stored
in your set of personal files ?
Y
Give the generic name of your files (1 to 14 characters)
MS-CAM
Are you absolutely certain that the name is correct ?
Y
MS-CAM-W:TEXT         is to be used as OUTPUTFILE
Note : The old content of that file is erased at the start of a new run
Press CR key when you want to continue


CAMEROUN - PROJET DE DEVELOPPEMENT DU CENTRE-SUD

Current data set is ANALYSE ECON. ET FINANCIERE



Code = 1     Data processing
Code = 2     Data editing (correcting headings or variables)------------
Code = 3     Data listing (producing tables of variables)   | Main menu |
Code = 4     Print/display a list of variable names         ------------
Code = 5     Display MANIP user's guide
Code = 6     Exit MANIP


Enter a number between  1 and  6
```

START WITH A NEW DATA

```
@MANIP

              ******    M A N I P   ******

          PROJECT DATA MANIPULATION PACKAGE
 VERSION 7.1 -  COPYRIGHT FAO INVESTMENT CENTER - ROME

Do you want a general description of the program?  (Y or N)
N
Do you want to use data which have previously been stored
in your set of personal files ?
N
Before entering a new data set you have to give an unused
name with 1 to 14 characters for the files where your
data will be stored automatically
Give the generic name of your files (1 to 14 characters)
SAMPLE-RUN
Are you absolutely certain that the name is correct ?
Y
"SAMPLE-RUN-W:TEXT"     is to be used as OUTPUTFILE
Press CR key when you want to continue

          Input of data

You are starting with a new data set          | Initial input of data |
Enter the following information :
Project name (max 50 chrs) ?
SAMPLE RUN
Data set name (max 30 chrs) ?
RICE
Currency name (max 10 chrs) ?
000 B
Project life  ?

Enter a number between  1 and 50
20
How many variables do you intend to enter now ?

Enter a number between  0 and1500
3

Var. No    1 - Name of variable (max 25 chrs) ?
RICE YIELD

RICE YIELD

ALTERNATIVES OF ENTERING VARIABLE VALUES

Code = 1     Actual values
Code = 2     Linear interpolation
Code = 3     Percentages
Code = 4     Compound rate
Code = 5     S-curve
```

```
Enter a number between  1 and  5
2
Value of variable RICE YIELD in year 0 ?
1.1
Which year will this value change ?

Enter a number between  1 and 20
2
Until which year will the values be changing ?

Enter a number between  2 and 20
7
Value of variable RICE YIELD in year  7 ?
2.2
RICE YIELD
   0 :       1.1   1 :       1.1   2 :       1.3   3 :       1.5   4 :       1.7
   5 :       1.8   6 :       2.0   7 :       2.2   8 :       2.2   9 :       2.2
  10 :       2.2  11 :       2.2  12 :       2.2  13 :       2.2  14 :       2.2
  15 :       2.2  16 :       2.2  17 :       2.2  18 :       2.2  19 :       2.2
  20 :       2.2

Are the values correct ?
Y

Var. No    2 - Name of variable (max 25 chrs) ?
RICE AREA

RICE AREA

ALTERNATIVES OF ENTERING VARIABLE VALUES

Code = 1     Actual values
Code = 2     Linear interpolation
Code = 3     Percentages
Code = 4     Compound rate
Code = 5     S-curve
Code = 6     By reference

Enter a number between  1 and  6
1
Which year do the values reach a constant level ?
(If never, enter 20)

Enter a number between  0 and 20
3
Value of variable RICE AREA in year  0 = ?
3
Value of variable RICE AREA in year  1 = ?
4.5
Value of variable RICE AREA in year  2 = ?
5
Value of variable RICE AREA in year  3 = ?
6
```

Generating data using
linear interpolation

```
RICE AREA
   0 :        3.0   1 :        4.5   2 :        5.0   3 :        6.0   4 :        6.0
   5 :        6.0   6 :        6.0   7 :        6.0   8 :        6.0   9 :        6.0
  10 :        6.0  11 :        6.0  12 :        6.0  13 :        6.0  14 :        6.0
  15 :        6.0  16 :        6.0  17 :        6.0  18 :        6.0  19 :        6.0
  20 :        6.0

Are the values correct ?
Y

Var. No    3 - Name of variable (max 25 chrs) ?
RICE - COST/HA

RICE - COST/HA

ALTERNATIVES OF ENTERING VARIABLE VALUES

Code = 1     Actual values
Code = 2     Linear interpolation
Code = 3     Percentages
Code = 4     Compound rate
Code = 5     S-curve
Code = 6     By reference

Enter a number between  1 and  6
1
Which year do the values reach a constant level ?
(If never, enter 20)

Enter a number between  0 and 20
4
Value of variable RICE - COST/HA in year  0 = ?
75
Value of variable RICE - COST/HA in year  1 = ?
90
Value of variable RICE - COST/HA in year  2 = ?
120
Value of variable RICE - COST/HA in year  3 = ?
140
Value of variable RICE - COST/HA in year  4 = ?
150

RICE - COST/HA
   0 :       75.0   1 :       90.0   2 :      120.0   3 :      140.0   4 :      150.0
   5 :      150.0   6 :      150.0   7 :      150.0   8 :      150.0   9 :      150.0
  10 :      150.0  11 :      150.0  12 :      150.0  13 :      150.0  14 :      150.0
  15 :      150.0  16 :      150.0  17 :      150.0  18 :      150.0  19 :      150.0
  20 :      150.0

Are the values correct ?
Y
```

Main menu follows

LISTING OF VARIABLES
Code 4 of Main Menu

```
       Listing of variables

Code = 1      All variables
Code = 2      Block of variables
Code = 3      No more listing

Enter a number between  1 and  3
1
You have the following options :
Code = 1      Terminal CRT
Code = 2      Line-printer (132 chrs/l)
Code = 3      HP-printer (227 chrs/l)
Code = 4      Outputfile (227 chrs/l)

Enter a number between  1 and  4
1

Variables with their numbers :

    1  RICE YIELD                    2  RICE AREA
    3  RICE - COST/HA          Press CR key when you want to continue

       Listing of variables

Code = 1      All variables
Code = 2      Block of variables
Code = 3      No more listing

Enter a number between  1 and  3
3

SAMPLE RUN

Current data set is RICE

Code = 1      Data processing
Code = 2      Data editing (correcting headings or variables)
Code = 3      Data listing (producing tables of variables)      Back to
Code = 4      Print/display a list of variable names             Main Menu
Code = 5      Display MANIP user's guide
Code = 6      Exit MANIP
```

CHANGING THE DATA

DATA EDITING
Code 2 of Main Menu

```
Code = 1     Headings
Code = 2     Variable name
Code = 3     Variable values
Code = 4     Delete a variable
Code = 5     Add new variables
Code = 6     Change project life
Code = 7     No more editing

Enter a number between  1 and  7
3
Variable number = ?
Highest variable defined =   3
1

RICE YIELD
   0 :      1.1   1 :      1.1   2 :      1.3   3 :      1.5   4 :      1.7
   5 :      1.8   6 :      2.0   7 :      2.2   8 :      2.2   9 :      2.2
  10 :      2.2  11 :      2.2  12 :      2.2  13 :      2.2  14 :      2.2
  15 :      2.2  16 :      2.2  17 :      2.2  18 :      2.2  19 :      2.2
  20 :      2.2

Code = 1     Alter individual values
Code = 2     Enter another time series
Code = 3     No more editing
Code = 4     Change number of dec. digits in display (Current = 1)

Enter a number between  1 and  4
1
Year = ?

Enter a number between  0 and 20
8
Old value =        2.20000
Correct value = ?
2.3
Should this value be repeated for the following years ?
Y
```

```
RICE YIELD
   0 :      1.1    1 :      1.1    2 :      1.3    3 :      1.5    4 :      1.7
   5 :      1.8    6 :      2.0    7 :      2.2    8 :      2.3    9 :      2.3
  10 :      2.3   11 :      2.3   12 :      2.3   13 :      2.3   14 :      2.3
  15 :      2.3   16 :      2.3   17 :      2.3   18 :      2.3   19 :      2.3
  20 :      2.3

Code = 1     Alter individual values
Code = 2     Enter another time series
Code = 3     No more editing
Code = 4     Change number of dec. digits in display (Current = 1)

Enter a number between  1 and  4
3

                                 DATA EDITING

Code = 1     Headings
Code = 2     Variable name
Code = 3     Variable values
Code = 4     Delete a variable
Code = 5     Add new variables
Code = 6     Change project life
Code = 7     No more editing

Enter a number between  1 and  7
7

SAMPLE RUN

Current data set is RICE

Code = 1     Data processing
Code = 2     Data editing (correcting headings or variables)  +-----------+
Code = 3     Data listing (producing tables of variables)     | Back to   |
Code = 4     Print/display a list of variable names           | Main Menu |
Code = 5     Display MANIP user's guide                       +-----------+
Code = 6     Exit MANIP

Enter a number between  1 and  6
```

PROCESSING THE DATA

DATA PROCESSING
Code 1 of Main Menu

```
Current data set is :
Project  = SAMPLE RUN
Data set = RICE
Currency = 000 B

Code = 1     Manual operation                              | Two modes:
Code = 2     Define a sequence of operations               | - direct (manual)
Code = 3     Print/display the definition of a sequence    | - programmes (de-
Code = 4     Edit a pre-defined sequence of operations     |   fine sequence)
Code = 5     Run a pre-defined sequence of operations
Code = 6     Print list/names of pre-defined sequences
Code = 7     End of processing

Enter a number between  1 and  7
2                                          | Select define sequence |

This is your sequence No.  1 to be defined
You have two options :
  1 : Copy an existing sequence
  2 : Define a new sequence

Enter a number between  1 and  2
2
Give a name to the sequence (max 25 chrs)
RICE

In the following list, X and Y refer to code number of variables
and A, B and C to parameters
Operations 3 to 7 and 9 to 12 are on blocks of variables, (X is a block,
Y is either a block or a single variable)

 1. Display/Print X                     11. AX/Y + B
 2. Rate of return on X                 12. AX + BY + C
 3. Incremental values over year 0      13. If X > 0, AX + BY + C
 4. Cumulative values of X              14. If X < 0, AX + BY + C
 5. Add X over A years                  15. If X and Y >= 0, AX + BY + C
 6. Delay X by A years                  16. Carry over if AX + BY + C < 0
 7. Advance X by A years                17. Depreciation on X
 8. Interest on overdraft               18. Debt service on X
 9. AX + B                              19. Phasing a block of variables
10. AXY + B                             20. Balance & switching values

           21. Read from/write to another data set
           22. Run another sequence of operations
           23. Print a table using a predefined format
           24. End of processing
           25. Print/display a list of variable names
           26. Data editing (correcting headings or variables)

| Menu of available operations |
```

```
Operation No. 1 in the sequence : Code = ?
10
Selected operation = AXY + B
Variable X
Variable number = ?
Highest variable defined =   3
1
Variable Y
Variable number = ?
Highest variable defined =   3
2
Number of variables in the X block ?
Enter a number between  1 and  3
1
Should Y be treated as a single variable ?
Y
Parameter A = ?
1
Parameter B = ?
0
The result will be stored in variable Z
No. of variable Z = ? (Highest defined =   3)
Enter a number between  1 and1500
4
Name of variable No     4 (max 25 chrs) ?
PROD (ANNUAL MODE)

AXY + B

X =  1 RICE YIELD
Y =  2 RICE AREA
Parameter A =  1.00
Parameter B =  0.00
Result in variable No.  4 prod (annual mode)
Is this correct ?
Y

No.  1 in the sequence :
Selected operation No. 10 :
AXY + B

X =  1 RICE YIELD
Y =  2 RICE AREA                         | First operation defined |
Parameter A =  1.00
Parameter B =  0.00
Result in variable No.  4 prod (annual mode)
Press CR key when you want to continue

You may have 49 more operations in the seq.

In the following list, X and Y refer to code number of variables
and A, B and C to parameters
Operations 3 to 7 and 9 to 12 are on blocks of variables, (X is a block,
Y is either a block or a single variable)
```

```
--------------------------------------------------------------------------
| 1. Display/Print X                        11. AX/Y + B                  |
| 2. Rate of return on X                    12. AX + BY + C               |
| 3. Incremental values over year 0         13. If X > 0, AX + BY + C     |
| 4. Cumulative values of X                 14. If X < 0, AX + BY + C     |
| 5. Add X over A years                     15. If X and Y >= 0, AX + BY + C |
| 6. Delay X by A years                     16. Carry over if AX + BY + C < 0 |
| 7. Advance X by A years                   17. Depreciation on X         |
| 8. Interest on overdraft                  18. Debt service on X         |
| 9. AX + B                                 19. Phasing a block of variables |
|10. AXY + B                                20. Balance & switching values |
|                                                                         |
|              21. Read from/write to another data set                   |
|              22. Run another sequence of operations                    |
|              23. Print a table using a predefined format               |
|              24. End of processing                                     |
|              25. Print/display a list of variable names                |
|              26. Data editing (correcting headings or variables)       |
--------------------------------------------------------------------------
                ↑
    --------------------------------
    | Menu of available operations |
    --------------------------------

Operation No.  2 in the sequence : Code = ?
19
NOTICE :  This routine calculates aggregated values of
a group of time series (unit variables, in one block)
- maximum size of block is 50 -
phased in by a certain number of units year by year (phasing variable).

There are two options :

1 - Without project situation is constant, represented by year 0 values :
    Only one block of unit variables is required, results are TOTAL values

2 - Without project situation is not constant.  Two blocks of unit variables
    are required, for with and without project respectively.
    On option, the results are either TOTAL or INCREMENTAL

    It is assumed that unit and phasing variables are already part of the
    data set. If not, you should enter these variables now

Are any of the required variables missing in your data set ?
N

Is WITHOUT PROJECT situation constant ?
Y
FIRST unit variable
Variable number = ?
Highest variable defined =   4
1
LAST unit variable
Variable number = ?
Highest variable defined =   4
1
```

```
Variables to be aggregated :     No.    Name

                                   1   RICE YIELD
Phasing variable
Variable number = ?
Highest variable defined =   4
2

Phasing data are in variable No.   2   RICE AREA
Code = 1     Number of units is TOTAL
Code = 2     Number of units is INCREMENTAL

Enter a number between  1 and  2
1
Number of units is total
Phasing period (max 20 years) = ?
Enter a number between  1 and 20
3
Phasing over years 1 to  3
The results will be stored in a block of  1 consecutive variables
First variable in the block
5
The result will be stored in variables :
                                         5
Is this correct ?
Y
Variable No.   5
Name of variable No     5 (max 25 chrs) ?
PROD (PHASING MODE)

No.  2 in the sequence :                         | Second operation defined |
Selected operation No. 19 :
Phasing a block of variables

Unit variables
   1 RICE YIELD
Phasing data in variable No.  2 RICE AREA
TOTAL over years 1 to  3
Result stored in variables :
   5 PROD (PHASING MODE)

Press CR key when you want to continue

You may have 48 more operations in the seq.

In the following list, X and Y refer to code number of variables
and A, B and C to parameters
Operations 3 to 7 and 9 to 12 are on blocks of variables, (X is a block,
Y is either a block or a single variable)
```

```
-------------------------------------------------------------------------------
| 1. Display/Print X                          11. AX/Y + B                    |
| 2. Rate of return on X                      12. AX + BY + C                 |
| 3. Incremental values over year 0           13. If X > 0, AX + BY + C       |
| 4. Cumulative values of X                   14. If X < 0, AX + BY + C       |
| 5. Add X over A years                       15. If X and Y >= 0, AX + BY + C|
| 6. Delay X by A years                       16. Carry over if AX + BY + C < 0
| 7. Advance X by A years                     17. Depreciation on X           |
| 8. Interest on overdraft                    18. Debt service on X           |
| 9. AX + B                                   19. Phasing a block of variables|
|10. AXY + B                                  20. Balance & switching values  |
|                                                                             |
|              21. Read from/write to another data set                        |
|              22. Run another sequence of operations                         |
|              23. Print a table using a predefined format                    |
|              24. End of processing                                          |
|              25. Print/display a list of variable names                     |
|              26. Data editing (correcting headings or variables)            |
-------------------------------------------------------------------------------
                 ^
   --------------------------------
   | Menu of available operations |
   --------------------------------


Operation No.  3 in the sequence : Code = ?   ------------------------------
24                                             | End of sequence definition |
You have now stored sequence No.  1            ------------------------------
It includes  2 operations
Press CR key when you want to continue

                                   DATA PROCESSING

Current data set is :
Project  = SAMPLE RUN
Data set = RICE
Currency = 000 B

Code = 1      Manual operation
Code = 2      Define a sequence of operations
Code = 3      Print/display the definition of a sequence
Code = 4      Edit a pre-defined sequence of operations
Code = 5      Run a pre-defined sequence of operations
Code = 6      Print list/names of pre-defined sequences
Code = 7      End of processing

Enter a number between  1 and  7              -----------------
3                                             | List sequence |
Enter number of sequence                      -----------------

Enter a number between  0 and  1
1
 RICE                       Is it OK ?
Y
SELECT PRINTER
You have the following options :
```

```
Code = 1     Terminal CRT
Code = 2     Line-printer (132 chrs/l)
Code = 3     HP-printer (227 chrs/l)
Code = 4     Outputfile (227 chrs/l)

Enter a number between  1 and  4
1

Sequence of operations No    1  RICE

  1 : Selected operation =AXY + B

X =  1 RICE YIELD
Y =  2 RICE AREA
Parameter A =  1.00
Parameter B =  0.00
Result in variable No.  4 prod (annual mode)
Press CR key when you want to continue


  2 : Selected operation =Phasing a block of variables

Unit variables
   1 RICE YIELD
Phasing data in variable No.  2 RICE AREA
TOTAL over years 1 to  3
Result stored in variables :
   5 prod (phasing mode)

                                  DATA PROCESSING

Current data set is :
Project  = SAMPLE RUN
Data set = RICE
Currency = 000 B

Code = 1     Manual operation
Code = 2     Define a sequence of operations
Code = 3     Print/display the definition of a sequence
Code = 4     Edit a pre-defined sequence of operations
Code = 5     Run a pre-defined sequence of operations
Code = 6     Print list/names of pre-defined sequences
Code = 7     End of processing

Enter a number between  1 and  7          ------------------
4                                         | Modify sequence |
Enter number of sequence                  ------------------

Enter a number between  0 and  1
1
 RICE                       Is it OK ?
Y
```

```
RICE
The sequence includes  2 operations
Code = 1      Delete
Code = 2      Add/insert
Code = 3      Modify
Code = 4      Change seq name
Code = 5      End of editing

Enter a number between  1 and  5
2
Sequential number of the operation AFTER which to insert an op.


Enter a number between  0 and  2
2

In the following list, X and Y refer to code number of variables
and A, B and C to parameters
Operations 3 to 7 and 9 to 12 are on blocks of variables, (X is a block,
Y is either a block or a single variable)
-----------------------------------------------------------------------------
| 1. Display/Print X                       11. AX/Y + B                     |
| 2. Rate of return on X                   12. AX + BY + C                  |
| 3. Incremental values over year 0        13. If X > 0, AX + BY + C        |
| 4. Cumulative values of X                14. If X < 0, AX + BY + C        |
| 5. Add X over A years                    15. If X and Y >= 0, AX + BY + C |
| 6. Delay X by A years                    16. Carry over if AX + BY + C < 0|
| 7. Advance X by A years                  17. Depreciation on X            |
| 8. Interest on overdraft                 18. Debt service on X            |
| 9. AX + B                                19. Phasing a block of variables |
|10. AXY + B                               20. Balance & switching values   |
|                                                                           |
|              21. Read from/write to another data set                      |
|              22. Run another sequence of operations                       |
|              23. Print a table using a predefined format                  |
|              24. End of processing                                        |
|              25. Print/display a list of variable names                   |
|              26. Data editing (correcting headings or variables)          |
-----------------------------------------------------------------------------
                          ↑
        -------------------------------
        | Menu of available operations |
        -------------------------------

Operation No.  3 in the sequence : Code = ?
9
```

```
Selected operation = AX + B
Variable X
Variable number = ?
Highest variable defined =   5
5
Number of variables in the X block ?
1
Parameter A = ?
185
Parameter B = ?
0
The result will be stored in variable Z
No. of variable Z = ? (Highest defined =   5)
Y
 ILLEGAL NUMBER SYNTAX                        | Watch what you are doing! |
6
Name of variable No     6 (max 25 chrs) ?
PROD VALUE (PH)

AX + B

X =  5 prod (phasing mode)              | Third operation now in the sequence |
Parameter A =185.00
Parameter B =  0.00
Result in variable No.  6 prod value (ph)
Is this correct ?
Y


RICE
The sequence includes  3 operations
Code = 1      Delete
Code = 2      Add/insert
Code = 3      Modify
Code = 4      Change Seq Name
Code = 5      End of editing

Enter a number between  1 and  5
4
RICE                                         | Checking the name of the |
Do you want to change seq name ?             | sequence without editing |
N

RICE
The sequence includes  3 operations
Code = 1      Delete
Code = 2      Add/insert
Code = 3      Modify
Code = 4      Change Seq Name
Code = 5      End of editing

Enter a number between  1 and  5
5
                              DATA PROCESSING

Current data set is :
Project  = SAMPLE RUN
Data set = RICE
Currency = 000 B
```

```
Code = 1     Manual operation
Code = 2     Define a sequence of operations
Code = 3     Print/display the definition of a sequence
Code = 4     Edit a pre-defined sequence of operations
Code = 5     Run a pre-defined sequence of operations
Code = 6     Print list/names of pre-defined sequences
Code = 7     End of processing

Enter a number between  1 and  7
5
Enter number of sequence

Enter a number between  0 and  1
1
 RICE                       Is it OK ?
Y
The sequence of operations stored in RICE
is being performed
Do you want to suppress listing of operations ?
Y
```

The calculation is performed

```
                                   DATA PROCESSING

Current data set is :
Project  = SAMPLE RUN
Data set = RICE
Currency = 000 B

Code = 1     Manual operation
Code = 2     Define a sequence of operations
Code = 3     Print/display the definition of a sequence
Code = 4     Edit a pre-defined sequence of operations
Code = 5     Run a pre-defined sequence of operations
Code = 6     Print list/names of pre-defined sequences
Code = 7     End of processing

Enter a number between  1 and  7
6
SELECT PRINTER
You have the following options :
Code = 1     Terminal CRT
Code = 2     Line-printer (132 chrs/l)
Code = 3     HP-printer (227 chrs/l)
Code = 4     Outputfile (227 chrs/l)

Enter a number between  1 and  4
1
LIST OF SEQUENCES OF OPERATIONS IN FILE

     1  RICE

Press CR key when you want to continue
```

Which sequences are already defined?

```
                                   DATA PROCESSING

Current data set is :
Project  = SAMPLE RUN
Data set = RICE
```

```
Currency = 000 B

Code = 1      Manual operation
Code = 2      Define a sequence of operations
Code = 3      Print/display the definition of a sequence
Code = 4      Edit a pre-defined sequence of operations
Code = 5      Run a pre-defined sequence of operations
Code = 6      Print list/names of pre-defined sequences
Code = 7      End of processing

Enter a number between  1 and  7
7
                        ---------------------
                        | Main menu follows |
                        ---------------------
```

PRINTING TABLES
DATA LISTING
Code 3 of main menu

DATA LISTING OPTIONS

```
Code = 1     Standard output of all/block of data
Code = 2     Produce a table using a previously defined format
Code = 3     Define a table format
Code = 4     Print/display previously defined format
Code = 5     Edit a previously defined format
Code = 6     Print List/Names of defined formats
Code = 7     End of output operations

Enter a number between  1 and  7
3
```

Defining a table

```
DEFINITION OF TABLE FORMAT No    1

You have two options :
 1 : Copy an existing table format
 2 : Define a new format

Enter a number between  1 and  2
2
Where is the table to be printed ?
You have the following options :
Code = 1     Terminal CRT
Code = 2     Line-printer (132 chrs/l)
Code = 3     HP-printer (227 chrs/l)
Code = 4     Outputfile (227 chrs/l)

Enter a number between  1 and  4
1
Do you want a sub-title for the table ?
Y
Enter the sub-title (max 80 chrs)
RICE PRODUCTION - PHASING
Do you want to assign a number to the table ?
Y
Table No. = ?
1
You may include up to 50 variables in the table

Data printing options :
Code = 1     Block of variables
Code = 2     Selected variables

Enter a number between  1 and  2
2
Do you want a list of variables ?
N


How many variables do you want to enter ?
3
Variable to be in line  1 of the table
Variable number = ?
```

```
Highest variable defined =   6
1
    1 RICE YIELD                 OK ?
Y
Variable to be in line  2 of the table
Variable number = ?
Highest variable defined =   6
2
    2 RICE AREA                  OK ?
Y
Variable to be in line  3 of the table
Variable number = ?
Highest variable defined =   6
5
    5 PROD (PHASING MODE)        OK ?
Y
Enter the number of years (<=21) to be printed

Enter a number between  1 and 21
21

All years to be printed
Is it correct ?Y

Do you want scaling of your data ?
N

How many decimals do you want to be printed ?

Enter a number between  0 and  8
1
Do you want totals to be printed for each line ?
N

Do you want any blank lines in the table ?
N

                              LIST/EDIT FORMAT No.  1
You have the following options :
Code = 1     Terminal CRT
Code = 2     Line-printer (132 chrs/l)
Code = 3     HP-printer (227 chrs/l)
Code = 4     Outputfile (227 chrs/l)

Enter a number between  1 and  4
1
```

```
This table has been given No.  1
Sub-title of the table :
RICE PRODUCTION - PHASING
The table will be presented on the CRT
The data will be printed with 1 decimals
All years to be printed
Variables to be printed
Variables with their numbers :

 1    1 RICE YIELD               2    2 RICE AREA
 3    5 prod (phasing mode)
Press CR key when you want to continue

Do you want to change some format specifications ?
Y
                                          ----------------------------
1    Change table No.                     | Modifying the definition |
2    Change sub-title                     | of a table               |
3    Change printer                       ----------------------------
4    Change scale factor
5    Change number of decimals
6    Change line totals
7    Change years to be printed
8    Change variables to be printed
9    Specify blank lines in the table
10   End of editing

Enter a number between  1 and 10
8
Variables with their numbers :

 1    1 RICE YIELD               2    2 RICE AREA
 3    5 prod (phasing mode)
Press CR key when you want to continue

Code = 1     Replace a variable
Code = 2     Add a variable
Code = 3     Delete a variable
Code = 4     No more changes

Enter a number between  1 and  4
2
Number of line after which a new variable should be inserted ?

Enter a number between  0 and  3
3
No. of variable to be inserted ?
6
Variables with their numbers :

 1    1 RICE YIELD               2    2 RICE AREA
 3    5 prod (phasing mode)      4    6 prod value (ph)
Press CR key when you want to continue
```

```
Code = 1     Replace a variable
Code = 2     Add a variable
Code = 3     Delete a variable
Code = 4     No more changes

Enter a number between  1 and  4
4

1    Change table No.
2    Change sub-title
3    Change printer
4    Change scale factor
5    Change number of decimals
6    Change line totals
7    Change years to be printed
8    Change variables to be printed
9    Specify blank lines in the table
10   End of editing

Enter a number between  1 and 10
9
Do you want any blank lines in the table ?
Y
The following variables are included in the table :
Variables with their numbers :

 1    1 RICE YIELD                 2    2 RICE AREA
 3    5 prod (phasing mode)        4    6 prod value (ph)
Press CR key when you want to continue

Do you want a blank line after var. RICE YIELD                ?
N
Do you want a blank line after var. RICE AREA                 ?
Y
Do you want a blank line after var. PROD (PHASING MODE)       ?
N
Do you want a blank line after var. PROD VALUE (PH)           ?
N

1    Change table No.
2    Change sub-title
3    Change printer
4    Change scale factor
5    Change number of decimals
6    Change line totals
7    Change years to be printed
8    Change variables to be printed
9    Specify blank lines in the table
10   End of editing

Enter a number between  1 and 10
10
```

Specifying blank lines to improve presentation

```
This table has been given No.  1          | New definition of the table |
Sub-title of the table :
RICE PRODUCTION - PHASING
The table will be presented on the CRT
The data will be printed with 1 decimals
All years to be printed
Variables to be printed
Variables with their numbers :

 1    1 RICE YIELD                2    2 RICE AREA
 3    5 PROD (PHASING MODE)       4    6 PROD VALUE (PH)
Press CR key when you want to continue

A blank line will be inserted after :
 No. 2    2 RICE AREA
Do you want to change some format specifications ?
N

                    DATA LISTING OPTIONS

Code = 1     Standard output of all/block of data
Code = 2     Produce a table using a previously defined format
Code = 3     Define a table format
Code = 4     Print/display previously defined format
Code = 5     Edit a previously defined format
Code = 6     Print list/names of defined formats
Code = 7     End of output operations

Enter a number between  1 and  7
2                                         | Getting the table printed |

                                SELECT TABLE FORMAT

Input format number

Enter a number between  0 and  1
1                                         | The table is printed |
   1 RICE PRODUCTION - PHASING
Is it OK ?
Y

                    DATA LISTING OPTIONS

Code = 1     Standard output of all/block of data
Code = 2     Produce a table using a previously defined format
Code = 3     Define a table format
Code = 4     Print/display previously defined format
Code = 5     Edit a previously defined format
Code = 6     Print list/names of defined formats
Code = 7     End of output operations

Enter a number between  1 and  7
7

                       | Main menu follows |
```

ANNEX 3

13 January 1983

RPV - RATE OF RETURN/PRESENT VALUE

DEBT - DEBT SERVICE CALCULATIONS

TS - TIME SERIES GENERATOR

PHASE - PHASING CALCULATIONS

FOUR PROGRAMMES FOR HP 41 CV CALCULATOR

by M. Siméon

FAO / WORLD BANK
COOPERATIVE PROGRAMME

INVESTMENT CENTRE

FOOD AND AGRICULTURE ORGANIZATION OF THE UNITED NATIONS
ROME

RPV - Rate of Return/Present Value

A Programme for HP 41 C Calculator

I. Definitions

The programme allows to perform calculations on a number of so-called "variables".

- A variable is a yearly time series of values associated to a name (1 to 6 characters), representing flows of benefits/expenditures in a project (costs are specified as negative numbers). The series includes year Ø which is used to represent the without project situation, when it is assumed to be constant.

- Project life (PLIFE) is the period considered for the analysis (typically twenty years, in addition to year Ø).

- Maximum development period (MAXDEV). Usually, during project life, variable values are changing over the initial years (called the development period) and then are constant. MAXDEV is the longest development period for a given set of variables. It is used in the programme to save memory when organizing storage of data.

- Opportunity cost of capital (OCC) is the discount rate to be specified for present values/switching percentages calculations. It is to be specified in percentage, e.g. 12%.

- Maximum number of variables (MAXNV). The maximum number of variables to be part of a given data set needs to be specified, in order for the programme to organize data storage and to check whether enough memory is available. The programme can then be run with any number of variables up to this maximum.

This maximum depends on the maximum development period. With the maximum memory configuration (319 registers), MAXDEV is 28 years for 20 variables, 19 for 30 variables, 9 for 40 variables.

II. Purpose of the Programme

The programme allows to input names and values corresponding to a number of variables, calculate the balance and the incremental balance (i.e. balance over year Ø value) on the current number of variables, calculate the internal rate of return on the incremental balance, and calculate present values and switching percentages (1) on:

- Each variable (on total values from year 1 to PLIFE, not on incremental values).

---

(1) Percentage change to be applied to a variable, the others being constant, in order to bring net balance present value to zero, i.e. to make the internal rate of return equal to the opportunity cost of capital.

- Without project situation, corresponding to the projection over PLIFE of the year Ø balance.

- Net (i.e. incremental) balance.

Variables can be listed at the time of input, and name and values can be changed at that time.

Variable name and values can also be edited (modified) afterwards, but then the change in value (positive if increase, negative if decrease) is to be input instead of the new value itself.

Variables can be stored on magnetic card, as well as Balance. Previously stored variables - including Balance from a previous run, or variables created by compatible programmes, (1) can be used as input variables to RPV.

III. Organization of the Programme

The programme is - from the user's point of view - organized into eight modules, corresponding to the various functions that can be performed.

The starting module (labelled RPV) is always to be run first when a new set of data is to be worked on. This module will call for the key variables necessary to organize data storage, i.e. MAXDEV and MAXNV, check on memory availability and call for the input of PLIFE and OCC.

The other seven modules are:

	Label
- input of variables	INP
- listing of balance	BL
- storage of balance on card	STB
- listing of incremental balance	IBL
- calculation of present values and switching values	SW
- calculation of internal rate of return on incremental balance	RE
- editing of variable values or name	EDT

Once at least one variable has been input, the seven modules can be run successively in any order. Calculations will be performed on the current number of variables.

IV. Use of the Printer

The programme can be used with or without printer.

---

(1) PHASE, DEBT, Time Series. See corresponding manuals.

If the printer is connected and switched on, on the position MAN, variables and results will be printed automatically when the proper function is selected. If the printer is not connected, or if it is connected but switched OFF, the programme will stop every time a result is displayed, to give time to the user to write down the information, and will continue when R/S key is pressed.

V. Running the Programme (See sample run in Appendix 1)

1. Loading the programme

The programme is stored on seven magnetic cards labelled RPV1 to RPV7. The card reader being inserted, perform the following steps:

- Switch on the calculator.
- Set user mode.
- Set run mode.
- If necessary, clear memory from other programmes.
- Execute SIZE 020.
- Read side 1 of card RPV1, then press GTO.. for packing.
- Read cards RPV2 and RPV3, and press GTO..
- Do the same for cards RPV4 to RPV7, i.e. after each card is read press GTO..
- Install on the keyboard an overlay indicating assignment of functions to keys.

2. Assigned keys

The labels corresponding to the eight functional modules of the programme (Chapter III) are assigned to the following keys, for user mode :

Label:	RPV	INP	BL	IBL	SW	RE	EDT	STB
Key:	X<->Y	R↓	SIN^{-1}	COS^{-1}	SIN	COS	TAN	%

In addition, the function SIZE is assigned to RCL.

3. Starting

The calculator being in user mode, press RPV (X<->Y) key. At least 20 registers should be assigned for data; if not, an error will occur. Input data as requested by prompts, and press R/S key.

After the input of MAXNV and MAXDEV, the programme will check if enough memory is allocated for data. If not, it will display "XEQ SIZE nn", nn being the required number of data registers. (1) When the proper number

---

(1) nn = 2 MAXDEV + 2 MAXNV + 15 if MAXDEV >= 8
nn = MAXDEV + 2 MAXNV + 23 if MAXDEV < 8

of data registers is allocated, press R/S. The programme will then ask to input PLIFE and OCC% and display READY.

Remark: any value is accepted for OCC, except -100%.

4. Input of variables - constant factor - listing - changes - storage on card

When pressing INP (R ↓) and within the limits of MAXNV, the programme will display "CARD ?", allowing to input a variable from a previously recorded magnetic card. Pressing R/S, or Y followed by R/S, will cause the programme to display "CARD" and wait for a card to be passed through the card reader.

If MAXDEV for the data on card is more than the current value, the data will be rejected and "WRONG DATA" will be displayed. Press R/S to get "READY" again.

If MAXDEV of the data on card is less than the current value, the last year stored on card will be copied into the remaining years.

If "N" was answered to the question "CARD ?", the programme will call for the name of the variable and then for the values from year zero to MAXDEV.

Input costs as negative values. If the values of a variable become constant before the year MAXDEV, the input of PI (Pi number) is understood as a code to copy the previous year value till the end of the time series.

After input of values is complete, or a card has been read, the message "* k ?" is displayed to ask the user whether he wants all the values of the variable to be multiplied by a constant factor k. If yes, k = ? is displayed and the factor can be specified.

The user is then asked if he wants to print all the values of the variable just introduced, by the prompt "list ?".

Whether or not the data was listed, the programme will then ask "CHANGE?". If the answer is yes (by just pressing R/S) the programme first displays the name of the variable, allowing to type a new name if required, and then asks "CARD?", allowing to input a new variable from another card as described above. If N is answered, the programme asks to input the number of the year to be changed, and then displays the old value (e.g. YR3 = 60.00?). Type the new value and press R/S, or just press R/S if there is no change. The question "*k?" will then come again, and then LIST? and CHANGE? as above. At the end of input/listing/changes, the message "STORE?" will be displayed. If the answer is R/S or Y followed by R/S, the programme will display "RDY 1 of n", where n is the number of card sides required to store the variable. Insert a blank (unclipped) card through the reader to record the data. Finally, the calculator will work for 10-20 seconds and then display READY.

5. Balance and incremental balance

By pressing BL (key SIN^{-1}, i.e. SHIFT SIN) or IBL (COS^{-1}), the programme will list either the balance or the incremental balance (i.e. over year zero) of the current number of variables (assuming that at least one variable was input).

6. Store balance

By pressing "STB" (%) key, the programme will display "RDY 1 of n" as above, to record the current balance.

7. Present values and switching values

By pressing SW (SIN), the programme will calculate for 10-20 seconds and then will list the following results:

- for each variable: variable name, present value of the total flow from year 1 to PLIFE, and switching percentage;

- for the without project situation, i.e. the year zero balance projected over the period 1 to PLIFE, present value and switching percentage;

- present value of the incremental balance;

- opportunity cost of capital (as a remainder) used for the calculations.

Notes:

a) The switching percentage of a variable is the percentage of change to be applied to all values of a variable (from year 1 to PLIFE), all other variables remaining constant, to make Incremental Balance Present Value equal to zero. The percentage is positive if the variable is to be increased, negative if it is to be decreased.

b) Without project present value is shown negative, i.e. as a cost to the project, when the value of year zero balance is positive.

8. Internal rate of return

The rogramme, by pressing RE (COS) key, will calculate the IRR - if it exists - within a given interval. The user needs to input the interval (to the questions GUESS1? and GUESS2?), as percentages (e.g. for 10%, input 10). If there is no solution within the given interval, the programme will display READY directly. If there is a solution, it will be calculated and listed.

Notes:

a) The calculation might take as long as two minutes.

b) The calculation will be faster if the interval is smaller.

c) The programme would not work properly if the values of incremental balance were so small that Present Value would be less than 1 with any discount rate.

9. Changing data

Although all variable values are not stored in the calculator, data can be corrected by pressing key EDT (TAN). The user will be asked to input the variable number (VAR NBR?), the year to be corrected (YR?), and the change in value (not the new value). Before asking YR? the programme will display variable name, allowing for changing it.

Examples:

old value = 50 - new value 65

to the prompt YRn CHGE?, input 15

- old value = 50 - new value = 35 - change = -15.

The programme will run some calculations and display READY.

10. Current number of variables

When the programme has stopped displaying READY, it is possible to check on the number of variables that have been already input by executing RCLO3.

11. Switching OFF and ON again

When the programme has stopped displaying READY, it is possible to switch OFF the calculator, and to resume the calculations after some time by switching ON again, and then select one of the appropriate keys, provided of course that the content of the data registers was not changed by running another programme in the meantime (it is possible to perform calculations if only the operating registers are used).

12. Data storage organization on magnetic card and programme listing

Data on card is organized the following way:

1st data : Variable name
2nd data : MAXDEV
Following data : variable values from year Ø to MAXDEV.

An annotated listing of the programme is presented in Appendix 2.

RPV - Rate of Return/Present Values Programme

SAMPLE RUN

A. Printer in mode NORM

1. Starting Module (RPV) (having or not having to execute size)

```
            XEQ "RPV"
MAXNV?
         30.00    RUN
MAXDEV?
         10.00    RUN
PLIFE?
         20.00    RUN
OCC %
         12.00    RUN
READY
```

```
            XEQ "RPV"
MAXNV?
         30.00    RUN
MAXDEV?
         10.00    RUN
XEQ SIZE 94
            SIZE 094
                  RUN
PLIFE?
         20.00    RUN
OCC %
         12.00    RUN
READY
```

2. Input of a variable, listing, no editing No use of constant factor

```
                    XEQ "INP"
VAR1-CARD ?
N                       RUN
VAR1-NAME?
NET WH                  RUN
NET WH-YR0=?
                 10.00  RUN
NET WH-YR1=?
               -150.00  RUN
NET WH-YR2=?
                 25.00  RUN
NET WH-YR3=?
                 30.00  RUN
NET WH-YR4=?
                 45.00  RUN
NET WH-YR5=?
                 52.00  RUN
NET WH-YR6=?
                 64.00  RUN
NET WH-YR7=?
                 55.00  RUN
NET WH-YR8=?
                 58.00  RUN
NET WH-YR9=?
                 60.00  RUN
NET WH-YR10=?
                 62.50  RUN
*K?
N                       RUN
LIST?
                        RUN
NET WH-VAR1
YR0=10.00
YR1=-150.00
YR2=25.00
YR3=30.00
YR4=45.00
YR5=52.00
YR6=64.00
YR7=55.00
YR8=58.00
YR9=60.00
YR10=62.50

CHANGE?
N                       RUN
STORE?
N                       RUN
READY
```

3. Input of a variable using PI for automatic repeat, k factor, editing

```
                    XEQ "INP"
VAR2-CARD ?
N                        RUN
VAR2-NAME?
INVEST                   RUN
INVEST-YR0=?
               0.00      RUN
INVEST-YR1=?
            -250.00      RUN
INVEST-YR2=?
            -350.00      RUN
INVEST-YR3=?
               0.00      RUN
INVEST-YR4=?
                          PI
                         RUN
*K?
                         RUN
K=?
               1.12      RUN
LIST?
N                        RUN
CHANGE?
                         RUN
INVEST
                         RUN
VAR2-CARD ?
N                        RUN
YR?
               2.00      RUN
YR2=-392.00?
            -397.00      RUN
*K?
N                        RUN
LIST?
                         RUN
INVEST-VAR2
YR0=0.00
YR[illegible].00
[illegible].00
[illegible]3=0.00
YR4=0.00
YR5=0.00
YR6=0.00
YR7=0.00
YR8=0.00
YR9=0.00
YR10=0.00

CHANGE?
N                        RUN
STORE?
N                        RUN
READY
```

4. Input of a variable from card - Listing

```
                    XEQ "INP"
VAR3-CARD ?
                         RUN
*K?
N                        RUN
LIST?
                         RUN
EXT SV-VAR3
YR0=0.00
YR1=-25.00
YR2=-31.00
YR3=-34.00
YR4=-34.00
YR5=-34.00
YR6=-34.00
YR7=-34.00
YR8=-34.00
YR9=-34.00
YR10=-34.00

CHANGE?
N                        RUN
STORE?
N                        RUN
READY
```

5. Editing data (EDT) (1)

```
            XEQ "EDT"
VAR NBR?
           2.00    RUN
INVEST
                   RUN
YR?
           2.00    RUN
YR2 CHGE?
         397.00 ENTER↑
           2.00     *
                   CHS
                   RUN
READY
```

6. Rate of return (RE) (2)

```
             XEQ "RE"
GUESS1?
          12.00    RUN
GUESS2?
          25.00    RUN
IRR=13.64%

READY
```

(1) To correct 397 into -397, the change is -794. Note that it can be calculated using the calculator.

(2) Calculated on data shown in A-2, A-3, A-4, B-1, after the editing presented in A-5.

B. Printer in mode MAN

1. Listing of a variable after input

```
OT INC-VAR4
YR0=25.00
YR1=45.00
YR2=77.00
YR3=110.00
YR4=146.30
YR5=152.70
YR6=152.70
YR7=152.70
YR8=152.70
YR9=152.70
YR10=152.70
```

2. Present values, switching percentages, and rate of return (1)

```
READY

NET WH-VAR1
PV=201.65
SWITCH %=-353.48

INVEST-VAR2
PV=66.49
SWITCH %=-1,072.09

EXT SV-VAR3
PV=-243.53
SWITCH %=292.68

OT INC-VAR4
PV=949.62
SWITCH %=-75.06

WITHOUT P
PV=-261.43
SWITCH %=272.65

INCR BAL
PV=712.79

OCC %=12.00

READY
IRR=49.25%

READY
```

3. Incremental balance (1) (2)

```
BALANCE-INCR
YR1=-445.00
YR2=433.00
YR3=71.00
YR4=122.30
YR5=135.70
YR6=147.70
YR7=138.70
YR8=141.70
YR9=143.70
YR10=146.20

READY
```

(1) On the data shown in A-2, A-3, A-4 and B-1, and before data editing shown in A-5.

(2) The value in year 2 reflects the error in the input of variable INVEST, year 2, corrected in A-5.

4. Balance and Incremental Balance (after editing in A-5)

```
BALANCE
YR0=35.00
YR1=-410.00
YR2=-326.00
YR3=106.00
YR4=157.30
YR5=170.70
YR6=182.70
YR7=173.70
YR8=176.70
YR9=178.70
YR10=181.20

READY
BALANCE-INCR
YR1=-445.00
YR2=-361.00
YR3=71.00
YR4=122.30
YR5=135.70
YR6=147.70
YR7=138.70
YR8=141.70
YR9=143.70
YR10=146.20

READY
```

5. Present values and switching percentages (after editing in A-5)

```
NET WH-VAR1
PV=201.65
SWITCH %=-39.58

INVEST-VAR2
PV=-566.49
SWITCH %=14.09

EXT SV-VAR3
PV=-243.53
SWITCH %=32.77

OT INC-VAR4
PV=949.62
SWITCH %=-8.40

WITHOUT P
PV=-261.43
SWITCH %=30.53

INCR BAL
PV=79.81

OCC %=12.00

READY
```

RPV - Rate of Return/Present Values
A Programme for HP41C Calculator

Annotated Programme Listing

1. Catalogue of Programmes and User Keys

```
           CAT 1
LBL'RPV
LBL'SN
END       109 BYTES
LBL'INP
LBL'STB
END       447 BYTES
LBL'SW
LBL'CN
END       215 BYTES
LBL'BL
LBL'IBL
LBL'PR
LBL'EDT
END       215 BYTES
LBL'RE
END       223 BYTES
LBL'O
LBL'YR
LBL'K
LBL'Y
LBL'N
LBL'T
LBL'R
LBL'V
LBL'PV
END       223 BYTES
.END.     03 BYTES
```

```
        PRKEYS
USER KEYS:
 21 "RPV"
 22 "INP"
-22 "STB"
 23 "SW"
-23 "BL"
 24 "RE"
-24 "IBL"
 25 "EDT"
 34 SIZE
```

Total: 1435 bytes
= 205 registers
Available for data: 114 registers

2. Card RPV 1

Starting Module

PRP "RPV"	
01♦LBL "RPV"	Ignore overflow errors - Set user mode
02 SF 24	
03 SF 27	
04 CLRG	
05 "MAXNV?"	MAXNV stored in 01
06 PROMPT	
07 STO 01	
08 "MAXDEV?"	
09 PROMPT	
10 STO 00	
11 8	Maximum between 8 and MAXDEV in X
12 X<=Y?	
13 X<>Y	
14 13	Address of year zero balance: MAX (8, MAXDEV) + 13
15 +	
16 STO 09	
17 RCL 00	Address of 1st variable Present value - 1
18 +	
19 STO 10	Size - 1
20 RCL 01	Size = MAXDEV + MAX (MAXDEV, 8) + 2 MAX NV + 15
21 2	
22 *	
23 +	
24♦LBL 01	
25 SF 25	
26 RCL IND X	
27 FS?C 25	Test size
28 GTO 02	
29 "XEQ SIZE	
30 1	
31 +	
32 XEQ "N"	
33 PROMPT	
34 1	
35 -	
36 GTO 01	
	Test size again
37♦LBL 02	
38 "PLIFE?"	
39 RCL 00	
40 PROMPT	
41 INT	Check P LIFE $\leq$ MAXDEV
42 X<Y?	
43 GTO 02	
44 STO 11	P LIFE stored in 11
45 GTO "0"	Input of OCC
46♦LBL "SN"	<u>SN</u>: Store name of variable with number in X
47 RCL 10	Address of name is calculated
48 +	
49 RCL 01	
50 +	
51 1	
52 +	
53 ASTO IND X	
54 END	

3. Card RPV 2/3 (INP)

PRP "INP"	Input/Listing/Changes of a Variable
01♦LBL "INP"	Current number of variables (NV) in 03
RCL 01 RCL 03 X=Y?	Test against MAXNV – XEQ 08: Input from card
GTO "R" 1 ST+ 03	Increment NV by 1 – Year counter to zero
XEQ 08 12 STO 04	Year zero of new variable stored in 12
"VAR" RCL 03 XEQ "N"	Register 04 used as counter/indirect address
"├-NAME?" AON PROMPT	Input of name
AOFF XEQ "SN"	
15♦LBL 04	
RCL 03 CLA XEQ "CN"	
"├-YR" RCL 00 INT	
XEQ "N" "├=?" PI	
PROMPT X=Y? GTO 00	Automatic repeat if PI is input
STO IND 04 1 ST+ 04	
ISG 00 GTO 04	04: Loop to input variable values
37♦LBL 05	05: Menu for constant factor, listing, editing, storage.
"*K?" XEQ "Y" FS?C 01	
XEQ 10 "LIST?" XEQ "Y"	
FS?C 01 XEQ 06	
"CHANGE?" XEQ "Y"	
FS?C 01 GTO 02	
"STORE?" XEQ "Y"	
FS?C 01 XEQ 01 13	
STO 04 RCL 10 RCL 03	PV address in 07 – Calculates PV of last variable
+ STO 07 XEQ "PV" 12	Set counter in 07
STO 04 RCL 09 STO 07	
XEQ "K"	
65♦LBL 07	07: Loop to add last variable to balance
RCL IND 04 ST+ IND 07	
1 ST+ 04 ST+ 07	
ISG 00 GTO 07 GTO "R"	
75♦LBL 02	02: Editing last variable
CLA RCL 03 XEQ "CN"	Name displayed – New name can be input
AON PROMPT AOFF	
ASTO IND X XEQ 08	
XEQ "YR" 12 + STO 04	
"├=" RCL IND 04 ARCL X	Old value of year to edit is displayed
"├?" PROMPT STO IND 04	New value can be input
GTO 05	
95♦LBL 06	06: Listing of variable
CF 03 CF 02 XEQ "PR"	
RTN	
100♦LBL 00	00: Automatic repeat if PI was input or after input from card
1 ST- 04 RCL IND 04	
STO 07 1 ST+ 04	
107♦LBL 03	03: Loop for automatic repeat
RCL 07 STO IND 04 1	
ST+ 04 ISG 00 GTO 03	
GTO 05	

3. Card RPV 2/3 (contd.)

```
115♦LBL 08
XEQ "K" "VAR?" RCL 03
XEQ "N" "F-CARD ?"
XEQ "Y" FC?C 01 RTN
RCL 11 STO 04 RCL 10
STO 05 10.012 RCL 00
+ RDTAX CLA ARCL 10
RCL 05 STO 10 RCL 03
XEQ "SN" RCL 00 1 E3
* RCL 04 X<> 11 X=Y?
GTO 05 X>Y? GTO 09 1
+ ST+ 00 12 + STO 04
GTO 00
```

08: Input from card

Data in 11 and 10 temporarily stored in 04-05

Name read into 10 stored by 'SN'

Test on MAXDEV read from card

Automatic repeat if series read from card is shorter than MAXDEV

```
154♦LBL 01
RCL 11 STO 04 RCL 10
STO 05 RCL 03 XEQ "CN"
ASTO 10 1 E3 STO 11
10.012 RCL 00 FRC
ST* 11 + WDTAX RCL 04
STO 11 RCL 05 STO 10
RTN
```

01: Storage on card of last variable
MAXDEV temporarily in 11 and name in 10 for storage

```
175♦LBL 09
1 ST- 03 "WRONG DATA"
PROMPT GTO "R"
```

09: MAXDEV read from card is more than MAXDEV of current data set
NV is updated back

```
181♦LBL 10
"K=?" PROMPT STO 04
RCL 00 FRC 12 1 E3 /
+ 12 + STO 05
```

10: Repeat factor stored in 04

Counter set in 05

```
194♦LBL 11
RCL IND 05 RCL 04 *
STO IND 05 ISG 05
GTO 11 RTN
```

11: Loop to multiply variable values by k

```
262♦LBL "STB"
"BL" RCL 09 2 -
ASTO IND X 1 + RCL 00
FRC 1 E3 * STO IND Y
RCL 09 + 1 E3 / + 1
+ WDTAX GTO "R" END
```

STB: Storage of balance
Address of year Ø stored in 09
Name 'BL' stored in (year Ø address) - 2
MAXDEV stored in (year Ø address) - 1

4. Card RPV 4 (SW)

Present Values/Switching % Module

PRP "SW"

01♦LBL "SW"	Test if at least one variable was input
02 XEQ "T"	
03 1	
04 RCL 09	
05 +	
06 STO 04	Address of year 1 balance
07 6	Address of balance present value
08 STO 07	Calculates balance PV
09 XEQ "PV"	
10 RCL 02	
11 CF 06	
12 X=0?	
13 SF 06	Flag set if OCC = 0
14 FS? 06	if OCC ≠ 0 :
15 GTO 03	
16 1	Without project PV =
17 +	
18 RCL 11	$-\dfrac{1-(1+OCC/100)\uparrow(-PLIFE)}{OCC/100}$ x (yr 0 balance)
19 CHS	
20 Y↑X	if OCC = 0 :
21 CHS	
22 1	
23 +	Without project PV = – P LIFE x (yr Ø balance)
24 RCL 02	
25 /	
26♦LBL 03	
27 FS? 06	
28 RCL 11	P LIFE
29 RCL IND 09	Yr Ø balance
30 *	
31 CHS	
32 STO 05	Without project PV stored in 05
33 ST+ 06	Incremental balance PV stored in 06
34 RCL 03	
35 1 E3	
36 /	
37 1	
38 +	
39 STO 08	Counter for number of variables
40 RCL 10	
41 1	
42 +	
43 STO 07	Address of variable 1 PV

4. Card RPV 4 (SW) (contd.)

```
 44♦LBL 12
XEQ 13  XEQ 05
RCL IND 07  XEQ 07  1
ST+ 07  ADV  ISG 08
GTO 12  "WITHOUT P"
AVIEW  XEQ 05  RCL 05
XEQ 07  ADV  "INCR BAL"
AVIEW  XEQ 05  RCL 06
XEQ "V"  ADV  "OCC %="
RCL 02  100  *  XEQ "V"
ADV  GTO "R"
```

12: Loop to List PVs and Switching Percentages

Recall PV
Increment PV address

```
 73♦LBL 07
XEQ "V"  X=0?  RTN
RCL 06  CHS  1 E2  *
X<>Y  /  XEQ 06  XEQ "V"
RTN
```

07: List PV. No switch % if PV = 0

$$\text{Switch \%} = \frac{-\text{ Balance PV x 100}}{\text{PV}}$$

List switch %

```
 86♦LBL 13
RCL 08  INT  XEQ "CN"
"F-VAR"  RCL 08  INT
XEQ "N"  AVIEW  RTN
```

13: Generates variable name and corresponding variable number

```
 96♦LBL 05
"PV="  RTN
```

```
 99♦LBL 06
"SWITCH %="  RTN
```

```
102♦LBL "CN"
CLA  RCL 10  +  RCL 01
+  1  +  ARCL IND X  END
```

CN: Call name - Name corresponding to variable number in X is called into ALPHA register.

Sets display with variable number

5. RPV 5 (BL)

Printout Module

```
   PRP "BL"

01♦LBL "BL"
02 XEQ "T"        Set flags for balance ---> listing
03 SF 03
04 CF 02
05 XEQ "PR"
06 GTO "R"

07♦LBL "IBL"
08 XEQ "T"        Set flags for incremental balance ---> listing
09 SF 03
10 SF 02
11 XEQ "PR"
12 GTO "R"

13♦LBL "PR"
14 RCL 03
15 XEQ "CN"       Routine to list time series of values :
16 "⊦-VAR"
17 RCL 03           either balance if flag 03 set
18 XEQ "N"
19 FS? 03           or last variable if flag 03 clear
20 "BALANCE"
21 FS? 02         Total values (including year Ø) if flag 02 clear
22 "⊦-INCR"
23 AVIEW          Incremental values (from year 1) if flag 02 set
24 RCL 09
25 FC? 03
26 12
27 STO 07
28 STO 04         Year Ø address in 07
29 1
30 FS? 02
31 ST+ 04         Year 1 address in 04 if incremental values to be listed
32 XEQ "K"        Year counter to Ø
33 1
34 FS? 02
35 ST+ 00         Year counter to 1 if incremental values to be listed

36♦LBL 01
37 "YR"
38 RCL 00
39 INT
40 XEQ "N"
41 "⊦="           Loop to list values
42 RCL IND 07
43 CHS
44 RCL IND 04
45 FS? 02
46 +
47 XEQ "V"
48 1
49 ST+ 04
50 ISG 00
51 GTO 01
52 ADV
53 RTN
```

5. RPV 5 (BL) (contd.)

Editing Module

```
♦LBL "EDT"
XEQ "T"
"VAR NBR?"
RCL 03
PROMPT
X>Y?
GTO "R"
X<=0?
GTO "R"           Check that VAR NBR being input is within the proper range
STO 05
XEQ "CN"          VAR NBR stored in 05
AON
PROMPT            Variable name is displayed and can be changed
AOFF
ASTO IND X
XEQ "YR"
STO 04            Year to be edited - stored in 04
RCL 09
+
STO 07            Address of year balance to be edited is stored in 07
"F CHGE?"
PROMPT
ST+ IND 07        Updates balance
RCL 05            VAR NBR back in x
RCL 10
+
STO 07            Address for PV change
RDN               Change back in x
RCL 02
1                 1 + OCC
+
RCL 04            Year number
X=0?              If year is 0, no adjustment on PV
GTO "R"
Y↑X               (1+ OCC)    year
/
ST+ IND 07        PV is adjusted
GTO "R"
END
```


6. RPV 6 (RE)

Rate of Return Calculation
(Illinois algorithm)

```
♦LBL "RE"
XEQ "T"
RCL 02
STO 20          OCC saved in 20
-1
XEQ 01          Balance transformed into incremental balance
"GUESS1?"
PROMPT          Store guesses
STO 13
"GUESS2?"           R1 in 13
PROMPT
STO 14              R2 in 14
1 E2
ST/ 13
ST/ 14
RCL 13
STO 16
STO 02              R1 in 02
RCL 12
STO 04          Address of year 1
17
STO 07          PV with R1 to be stored in 17
XEQ "PV"
RCL 17
RCL 14
STO 16
STO 02              R2 in 02
RCL 12
STO 04          Address of year 1
18
STO 07          Address to store PV - PV with R2 to be stored in 18
XEQ "PV"
RCL 18
RCL 17
*
X>0?
GTO 03          If PV's for R1 and R2 of same sign, no root exists
                  within the interval
♦LBL 02         Begin loop
RCL 14
RCL 14
RCL 13
-
RCL 18
RCL 17
-
/
RCL 18
*
-
STO 16          New rate R
STO 02          R in 02
RCL 12          Address of year 1
STO 04
19              PV with R to be stored in 19
STO 07
XEQ "PV"        If PV = 0 then done
RCL 19
X=0?
GTO 04
```

6. RPV (RE) (contdt.)

```
 60 ABS                   Tolerance value if PV < .1 then done
 61 .1
 62 X>Y?
 63 GTO 04
 64 RCL 19                Select new guesses
 65 RCL 18
 66 *
 67 X>0?
 68 GTO 06
 69 RCL 14
 70 STO 13
 71 RCL 18
 72 STO 17

 73♦LBL 00
 74 RCL 16
 75 STO 14
 76 RCL 19
 77 STO 18
 78 GTO 02

 79♦LBL 06
 80 2
 81 ST/ 17
 82 GTO 00

 83♦LBL 04                Done display result
 84 XEQ 05
 85 TONE 8
 86 TONE 6
 87 "IRR="
 88 RCL 16
 89 1 E2
 90 *
 91 ARCL X
 92 "├%"
 93 AVIEW
 94 ADV
 95 GTO "R"

 96♦LBL 03
 97 XEQ 05
 98 GTO "R"

 99♦LBL 05
100 RCL 20                Store back OCC in 02
101 STO 02
102 1
103 XEQ 01                Incremental balance transformed into balance
104 RTN
```

6. <u>RPV (RE)</u> (contd.)

```
105•LBL 01
106 RCL IND 09
107 *            Year Ø balance is multiplied by value in x (1 or -1).
108 STO 06         and then added to all years of balance
109 XEQ "K"
110 1
111 ST+ 00
112 RCL 09
113 1
114 +
115 STO 04
116 STO 12

117•LBL 14
118 RCL 06
119 ST+ IND 04
120 1
121 ST+ 04
122 ISG 00
123 GTO 14
124 END
```

7. RPV 7 (O)

Various Routines

```
01♦LBL "O"
02 "OCC %"
03 -100
04 PROMPT
05 X=Y?
06 GTO "O"
07 1 E2
08 /
09 STO 02
10 1 E3
11 ST/ 00
12 GTO "R"
```

O = Input of OCC - Value -100 illegal
OCC divided by 100 and stored in 02
MAXDEV in 00 divided by 1000 for counter

```
13♦LBL "YR"
14 "YR?"
15 PROMPT
16 RCL 00
17 FRC
18 1 E3
19 *
20 X<Y?
21 GTO "YR"
22 RDN
23 X<0?
24 GTO "YR"
25 "YR"
26 XEQ "N"
27 RTN
```

YR = Ask to input year number
Test if year ≥ 0 and ≤ MAXDEV

Year number put in ALPHA

```
28♦LBL "K"
29 RCL 00
30 FRC
31 STO 00
32 RTN
```

K = Put integer part of register 00 to zero
(year counter)

```
33♦LBL "Y"
34 AON
35 PROMPT
36 AOFF
37 ASTO Y
38 "N"
39 ASTO X
40 SF 01
41 X=Y?
42 CF 01
43 RTN
```

Y = Wait for ALPHA input (yes or no)
Flag 01 set if input is not N (i.e. yes)
Flag 01 clear if input is N

```
44♦LBL "N"
45 FIX 0
46 CF 29
47 ARCL X
48 FIX 2
49 SF 29
50 RTN
```

N = Value in X is attached to the content of ALPHA
register, with no decimal digit and no decimal point.

```
51♦LBL "T"
52 SF 21
53 RCL 03
54 X=0?
55 GTO "R"
56 RTN
```

T = Set flag 21 for printer control
Test if variables already input

7. RPV 7 (0) (contd.)

```
•LBL "R"
"READY"
ASTO X
AVIEW
STOP
```

R = Displays READY and STOP

```
•LBL "V"
SF 21
ARCL X
AVIEW
RTN
```

V = Set flag for printer control
Appends content of X to ALPHA and displays ALPHA

```
•LBL "PV"
0
STO IND 07
XEQ "K"
1
ST+ 00
STO 08
```

PV = Routine to calculate present value of a flow from year 1 to P LIFE
Assumes address of year 1 stored in 04
PV to be stored at address stored in 07

```
•LBL 09
XEQ 11
```

Increment address for next year

```
1
ST+ 04
```

End of first loop (years 1 to MAXDEV)

```
ISG 00
GTO 09
1
ST- 04
RCL 00
FRC
1 E3
*
RCL 11
X=Y?
```

If MAXDEV = P LIFE then done

```
RTN
1 E3
/
+
1
+
STO 05
```

Counter from MAXDEV + 1 to P LIFE in 05

```
•LBL 10
XEQ 11
```

Second loop (years MAXDEV + 1 to P LIFE)

```
ISG 05
GTO 10
RTN

•LBL 11
RCL 02
1
+
ST* 08
```

Register 08 used to store $(1 + \text{Rate})^t$

```
RCL IND 04
RCL 08
/
ST+ IND 07
```

Value of year t divided by $(1 + \text{Rate})^t$ and added to PV

```
END
```

Storage Plan

00 : MAXDEV/1000 – Used as year counter

01 : MAXNV – 02: OCC – 03: NV

04-08 : Temporary storage

09 : Address of year zero balance: MAX (8, MAXDEV) + 13

10 : (Address of first Present value) – 1: MAX (8, MAXDEV) + MAXDEV + 13

11 : P LIFE

12 : Year zero of current variable (being input)

MAX (8, MAXDEV) + 13 : Year zero balance

MAX (8, MAXDEV) + MAXDEV + 13 : Year MAXDEV balance

MAX (8, MAXDEV) + MAXDEV + 14)
)
) Present values of variables 1 to MAXNV
)
id. + MAXNV)

MAX (8, MAXDEV) + MAXDEV + MAXNV + 15)
)
) Names of variables 1 to MAXNV
)
id. + MAXNV – 1)

Flags Used

24,27 : Ignore overflow – Set user mode
25 : Test on size
21 : Printer control
29 : Control of display
01 : Yes/no
02-03 : Control of printout (balance/variable – total/incremental)

TS - TIME SERIES GENERATOR

A PROGRAMME FOR HP 41 C/CV CALCULATOR

I. Purpose of the Programme

The programme allows to generate a so-called "variable", display or print it, store it on magnetic card.

A variable is a yearly time series of values, representing for example a flow of costs or benefits in a project, or yields or area of a given crop. The series includes year zero which is used to represent the without project situation, when it is assumed to be constant.

Usually, during project life, variable values are changing over the initial period (called the development period) and then are constant. The last year of the time series generated by the programme is called MAXDEV (maximum development period), and its value, by definition, remains constant till the end of project life. This value is used in the programme to organize storage of data.

A variable is thus made of a name (1 to 6 characters) associated with values from year zero to MAXDEV.

The programme allows to create a variable using six different modes, as detailed later. If stored on card, the variable can be read by other programmes: RPV and PHASE.

II. Using the Programme

1. Configuration of the Calculator

The programmne occupies 1064 bytes (152 registers). The number of data registers required is MAXDEV + 11. A HP41C calculator with two memory modules can run the programme with MAXDEV equal to up to 28.

The programme can be loaded from card (it is stored on five cards). An annotated programme listing is given in Appendix 2.

The programme can be used with or without printer (see manuals of RPV or DEBT programmes).

2. Running the Programme

The programme is started either by XEQ'TS' or by pressing key X<->Y in user mode.

The following steps are explained below, and illustrated by seven sample runs presented in Appendix 1 (one for each of the six available options, and one showing input from card, editing, storage on card).

(1) Calls for MAXDEV ? i.e. maximum number of years when data will change. If required, size will have to be executed to allocate the necessary data registers.

(2) Then asks "CARD ?". If yes, asks to pass a card to read data and check on MAXDEV compatibility (if shorter on card, last year copied to the end, if longer on card "WRONG DATA" is displayed). Go to (6).

(3) Asks to input Name of data series.

(4) Asks to input YR 0 value.

(5) Asks to select OPTION ?

Six options, selected with upper keys:

A : Actual. Actual values to be input. If PI is input for a year, previous year value is repeated to the end of the series.

a : Cyclical. "NBR OF YRS" : input duration of cycle, then the programme prompts for values for the first cycle, and then copies to the rest of the years.

B : % (Percentage). Input of a base value, and then input of each year as a percentage of that value. Use of PI as repeat factor.

C : Linear. Linear interpolation between two given years; values set equal to year Ø before beginning year and equal to last year of change till the end of the series.

D : Rate. Between two given years, value of a year is equal to previous year value increased (or decreased) by a given constant rate.

E : S-Curve : Values between two given years are generated following a Sigmoid Curve pattern.

For C, D and E, the programme will call :

- CHANGING FROM YR ?

and TILL YR ?

For C and E, it will call for the value of the variable in the last year of change.

(6) When all values have been calculated, or after input from card, the series can be listed on display or printed (question : LIST ?). Then the question CHANGES ? is asked. If yes, give the year to be changed, the value of that year is displayed, and a new value can be input before pressing R/S again. The last question is STORE ?, that allows to record the time series on to a magnetic card.

SAMPLE RUNS

All runs shown below are with the printer in the position NORM.

Input of actual values
Use of PI for repeat

```
              XEQ "TS"
MAXDEV?
         12.0000    RUN
NONEXISTENT
              SIZE 020


              XEQ "TS"
MAXDEV?
         12.0000    RUN
XEQ SIZE 23
              SIZE 023
                    RUN
CARD?
N                   RUN
NAME?
INCOME              RUN
YR0=?
           10.00    RUN
OPTION?
                  XEQ A
YR1=?
         -150.00    RUN
YR2=?
           25.00    RUN
YR3=?
           32.00    RUN
YR4=?
                     PI
                    RUN
LIST?
                    RUN
INCOME
YR0=10.00
YR1=-150.00
YR2=25.00
YR3=32.00
YR4=32.00
YR5=32.00
YR6=32.00
YR7=32.00
YR8=32.00
YR9=32.00
YR10=32.00
YR11=32.00
YR12=32.00

CHANGES?
N                   RUN
STORE?
N                   RUN
END
```

Cyclical repeat

```
              XEQ "TS"
MAXDEV?
           12.00    RUN
CARD?
N                   RUN
NAME?
INCOME              RUN
YR0=?
           10.00    RUN
OPTION?
                  XEQ a
NBR OF YRS?
            5.00    RUN
YR1=?
         -150.00    RUN
YR2=?
           25.00    RUN
YR3=?
                     PI
                    RUN
LIST?
                    RUN
INCOME
YR0=10.00
YR1=-150.00
YR2=25.00
YR3=25.00
YR4=25.00
YR5=25.00
YR6=-150.00
YR7=25.00
YR8=25.00
YR9=25.00
YR10=25.00
YR11=-150.00
YR12=25.00

CHANGES?
N                   RUN
STORE?
N                   RUN
END
```

Input using percentages

```
                    XEQ "TS"
MAXDEV?
             12.00     RUN
CARD?
N                      RUN
NAME?
YIELD                  RUN
YR0=?
             10.00     RUN
OPTION?
                     XEQ B
BASE VALUE?
             50.00     RUN
YR1 %=?
             30.00     RUN
YR2 %=?
             45.00     RUN
YR3 %=?
             65.00     RUN
YR4 %=?
             90.00     RUN
YR5 %=?
            100.00     RUN
YR6 %=?
            120.00     RUN
YR7 %=?
                        PI
                       RUN
LIST?
                       RUN
YIELD
YR0=10.00
YR1=15.00
YR2=22.50
YR3=32.50
YR4=45.00
YR5=50.00
YR6=60.00
YR7=60.00
YR8=60.00
YR9=60.00
YR10=60.00
YR11=60.00
YR12=60.00

CHANGES?
N                      RUN
STORE?
N                      RUN
END
```

Linear interpolation

```
                    XEQ "TS"
MAXDEV?
             12.00     RUN
CARD?
N                      RUN
NAME?
COST                   RUN
YR0=?
             10.00     RUN
OPTION?
                     XEQ C
CHANGING FROM YR?
              3.00     RUN
TILL YR?
             10.00     RUN
YR10=?
             80.00     RUN
LIST?
                       RUN
COST
YR0=10.00
YR1=10.00
YR2=10.00
YR3=18.75
YR4=27.50
YR5=36.25
YR6=45.00
YR7=53.75
YR8=62.50
YR9=71.25
YR10=80.00
YR11=80.00
YR12=80.00

CHANGES?
N                      RUN
STORE?
N                      RUN
END
```

Input using compound rate

```
             XEQ "TS"
MAXDEV?
        12.00    RUN
CARD?
N                RUN
NAME?
VALUE            RUN
YR0=?
        10.00    RUN
OPTION?
               XEQ D
CHANGING FROM YR?
         3.00    RUN
TILL YR?
        10.00    RUN
RATE?
         5.00    RUN
LIST?
                 RUN
VALUE
YR0=10.00
YR1=10.00
YR2=10.00
YR3=10.50
YR4=11.03
YR5=11.58
YR6=12.16
YR7=12.76
YR8=13.40
YR9=14.07
YR10=14.77
YR11=14.77
YR12=14.77

CHANGES?
N                RUN
STORE?
N                RUN
END
```

Generating data following a S-curve

```
             XEQ "TS"
MAXDEV?
        12.00    RUN
CARD?
N                RUN
NAME?
NET VL           RUN
YR0=?
               XEQ E
CHANGING FROM YR?
         3.00    RUN
TILL YR?
        10.00    RUN
YR10=?
        80.00    RUN
LIST?
                 RUN
NET VL
YR0=0.00
YR1=0.00
YR2=0.00
YR3=1.96
YR4=8.06
YR5=21.04
YR6=40.00
YR7=58.96
YR8=71.94
YR9=78.04
YR10=80.00
YR11=80.00
YR12=80.00

CHANGES?
N                RUN
STORE?
N                RUN
END
```

Input from card, changes, storage on card

```
             XEQ "TS"
MAXDEV?
        10.00    RUN
CARD?
                 RUN
LIST?
                 RUN
```

Note: The message CARD displayed by the system is not printed. The same happens with RDY 01 OF 01 when storing on card.

```
WHEAT
YR0=10.00
YR1=10.25
YR2=10.92
YR3=12.32
YR4=14.59
YR5=17.50
YR6=20.41
YR7=22.68
YR8=24.08
YR9=24.75
YR10=25.00

CHANGES?
                 RUN
WHEAT
```

A new name can be specified

```
WHEAT2           RUN
 YR?
         5.00    RUN
YR5=17.50?
        19.20    RUN
LIST?
```

Key in the new value if there is a change

```
N                RUN
CHANGES?
                 RUN
WHEAT2
                 RUN
 YR?
         3.00    RUN
YR3=12.32?
```

Just press R/S key if no change (for name as well as for year 3 value)

```
                 RUN
LIST?
                 RUN
WHEAT2
YR0=10.00
YR1=10.25
YR2=10.92
YR3=12.32
YR4=14.59
YR5=19.20
YR6=20.41
YR7=22.68
YR8=24.08
YR9=24.75
YR10=25.00

CHANGES?
N                RUN
STORE?
                 RUN
END
```

RDY 01 OF 01 is displayed
Data is stored on card

TIME SERIES PROGRAMME

```
 01♦LBL "TS"
SF 24  SF 27  CF 05
CF 06  CLRG  "MAXDEV?"
PROMPT  STO 09  STO 07
10  +
```

User mode - Ignore overflow

MAXDEV in 09 (and temp. in 07)

```
 13♦LBL 01
SF 25  RCL IND X
FS?C 25  GTO 22
"XEQ SIZE "  1  +
XEQ "N"  PROMPT  1  -
GTO 01
```

TEST ON size

```
 26♦LBL 22
"CARD?"  XEQ "Y"
FC?C 01  GTO 03  RCL 09
10  +  1 E3  /  8  +
RDTAX  RCL 07  RCL 09
X=Y?  GTO "L"  X>Y?
GTO 04  10  +  STO 01  9
-  STO 00  RCL 07
STO 09  1 E3  /  ST+ 00
XEQ "R"  GTO "L".
```

Input from card
Value of MAXDEV from card compared with value input at the beginning
Copy last value to the end of the series if number of years on card is smaller
Otherwise ⟶ error

```
 58♦LBL 04
"WRONG DATA"  PROMPT
GTO 22
```

```
 62♦LBL 03
"NAME?"  AON  PROMPT
AOFF  ASTO 08  "YR0=?"
PROMPT  STO 10
"OPTION?"  PROMPT  STOP
```

Input of name, stored in 08
Input of year zero value ⟶ stored in 09

```
 74♦LBL B
"BASE VALUE?"  PROMPT
1 E2  /  STO 06  SF 02
GTO b
```

B : input as % of a base value
Base value divided by 100 stored in 06
flag 02 set

```
 82♦LBL A
CF 02
```

A : input of actual values
flag 02 clear

```
 84♦LBL b
RCL 09  XEQ 09  XEQ 10
GTO "L"
```

```
 89♦LBL a
CF 02  CF 04
"NBR OF YRS?"  RCL 09
PROMPT  STO 03  X>Y?
GTO a  X=Y?  SF 03  /
INT  STO 04  RCL 03  *
CHS  RCL 09  +  STO 05
X=0?  SF 04  1  ST- 04
RCL 03  XEQ 09  XEQ 10
FS?C 03  GTO "L"
FS?C 04  GTO 06  RCL 05
XEQ 09  RCL 04  1  +
RCL 03  *  11  +  STO 05
XEQ 08
```

a : input of actual values over a years (from year 1 to a) that are then repeated to the end
a = number of years in cycle, in 03
(test that a ≤ MAXDEV)
N = number of full repeat cycles, in 04
n = number of years in the last "short" cycle, in 05
flag 04 set if n = 0
flag 03 set if a = MAXDEV

```
131•LBL 06
RCL 03  XEQ 09  RCL 04
RCL 03  *  11  +  STO 05
XEQ 08  DSE 04  GTO 06
GTO "L"
```

a * N + 10 = address of 1st year of current cycle, stored in 05

```
144•LBL 08
RCL IND 01  STO IND 05
1  ST+ 01  ST+ 05
ISG 00  GTO 08  RTN
```

loop to copy one cycle
year counter in 00

```
153•LBL 09
1 E3  /  1  +  STO 00
11  STO 01  RTN
```

sub routine to set year counter in 00 and address in 01

```
162•LBL "L"
"LIST?"  XEQ "Y"
FS?C 01  XEQ 09
"CHANGES?"  XEQ "Y"
FS?C 01  GTO 07
"STORE?"  XEQ "Y"
FS?C 01  XEQ 12  "END"
PROMPT  STOP
```

L : after input, prompts for listing/printing data, changes, storage, end

```
178•LBL 09
SF 21  RCL 09  1 E3  /
STO 00  10  STO 01  CLA
ARCL 08  AVIEW
```

<u>listing</u>
printer controlled by flag 21
listing of values : year counter in 00
address of values in 01

```
189•LBL 11
"YR"  RCL 00  INT
XEQ "N"  "⊦="
RCL IND 01  XEQ "V"  1
ST+ 01  ISG 00  GTO 11
ADV  RTN
```

loop to list values

```
203•LBL 07
CLA  ARCL 08  AON
PROMPT  AOFF  ASTO 08
CLA  XEQ "YR"  "YR"
XEQ "N"  10  +  STO 01
"⊦="  RCL IND 01  ARCL X
"⊦?"  PROMPT  STO IND 01
GTO "L"
```

<u>changes</u> - Name displayed and can be changed
year to be changed is input : test $0 \leq yr \leq$ MAXDEV
old value displayed with "?"

```
224•LBL 12
RCL 09  10  +  1 E3  /
8  +  WDTAX  RTN
```

<u>store</u> Name (in 08)
MAXDEV (in 09) and time series are recorded on card

```
234•LBL "Y"
AON  PROMPT  AOFF
ASTO Y  "N"  ASTO X
SF 01  X=Y?  CF 01  RTN
```

<u>Y</u> : waits for ALPHA input
flag 01 set if input is <u>not</u> N (i.e. if yes)
flag 01 clear if input is N

```
245•LBL "N"
FIX 0  CF 29  ARCL X
FIX 2  SF 29  RTN
```

<u>N</u> : value in X is attached to the content of ALPHA register, with no decimal digit and no decimal point

```
252•LBL "V"
SF 21  ARCL X  AVIEW
RTN
```

<u>V</u> : printer controlled by flag 21
appends content of X to ALPHA and displays ALPHA register

```
257•LBL "R"
RCL 01  1  +  STO 06
```

<u>Repeat loop</u> copies value in IND 01 following counter set in 00, to addresses starting in IND 01 + 1 (stored in 06)

```
262♦LBL 05
RCL IND 01  STO IND 06
1  ST+ 06  ISG 00
GTO 05  RTN
```

Repeat loop

```
270♦LBL 10
"YR"  RCL 00  INT
XEQ "N"  FS? 02  "⊢ %"
"⊢=?"  PI  PROMPT  X=Y?
GTO 09  RCL 06  X<>Y
FS? 02  *  STO IND 01  1
ST+ 01  ISG 00  GTO 10
RTN
```

<u>Input of a series of values</u>
- as value or as % depending on flag 02
- test against PI for automatic repeat
- yr counter in 00
- address to store values in 01
- values multiplied by content of 06 before storage if flag 02 set (input of percentages)

```
292♦LBL 09
1  ST- 01  XEQ "R"  RTN
```

Repeat if PI was input

```
297♦LBL "YR"
"⊢ YR?"  RCL 09  PROMPT
X>Y?  GTO "YR"  X<0?
GTO "YR"  RTN
```

<u>YR</u> : appends "YR?" to ALPHA, waits for input, tests input year number against Ø and MAXDEV

```
306♦LBL C
SF 05  GTO E
```

<u>C</u> : flag 05 set if <u>LINEAR</u> interpolation

```
309♦LBL D
SF 06  GTO E
```

<u>D</u> : flag 06 set if compound <u>RATE</u> is used

```
312♦LBL E
XEQ 09  FC? 06  XEQ 12
FS? 06  XEQ 04  FS? 01
GTO 03  RCL 09  1 E3  /
RCL 02  1  +  +  STO 00
RCL 02  10  +  STO 01
XEQ "R"
```

<u>E</u> : used for LINEAR, RATE, or <u>S-CURVE</u> (if S-CURVE, flags 05 and 06 are cleared)
XEQ 09 : input of By and Ey, loop 1 ⟶ By-1
XEQ 12 : input value year Ey, except if <u>Rate</u>
XEQ 04 : input Rate and calculates years By to Ey if <u>Rate</u>
loop to copy V(Ey) into years Ey+1 to MAXDEV if flag 01 is clear (yr counter in 00, address in 01)

```
333♦LBL 03
FS? 06  GTO "L"  FS? 05
GTO 07  RCL 07  1  -
XEQ "SC"  STO 03  RCL 02
XEQ "SC"  STO 04  RCL 02
10  +  RCL IND X  STO 05
RCL 03  *  CHS  STO 06
RCL 07  9  +  RCL IND X
ST- 05  RCL 04  *
ST+ 06  RCL 04  RCL 03
-  ST/ 05  ST/ 06
RCL 05  STO 03  RCL 06
STO 04  XEQ "K"
```

if <u>Rate</u>, already complete ⟶ listing
if <u>linear</u> ⟶ go to 07
<u>S-curve</u> : By-1 in X ⟶ XEQ"SC" gives yØ in 03
Ey in X ⟶ XEQ"SC" gives ym in 04

$$A = \frac{\text{value } (Ey) - V(By-1)}{ym - y0}$$ calculated in 05, then stored in 03

$$B = \frac{ym \times V(By-1) - y0 \times (Vy)}{ym - y0}$$ calculated in 06, then stored in 04

```
373♦LBL 02
RCL 00  INT  XEQ "SC"
RCL 03  *  RCL 04  +
STO IND 01  1  ST+ 01
ISG 00  GTO 02  GTO "L"
```

loop to calculate yearly values
YJ = value returned by "SC"
value year J = YJ x A + B A in 03
B in 04

```
387♦LBL 07
RCL 02  10  +  RCL IND X
RCL 10  -  RCL 02
RCL 07  -  1  +  /
STO 03  XEQ "K"  XEQ 14
GTO "L"
```

<u>Linear interpolation calculation</u>
step = (V(Ey) - V(By-1))/(Ey - By + 1) in 03

```
404♦LBL "K"
-1 RCL 02 FC? 06 +
1 E3 / RCL 07 +
STO 00 RCL 07 10 +
STO 01 RTN
```

K : sets counter in 00, to Ey – 1 if linear, Ey if rate

Address pf V (Ey) in 01

```
419♦LBL 14
RCL 01 1 - RCL IND X
RCL 03 FS? 05 +
FS? 06 * STO IND 01 1
ST+ 01 ISG 00 GTO 14
RTN
```

Loop for linear/rate
for year J, V (J–1) recalled
if linear, step stored in 03 is added
if rate, 1 + Rate/100 stores in 03 multiplied
Result in IND 01

```
435♦LBL 09
"CHANGING FROM"
XEQ "YR" X=0? XEQ "YR"
STO 07 1 X=Y? GTO 09
- 1 E3 / 1 + STO 00
10 STO 01 XEQ "R"
```

Input of first year of change (Ey)
test 0 By MAXDEV By in 07
if By ≠ 1, counter 1 to By–1 in 00 and address of V (∅) in 01 ⟶ repeat value into initial years

```
453♦LBL 09
"TILL" XEQ "YR" RCL 07
X<>Y X<=Y? XEQ "YR"
RCL 09 X=Y? SF 01
X<>Y STO 02 RTN
```

Input of last year of change (Ey)
test Ey > By Ey stored in 02
flag 01 set is Ey = MAXDEV

```
466♦LBL 12
RCL 02 "YR" XEQ "N"
10 + STO 01 "⊢=?"
PROMPT STO IND 01 RTN
```

Input of value in year Ey, stored in IND 01 (for linear and S-curve)

```
477♦LBL 04
"RATE?" PROMPT 1 E2 /
1 + STO 03 XEQ "K"
XEQ 14 RTN
```

Input of Rate, $1 + \frac{\text{Rate}}{100}$ in 03

```
488♦LBL "SC"
2 * RCL 07 - 1 +
RCL 02 - RCL 02
RCL 07 - 1 + / 2.5
* STO 05 CF 02 X<0?
SF 02 ABS .231642 *
1 + 1/X STO 06
-1.33027 * 1.82126 +
RCL 06 * CHS 1.78148
+ RCL 06 * CHS
.356564 + RCL 06 *
CHS .319382 + RCL 06
* RCL 05 ENTER↑ * 2
/ CHS E↑X PI 2 *
SQRT / * FS? 02 RTN
CHS 1 + RTN .END.
```

SC :
J = year number, in X
X = 2.5 (2J–By+1–Ey) / (Ey–By+1) ⟶ 05
Z = 1/(1+.231642*ABS(X)) ⟶ in 06
flag 02 set if X < 0

V = .319382–Zx(.356564–Zx(1.78148–Zx(1.82126–1.33027Z)))

$Y = V \times Z \times EXP(-X^2/2) / \sqrt{2PI}$

if X ≥ 0 THEN Y = 1 – Y

Storage plan

00	Year counter
01	used to store address of value in a given year
02	Ey (last year of change)
03 to 06	used temporarily for intermediate results
07	By (first year of change)
08	Name
09	MAXDEV
10	V (Ø) = value of year Ø in the series
10 + MAXDEV	V (MAXDEV)

SIZE = MAXDEV + 11

minimum size before start = 10

(SIZE assigned to RCL)

Start : key 06 (X Y) in user mode

Flags 1 to 6 used

Labels and programme size

```
                CAT 1
LBL'TS
LBL'L
LBL'Y
LBL'N
LBL'V
LBL'R
LBL'YR
LBL'K
LBL'SC
.END.      1064 BYTES
```

DEBT - Debt Service Calculation

A Programme for HP 41 C Calculator

I. Purpose of the Programme

The programme allows to input a series of values representing a loan disbursed over several years, to specify the various characteristics of the loan, and to calculate total debt service, interest component of debt service, and outstanding capital, on a yearly basis. Maximum disbursement period is years 1 to 12. Debt service will not be calculated beyond year 20.

Loan instalments are assumed to be disbursed at the beginning of years, and repayment to occur at the end of years.

Grace period is to be specified. It is counted from the year of first disbursement, and must be such that first repayment is at the earliest at the end of the year of last disbursement. For example, if a loan is disbursed in three yearly instalments, in years 2, 3 and 4, grace period must be at least equal to two years:

grace = zero would correspond to first repayment end of year 2
grace = one " " " " " 3

In addition to grace period, the programme will call for interest rate and number of instalments.

In addition, several options are possible:

- Interest can be paid during grace period or can be capitalized.
- Repayment can be of two types:
  - equal instalments (constant total amount), or
  - equal capital instalments (variable total amount).

II. Using the Programme

1. Configuration of the Calculator

The programme occupies 1,338 bytes (192 registers) and requires 95 data registers, i.e. a total of 287 registers, thus requiring the maximum memory capacity of the calculator (preferably an HP41CV or HP41C with an additional quad-memory module. An HP41C with four simple memory modules would not allow to use the card reader and/or the printer).

The programme can be loaded from cards (it is stored on 6 cards). An annotated programme listing is presented in Appendix 2.

The programme can be used with or without printer. Data will be printed automatically when the proper function is selected and the printer is connected and switched on. If the printer is not present, or if it is connected but switched off, the programme will stop every time a data is displayed - to give time to write down the information if required - and will continue when R/S key is pressed.

2. Storing data on card

If the card reader is connected, loan amounts can be input from a card, or can be stored on card after input.

After running the calculations, the results can be stored on card: debt service, interest component, outstanding.

Data corresponding to loan amounts is specific to the DEBT programme. Data stored as a result of calculations can be read by the programmes RPV and PHASE.

3. Naming Conventions

The series of values corresponding to a loan is to be given a name by the user. The name can be made of 1 to 6 characters.

The programme will create names for the three flows of results by appending the first five letters of loan name to the letter D, I and O for debt service, interest component and outstanding capital respectively.

4. Running the Programme

See sample run in Appendix 1.

In user mode, the programme is started by pressing the X<->Y key, or by XEQ'DEBT'.

The programme will call all necessary data, run the calculations, and allow to list all the results. READY is displayed at the end of the run, meaning that the programme is waiting for data editing and new calculations.

The programme will call for loan name, year of first and last loan instalment, and loan values for the corresponding years. The data can then be listed, changed, stored on card.

If yes is answered to the question 'change ?' (by typing Y R/S or only R/S), the name is displayed first. A new name can be typed, or the old name can be kept by just pressing R/S. Then the year to be corrected is called for, the old value is displayed, and it is possible either to keep it by pressing R/S or to type a new value before pressing R/S.

When the input of loan values is complete, the programme will call for the rest of the data, run the calculations, and ask questions to allow for printing and/or storage on card of the results.

After the initial run, it is possible to modify some of the data and to run again the calculations.

Each data is corrected by pressing the corresponding key, and the message READY comes back after each operation. New calculations are performed by pressing key E.

The following functions can be performed in any order:

Key	Function
A	New input of loan amounts
a (shift A)	Listing, changing, storing loan amounts (same as after full input)
B	Display of interest rate, that can be changed or not
C	Display/editing of grace period
c (shift C)	Display/editing of number of instalments
D	Specifying if interest paid or not during grace period
d	Specifying equal instalment or not (i.e. equal capital instalment)
b (shift B)	Display/printout of loan characteristics
E	Calculations and results.

Debt - Sample Run (Printer in NORM mode)

I and II : Initial run.
I : Input of loan amounts and characteristics.
II : Output of debt service calculation, interest component, outstanding capital and loan characteristics.

```
              XEQ "DEBT"
LOAN-CARD?
N                    RUN
LOAN NAME?
-LG T                RUN
LOAN-FIRST YR?
             2.00    RUN
LOAN-LAST YR?
             4.00    RUN
LOAN-YR2?
           150.00    RUN
LOAN-YR3?
           250.00    RUN
LOAN-YR4?
           180.00    RUN
LOAN-LIST?
Y                    RUN
LOAN--LG T
YR2=150.00
YR3=250.00
YR4=180.00

CHANGE?
N                    RUN
STORE?
Y                    RUN
RATE=%?
            12.50    RUN
GRACE=?
             2.00    RUN
NBR OF INSTALMENTS=?
             5.00    RUN
GRACE=INT PAID?
Y                    RUN
EQUAL INST?
Y                    RUN
```

```
DEBT SERV-LIST?
                     RUN
DEBT SERV-D-LG T
YR2=18.75
YR3=50.00
YR4=162.90
YR5=162.90
YR6=162.90
YR7=162.90
YR8=162.90
YR9=0.00

STORE?
N                    RUN
INTEREST-LIST?
                     RUN
INTEREST-I-LG T
YR2=18.75
YR3=50.00
YR4=72.50
YR5=61.20
YR6=48.49
YR7=34.19
YR8=18.10
YR9=0.00

STORE?
N                    RUN
OUTSTAND-LIST?
                     RUN
OUTSTAND-O-LG T
YR2=150.00
YR3=400.00
YR4=580.00
YR5=489.60
YR6=387.91
YR7=273.50
YR8=144.80
YR9=0.00

STORE?
N                    RUN
LOAN CHAR?
                     RUN
RATE=12.50%
GRACE=2YRS-INT PAID
NBR OF INSTALMENTS=5
EQUAL INSTALMENTS
LOAN DISB BEG OF YR
REPAID END OF YR

                     RUN
READY
```

III : Changing loan characteristics and recalculating debt service. Note the test for grace period.

```
                    XEQ C
GRACE=2.00?
             1.00    RUN
INPUT GRACE>=2
GRACE=1.00?
             3.00    RUN
                    XEQ d
EQUAL INST?
N                    RUN
                    XEQ E
DEBT SERV-LIST?
                     RUN
DEBT SERV-D-LG T
YR2=18.75
YR3=50.00
YR4=72.50
YR5=188.50
YR6=174.00
YR7=159.50
YR8=145.00
YR9=130.50
YR10=0.00

STORE?
N                    RUN
INTEREST-LIST?
                     RUN
INTEREST-I-LG T
YR2=18.75
YR3=50.00
YR4=72.50
YR5=72.50
YR6=58.00
YR7=43.50
YR8=29.00
YR9=14.50
YR10=0.00

STORE?
N                    RUN
OUTSTAND-LIST?
N                    RUN
STORE?
N                    RUN
                     RUN
READY
```

After STORE? Y, the system displays RDY ~~WOFn~~ and waits for CARD. This is not printed.

IV : Run with printer in mode MAN
V : Lean characteristics printed on request by pressing b (shift B)

```
LOAN- LT1
YR2=165.20
YR3=254.30
YR4=95.40

DEBT SERV-D LT1
YR2=20.65
YR3=52.44
YR4=144.61
YR5=144.61
YR6=144.61
YR7=144.61
YR8=144.61
YR9=0.00

INTEREST-I LT1
YR2=20.65
YR3=52.44
YR4=64.36
YR5=54.33
YR6=43.05
YR7=30.35
YR8=16.07
YR9=0.00

OUTSTAND-O LT1
YR2=165.20
YR3=419.50
YR4=514.90
YR5=434.65
YR6=344.37
YR7=242.80
YR8=128.54
YR9=0.00
```

```
XEQ b
RATE=12.50%
GRACE=2YRS-INT PAID
NBR OF INSTALMENTS=5
EQUAL INSTALMENTS
LOAN DISB BEG OF YR
REPAID END OF YR

READY
```

Note: When a question is to be answered by YES or NO, the letter N will be understood as NO and any other answer (including Y, or pressing directly R/S) will be understood as YES.

DEBT - Annotated Programme Listing

```
PRP "DEBT"

01♦LBL "DEBT"
CF 04  CF 05  CF 06
CF 07  SF 24  SF 27
SF 02  CLRG
```

Flags used: Flag 00 set if equal instalments, clear if equal capital repayment. Flag 01 yes/no. Flag 02 set during initial input. Flag 03 set if interest paid during grace. Flags 04, 05, 06, 07 used to identify respectively loan amount, debt service, interest component and outstanding. Flags 08, 09, 10 used in edit and calculation. Flag 24: user mode. Flag 27: ignore overflow. Flag 21: printer

```
10♦LBL 20
SF 25  RCL 94  FS?C 25
GTO 21  "XEQ SIZE 095"
PROMPT  GTO 20
```

Test on SIZE using flag 25 (ignore error)

```
18♦LBL 21
XEQ A  XEQ B  XEQ C
XEQ c  XEQ D  XEQ d
XEQ E  "LOAN CHAR?"
XEQ "Y"  FS? 01  XEQ b
CF 02  GTO "R"
```

Initial run (with flag 02 set) calling for input of all data and calculations

```
32♦LBL A
SF 04  XEQ H  "⊢-CARD?"
XEQ "Y"  FC?C 01  GTO 22
10.025  RDTAX  GTO a
```

A: Input of loan values
First = Year of 1st loan instalment stored in 10
Last = Year of last loan instalment stored in 12
Loan amounts in 13 to 25 (including year Ø)

```
42♦LBL 22
XEQ H  "⊢ NAME?"  AON
PROMPT  AOFF  ASTO 11
XEQ H  "⊢-FIRST YR?"  12
PROMPT  X<=0?  GTO 22
X>Y?  GTO 22  STO 10
```

Test 0 < First $\leq$ 12

```
58♦LBL 23
XEQ H  "⊢-LAST YR?"
PROMPT  X<Y?  GTO 23  12
X<Y?  GTO 23  RDN
STO 12  1 E3  /  +
STO 00  RCL 10  13  +
STO 01
```

Test First $\leq$ last $\leq$ 12

Year counter in 00 for loan amounts input

```
77♦LBL 01
XEQ H  "⊢-YR"  RCL 00
INT  XEQ "N"  "⊢?"
PROMPT  STO IND 01  1
ST+ 01  ISG 00  GTO 01
```

Loop for input of loan amounts

Address to store amounts in 01

```
90♦LBL a
FS? 09  XEQ 26  CF 09
SF 04  12  STO 04
XEQ "L"  "CHANGE?"
XEQ "Y"  FS? 01  GTO 24
CF 04  "STORE?"  XEQ "Y"
FS? 01  XEQ 25  FS? 09
XEQ 26  GTO 40
```

a: Editing of loan amounts:
if initial input (Flag 02 set) then no test;
also grace period is tested if values of First or Last are changed

```
110♦LBL 24
ARCL 11  AON  PROMPT
AOFF  ASTO 11  XEQ H
"⊢-YR?"  12  PROMPT
X>Y?  GTO 24  X<=0?
GTO 24  RCL 10  X<>Y
X<Y?  STO 10  X<Y?
SF 09  RCL 12  X<>Y
X>Y?  STO 12  X>Y?
SF 09  STO 01  "YR"
XEQ "N"  "⊢="  13
ST+ 01  RCL IND 01
ARCL X  "⊢?"  PROMPT
STO IND 01  X≠0?  GTO a
RCL 01  13  -  RCL 10
CF 10  X=Y?  SF 10  X<>Y
1  FS? 10  ST+ 10  X<>Y
RCL 12  CF 10  X=Y?
SF 10  X<>Y  1  FS? 10
ST- 12  GTO a
```

Display loan name to allow for change
Edit loan value: test 0 < year to edit $\leq$ 12

First year of loan (in 10) and last year (in 12) are updated if required, and flag 09 set

Input of new value for year to edit. The prompt shows the old value

Values of First or Last updated if value of loan in one of these years is changed to zero

```
170♦LBL 25
10.025  WDTAX  RTN
```

Storage of loan values on card

```
174♦LBL 26
FS? 02  RTN  XEQ 27  RTN
```

Routine 27 called if <u>not</u> initial input

```
179♦LBL B
100  ST* 07  "RATE="
RCL 07  FC? 02  ARCL X
"⊢%?"  PROMPT  100  /
STO 07  GTO 40
```

<u>B</u>: Rate input as % and stored as Rate/100
Current value displayed if not initial input

```
192♦LBL C
XEQ G  RCL 08  FC? 02
ARCL X  "⊢?"  PROMPT
STO 08  XEQ 27  FS? 09
RTN  GTO 40
```

<u>C</u>: Input of grace period. Display current value if not initial input
Return to loan edit if flag 09 set

```
204♦LBL 27
RCL 08  RCL 12  RCL 10
-  X<=Y?  RTN
"INPUT GRACE>="  XEQ "N"
AVIEW  GTO C
```

Test that grace $\geq$ Last-First
if not, input of ~~graces~~ grace called after displaying the minimum value required

```
215♦LBL c
"NBR OF "  XEQ "S"  "⊢="
RCL 09  FC? 02  XEQ "N"
"⊢?"  PROMPT  STO 09
GTO 40
```

<u>c</u>: Input number of instalments
Display current value if not initial input

```
226♦LBL D
XEQ G  "⊢INT PAID?"
XEQ "Y"  CF 03  FS? 01
SF 03  GTO 40
```

<u>D</u>: Interest paid or not during grace period indicated by status of Flag 03

```
234♦LBL d
"EQUAL INST?"  XEQ "Y"
CF 00  FS? 01  SF 00
GTO 40
```

<u>d</u>: Equal instalments or equal capital repayment indicated by status of Flag 00

```
241♦LBL E
26.094  STO 00  CLX

245♦LBL 03
STO IND 00  ISG 00
GTO 03  CLA  "D"
ARCL 11  ASTO 26  CLA
```

E: Calculations of Debt service

```
"I"  ARCL 11  ASTO 72
CLA  "O"  ARCL 11
ASTO 49  RCL 10  RCL 08
+  RCL 09  +  1  -  20
X>Y?  RDN  STO 27
STO 50  STO 73  RCL 07
FS? 03  CLX  STO 01  51
```

. Register 26 to 94 initialized to zero. Names of debt service, interest, outstanding, are created using suffixes (D, I, O) and 5 first letters of loan name
. YL : last year of repayment (≤ 20) stored in 27, 50, 73
. K (in 01) : 0 if interest paid during grace, else K = Rate/100

```
RCL 10  +  RCL 10  13  +
RCL IND X  STO IND Z
RCL 07  *  FC? 03  0
RCL 10  74  +  X<>Y
STO IND Y  RCL 10  28  +
X<>Y  STO IND Y  20
```

Outstand (First) = Loan (First)
Interest (First) = Debt (First) = Loan (First) * (Rate/100 or Ø)

```
RCL 10  RCL 08  +  CF 10
X>Y?  SF 10  FS? 10  RDN
1 E3  /  RCL 10  +  1  +
STO 00  INT  13  +
STO 03  15  +  STO 04
23  +  STO 05  23  +
STO 06
```

Counter in 00 for 1st loop. I = First + 1 to YF or 20 (YF: Year of first repayment). Flag 10 set if YF > 20
address of loan (I) = 13 + I in 03
address of outset (I) = 51 + I in 05
address of debt (I) = 28 + I in 04
address of interest (I) = 74 + I in 06

```
329♦LBL 04
RCL 05  1  -  RCL IND X
RCL 01  1  +  *
STO IND 05  RCL 12
RCL 00  INT  CF 08
X<=Y?  SF 08  RCL IND 03
FS?C 08  ST+ IND 05
RCL IND 05  RCL 07  *
FC? 03  CLX  STO IND 06
STO IND 04  1  XEQ 30
```

First loop: Outst (I) = Outst (I - 1) * (K + 1)
If I ≤ last then Outst (I) = Outst (I) + Loan (I)
Int (I) = Outst (I) * Rate/100 if interest paid
Debt (I) = Int (I)

```
ISG 00  GTO 04  FS? 10
GTO 31  -1  XEQ 30
RCL IND 05  RCL 07  *
LASTX  1  +  RCL 09  CHS
Y↑X  CHS  1  +  /
STO 02  RCL 07  CF 08
X=0?  SF 08  FC? 00
SF 08  RCL IND 05
RCL 09  /  FS?C 08
STO 02  RCL 27  1 E3
RCL 10  +  RCL 08  +
STO 00  RCL 07  FS? 00
CLX  STO 01
```

End of 1st loop. End of calculations if Flag 10 set
Address in 03-04-05-06 put back to year YF
Calculation of PMT
if equal capital or if Rate = 0 then
PMT = Outst (YF)/Nbr of repayments
Else (with N = nbr of repayments)
PMT = Outst (YF) * Rate/100/(1 - (1 + Rate/100)^{N})
Counter for 2nd loop : I = YF to minimum (YL, 20) in 00
K = 0 if equal instalments, else k = Rate/100

```
400♦LBL 05
RCL IND 05  RCL 01  *
RCL 02  +  STO IND 04
RCL IND 05  RCL 07  *
STO IND 06  LASTX  1  +
FC? 00  1  RCL IND 05  *
RCL 02  -  RCL 05  1  +
X<>Y  STO IND Y  1
XEQ 30  ISG 00  GTO 05
CLX  STO IND 05  20
RCL 27  CF 08  X>Y?
SF 08  1  FS?C 08  CLX
ST+ 27  RCL 27  STO 50
STO 73  GTO 31
```

Second loop: Debt (I) = PMT + Outst (I) * k
Int (I) = Oust (I) * Rate/100
Outst (I + 1) = Outst (I) * X - PMT
with X = 1 if equal capital, X = 1 + Rate/100 if equal instalments
End 2nd loop
Outst (YL + 1) adjusted to zero (suppress rounding errors)
MAXDEV for the three series (Debt, Interest and Outstanding) is increased by one if YL < 20 for compatibility with other programmes

```
444♦LBL 30
ST+ 03  ST+ 04  ST+ 05
ST+ 06  RTN
```

LBL 30: Sub-routine to increment address counter (for debt service calculations)

```
450♦LBL 31
SF 05  27  STO 04
XEQ 28  CF 05  SF 06  73
STO 04  XEQ 28  CF 06
SF 07  50  STO 04
XEQ 28  CF 07
```

LBL 31: Flags 05, 06, 07 set successively to list/store debt service, interest, outstanding.MAXDEV address stored in 04

```
466♦LBL 40
FS? 02  RTN  GTO "R"
```

Return if initial input, if not display READY

```
470♦LBL 28
XEQ "L"  "STORE?"
XEQ "Y"  FC? 01  RTN
RCL 27  RCL 04  +  1  +
1 E3  /  RCL 04  +  1  -
WDTAX  RTN
```

Storage of results in a format compatible with other programmes (Name and MAXDEV followed by year values, including year zero)

```
489♦LBL "L"
XEQ H  "⊢-LIST?"
XEQ "Y"  FC? 01  RTN
RCL IND 04  1 E3  /
RCL 10  +  STO 00
RCL 04  1  +  RCL 10  +
STO 01  SF 21  XEQ H
"⊢-"  RCL 04  1  -
ARCL IND X  AVIEW
```

List: Label 'L'
Listing of values from year First to MAXDEV
MAXDEV previously stored in 04
Year counter in 00
Address of year values in 01
(from year First to Last if loan amounts)
02: Loop to list values

```
515♦LBL 02
"YR"  RCL 00  INT
XEQ "N"  "⊢="
RCL IND 01  ARCL X
AVIEW  1  ST+ 01  ISG 00
GTO 02  ADV  RTN
```

```
530♦LBL "R"
"READY"  ASTO X  STOP
```

R: READY at the end of each run/edit

```
534♦LBL "Y"
AON  PROMPT  AOFF
ASTO Y  "N"  ASTO X
SF 01  X=Y?  CF 01  CLA
RTN
```

Y: Flag 01 if set of input is not N (i.e. if yes)

```
546♦LBL "N"
FIX 0  CF 29  ARCL X
FIX 2  SF 29  RTN
```

N: Value in X attached to the content of ALPHA register with no decimal digit and no decimal point

```
553♦LBL H
"LOAN"  FS? 05
"DEBT SERV"  FS? 06
"INTEREST"  FS? 07
"OUTSTAND"  RTN
```

H: Proper label on ALPHA depending on flag set

```
562♦LBL b
SF 21  "RATE="  RCL 07
100  *  ARCL X  "⊦%"
AVIEW  XEQ G  RCL 08
XEQ "N"  "⊦YRS-INT "
FC? 03  "⊦NOT "  "⊦PAID"
AVIEW  "NBR OF "
XEQ "S"  "⊦="  RCL 09
XEQ "N"  AVIEW  "EQUAL "
FS? 00  XEQ "S"  FC? 00
"⊦CAPITAL "  FC? 00
"⊦REPAYMENT"  AVIEW
"LOAN DISB "
"⊦BEG OF YR"  AVIEW
"REPAID END"  "⊦ OF YR"
AVIEW  ADV  GTO 40

601♦LBL "S"
"⊦INSTALMENTS"  RTN

604♦LBL G
"GRACE="  RTN  END
```

b: Listing/display of loan characteristics

```
                CAT 1
LBL'DEBT
LBL'L
LBL'R
LBL'Y
LBL'N
LBL'S
END        1338 BYTES
.END.      06 BYTES
```

(192 registers)

Storage Plan

00	:	Year counter
01 to 06	:	Temporary storage for calculations
07	:	Interest rate (Rate in % divided by 100)
08	:	Grace period
09	:	Repayment period (Number of instalments)
10	:	First: year of first loan instalment
11	:	Loan name
12	:	Last: year of last instalment
13 to 25	:	Loan amounts (years 0 to 12)
26	:	Debt service - Name
27	:	MAXDEV (Last year of debt service, + 1 if <20)
28-48	:	Debt service (year s 0 to 20)
49	:	Outstanding - Name
50	:	id. 27
51-71	:	Outstanding capital (Years 0 to 20)
72	:	Interest - Name
73	:	id. 27
74-94	:	Interest component (Years 0 to 20)

Size = 095

Flags used : 0 to 19 - 21 - 24 - 25 - 27
Programme size : 1378 bytes (6 cards, i.e. 12 half cards)

PHASE - Phasing Calculations

A Programme for HP 41 C Calculator

I. Purpose of the Programme

1. Definitions

A variable is a yearly time series of values, including year zero, associated with a name (1 to 6 characters).

Values are changing over the initial period, till the year called MAXDEV (maximum development period), and then are constant till the end of project life (see also manuals of RPV and TS programmes).

A unit variable contains a series of values that represent a pattern to be aggregated. The values correspond to the entry of one unit of this variable in a project in year 1. For example, the values will represent the yield of 1 ha of a perennial crop planted in year 1 (then value in year zero should be zero), or the changes in the value of production from one farm if it enters into a project in year 1.

In the programme, the unit variable is called PROJ (for project). As the without project situation might not be constant, it needs to be specified in another variable, called NOPR (for no project).

MAXDEV is limited to 20 for both PROJ and NOPR.

The number of units entering into the project year after year are specified in another variable, the phasing variable. As the number of units must be specified in incremental terms (e.g. the number of new hectares irrigated in year 3, and not the total number of hectares irrigated that year), the phasing variable is called INCR NB (Incremental number). MAXDEV is limited to 12 years for INCR NB.

2. The Phasing Calculation

The purpose of the programme is to aggregate values stored in the unit variable using the number of units in the phasing variable. The calculation is performed a different way depending whether the results are to be in total or incremental terms (incremental means total minus the without project value). This will be better illustrated by two examples:

Year	0	1	2	3	4	5	6
Yield-Project	1.0	1.5	2.0	2.5	3.0 ---> yr 20		
Yield-NOPR	1.0	1.1	1.2	1.2	1.2 ---> yr 20		
Nbr of hectares (incremental)	-	10	20	20	0 ---> yr 20		
Total results							
has of year 1	10	15	20	25	30	30	30
has of year 2	20	22	30	40	50	60	60
has of year 3	20	22	24	30	40	50	60
Total	50	59	74	95	120	140	150
Incremental Results							
has of year 1	0	4	8	13	18	18	18
has of year 2	0	0	6	16	26	36	36
has of year 3	0	0	0	6	16	26	36
Total	0	4	14	35	60	80	90

For example, value year 3 for the 20 ha entering year 2 is :
yield year (3-2) = 2.0 minus yield NOPR year 3 = 1.2
multiplied by 20 equal 16.

The results will be calculated up to MAXDEV = 20 for results.

II. Using the Programme

1. Configuration of the Calculator

The programme occupies 1,081 bytes (155 registers), and the number of data registers required is 89. The programme can be loaded from cards (it is stored on 5 cards). An annotated programme listing is given in Appendix 2.

The programme can be used with or without printer.

2. Running the Programme

The programme is started either by XEQ'PHASE' or by pressing key X<->Y in user mode.

A sample run is presented in Appendix 1.

The programme will prompt the user for the input of three time series: PROJ, NOPR and INCR NB (see definitions above).

Time series can be input from card or directly. MAXDEV is requested first.

For direct input, name and yearly values are called for. It is then possible to list values, correct them (as well as the name) or read a new card, store the variable on card.

Data storage on card is compatible with TS and RPV programmes. It is thus possible to use as input a series of values prepared using the TS (time series) programme, or to use the results of a phasing calculation as an input for RPV (Rate of return/Present values calculations). After input of the three time series, the question TOT RESULT ? is displayed (if N, it will be incremental results), and the calculation is performed.

READY is displayed at the end of a run.

Using the upper keys, it is then possible to change some of the input data, in any order, and perform new calculations on the current data. The keys are:

A : input PROJ - a (shift A) : list/edit/store PROJ
B : input NOPR - b (shift B) : list/edit/store NOPR
C : input INCR NB - c (shift C) : list/edit/store INCR NB
D : total/incremental results - d : display/edit result name
E : calculations - listing/storage of results.

PHASE - SAMPLE RUN
(Printer in NORM mode)

1. Input of with project values from card - Listing - No change
   Direct input of without project values - Listing - No change
   Storage: the display RDY 01 OF 01 does not get printed.

```
                 XEQ "PHASE"
PROJ MAX DEV?
                 6.00     RUN
CARD?
                          RUN
LIST?
                          RUN
PROJ -WHEAT
YR0=10.00
YR1=15.00
YR2=20.00
YR3=35.00
YR4=30.00
YR5=35.00
YR6=40.00

CHANGE?
N                         RUN
STORE?
N                         RUN
NOPR MAX DEV?
                 3.00     RUN
CARD?
N                         RUN
NOPR NAME?
TR WH                     RUN
TR WH YR0 ?
                10.00     RUN
TR WH YR1 ?
                12.00     RUN
TR WH YR2 ?
                14.00     RUN
TR WH YR3 ?
                15.00     RUN
LIST?
                          RUN
NOPR -TR WH
YR0=10.00
YR1=12.00
YR2=14.00
YR3=15.00

CHANGE?
N                         RUN
STORE?
                          RUN
```

2. Direct input of phasing variable - Listing - Editing of a year with copy to end - new listing.

```
INCR NB MAX DEV?
               4.00    RUN
CARD?
N                      RUN
INCR NB NAME?
HA                     RUN
HA YR1 ?
              10.00    RUN
HA YR2 ?
              20.00    RUN
HA YR3 ?
              30.00    RUN
HA YR4 ?
               0.00    RUN
LIST?
N                      RUN
CHANGE?
                       RUN
CARD?
N                      RUN
HA
                       RUN
HA-YR?
               3.00    RUN
YR3=30.00?
               0.00    RUN
COPY TO END?
                       RUN
LIST?
                       RUN
INCR NB -HA
YR1=10.00
YR2=20.00
YR3=0.00

CHANGE?
N                      RUN
STORE?
N                      RUN
```

3. Calculation of total results - Listing - Keys D, d and E used to specify incremental results, change name and rerun calculations.

```
TOT RESULT ?
Y                    RUN
RESULT NAME?
TOT PR               RUN
LIST?
                     RUN
TOT RESULT-TOT PR
YR0=300.00
YR1=390.00
YR2=500.00
YR3=750.00
YR4=1,000.00
YR5=950.00
YR6=1,100.00
YR7=1,200.00
YR8=1,200.00

STORE?
N                    RUN
READY
                   XEQ D
TOT RESULT ?
N                    RUN
READY
                   XEQ d
TOT PR
INCR P               RUN
READY
                   XEQ E
LIST?
                     RUN
INCR RESULT-INCR P
YR0=0.00
YR1=30.00
YR2=80.00
YR3=300.00
YR4=550.00
YR5=500.00
YR6=650.00
YR7=750.00
YR8=750.00

STORE?
N                    RUN
READY
```

PHASE Programme
Annotated Listing

```
PRP "PHASE"

01♦LBL "PHASE"
CF 03  CF 04  CF 05
SF 24  SF 27  SF 02
CLRG
```

Flags used: 00: SET if total results - 01: for YES/No - 02: set during initial input - 03: project values - 04: without project values - 05: number of units - 06: set when MAXDEV = max life - 07: list of results - 08: used in edit - 24: user mode - 27: ignore overflow - 21: printer control - 25: ignore error.

```
09♦LBL 20
SF 25  RCL 88  FS?C 25
GTO 21  "XEQ SIZE 089"
PROMPT  GTO 20
```

20: test on SIZE

```
17♦LBL 21
XEQ A  XEQ B  XEQ C
XEQ D  "RESULT NAME?"
AON  PROMPT  AOFF
ASTO 66  XEQ E  CF 02
GTO "R"
```

21: Module calling successively for the input of all data and the calculation

```
30♦LBL A
SF 03  44  STO 04  45
STO 01  XEQ "INP"  RTN

38♦LBL B
SF 04  21  STO 04  22
STO 01  XEQ "INP"  RTN

46♦LBL C
SF 05  6  STO 04  7
STO 01  XEQ "INP"  RTN
```

A, B, C: New input of project values, without project values and number of units respectively
Address of MAXDEV stored in 04
Address of year zero value stored in 01

```
54♦LBL D
"TOT RESULT ?"  XEQ "Y"
CF 00  FS? 01  SF 00
FS? 02  RTN  GTO "R"
```

Flag 00 set if total results, clear if incremental

```
63♦LBL a
SF 03  44  STO 04
GTO "LST"

68♦LBL b
SF 04  21  STO 04
GTO "LST"

73♦LBL c
SF 05  6  STO 04
GTO "LST"
```

a, b, c: Listing/editing/storing of PROJ, NOPR and INCR NB values respectively

```
78♦LBL d
CLA  ARCL 66  AON
PROMPT  AOFF  ASTO 66
GTO "R"
```

d: Editing of result name

Calculation algorithm:

```
For each year J = 1 to phasing period do:
  If result is total then
     for each year N = 0 to J-1 do RES(N) = RES(N) + NOPR(N) * NB(J)
     for each year N = J to MAXDEV of results do
         if results are total: value = PROJ(N-J+1)
         if results are incremental: value = PROJ(N-J+1) - NOPR(N)
                                     RES(N) = RES(N) + Value * NB(J)
```

86♦LBL E	
RCL 21 RCL 44 X<Y?	MAXDEV of PROJ and NOPR in 44 and 21, of Phasing in 06
X<>Y RCL 06 + 1 -	
20 X>Y? X<>Y STO 67	MAXDEV of results stored in 67
.020 STO 00 68 STO 01	
GTO 08	
	Results stored in 68 to 88 initialized to zero
104♦LBL 08	
0 STO IND 01 1 ST+ 01	year counter in 00, addresses in 01
ISG 00 GTO 08 12	
1 E3 / 1 + STO 00	J counter for main loop set in 00
GTO 01	
118♦LBL 01	Start main loop
RCL 00 INT 7 +	NB(J) address in 02
STO 02 FC? 00 GTO 23	First N loop skipped if result is incremental
8 - 1 E3 / STO 01	N counter from 0 to J-1 stored in 01
GTO 02	
132♦LBL 02	Start of first N loop
XEQ "S" RCL IND 03	
RCL IND 02 *	NOPR(N) * NB(J) added to previous value of RES(N)
ST+ IND 04 ISG 01	
GTO 02	
140♦LBL 23	N counter set for second loop
RCL 67 1 E3 / RCL 00	N = J to MAXDEV of results
INT + STO 01 GTO 09	
149♦LBL 09	Start second N loop
XEQ "S" RCL 01 INT	
RCL 00 INT - 46 +	Address of PROJ(N-J+1) calculated in X
RCL IND X RCL IND 03	NOPR(N) substracted if results incremental (flag 00)
FC? 00 - FS? 00 X<>Y	
RCL IND 02 *	
ST+ IND 04 ISG 01	End 2nd N loop
GTO 09 ISG 00 GTO 01	End main J loop
SF 07 67 STO 04	Flag 07 set to list results
TONE 8 TONE 6	Address of MAXDEV-Results stored in 04 for list
GTO "LST"	Two beeps warn that calculation is terminated

Subroutine 'S': sets RES(N) address in 04 and NOPR(N) address in 03

```
177♦LBL "INP"
CF 06  XEQ "HD"
"⊦MAX DEV?"  20  FS? 05
12  PROMPT  X=Y?  SF 06
X>Y?  GTO "INP"
STO IND 04  STO 03  1 E3
/  STO 00  1  FS? 05
ST+ 00  FS? 05  ST+ 01
"CARD?"  XEQ "Y"
FC?C 01  GTO 40
```

INP: Input of time series

Input series corresponding to status of flags 03/04/05
Input years Ø or 1 to MAXDEV and copies to the end of series (20 or 12 years) by calling 'CPY'
Year counter in 00, MAXDEV address in 04, yearly values address in 01

```
203♦LBL 22
XEQ "CRD"  RDTAX  RCL 03
RCL IND 04  X>Y?  GTO 05
GTO 04
```

On option, input from card. Data on card same as with RPV and time series programmes

```
211♦LBL 05
"WRONG DATA"  PROMPT
GTO "INP"
```

Data on card refused if MAXDEV on card is more than MAXDEV specified initially

```
215♦LBL 40
XEQ "HD"  "⊦NAME?"  AON
PROMPT  AOFF  XEQ "SN"
```

40: Input of series name

```
222♦LBL 03
CLA  XEQ "CN"  "⊦ YR"
RCL 00  INT  XEQ "N"
"⊦ ?"  PROMPT
STO IND 01  1  ST+ 01
ISG 00  GTO 03
```

03: Loop to input yearly values
CN, sub-routine recalls series name for prompt

```
236♦LBL 04   FC? 06
XEQ "CPY"  XEQ "LST"
FS? 02  RTN  GTO "R"
```

04: Return to module LBL 21 if initial run

```
242♦LBL "LST"
"LIST?"  XEQ "Y"
FS?C 01  XEQ 15  FS? 07
GTO 16  "CHANGE?"
XEQ "Y"  FS?C 01  GTO 17
```

LST: calls for list/change/store

```
253♦LBL 16
CF 07  CF 03  CF 04
CF 05  "STORE?"  XEQ "Y"
FS?C 01  XEQ 18  FS? 02
RTN  GTO "R"
```

16: clears all flags used

STORE on card on option

```
265♦LBL 15
SF 21  XEQ "HD"  "⊦-"
XEQ "CN"  AVIEW
RCL IND 04  1 E3  /
STO 02  RCL 04  1  +
STO 01  1  FS? 05
ST+ 00  FS? 05  ST+ 01
GTO 07
```

15: Listing of time series - Printer controlled by flag 21

Year counter in 00

Address of yearly values in 01

```
285♦LBL 07
"YR"  RCL 00  INT
XEQ "N"  "⊦="
RCL IND 01  XEQ "Y"  1
ST+ 01  ISG 00  GTO 07
ADV  RTN
```

07: Loop for listing

299♦LBL 17	
RCL IND 04 STO 03	
"CARD?" XEQ "Y"	Possibility of new input from card
FS?C 01 GTO 22 CF 08	
CLA XEQ "CN" AON	Name of series is displayed and can be changed
PROMPT AOFF XEQ "SN"	
"⊢-YR?" 20 FS? 05 12	Editing of a year value
PROMPT X=Y? SF 06	Depending on year changed,
X>Y? GTO 17 X<0?	MAXDEV value stored in 04 is adjusted
GTO 17 RCL IND 04 X<>Y	Flag 08 set if year corrected is higher than
X>Y? STO IND 04 X>Y?	previous MAXDEV
SF 08 X=Y? SF 08	
STO 02 "YR" XEQ "N"	Old value is displayed and can be changed
"⊢=" RCL 04 + 1 +	
STO 01 RCL IND 01	
ARCL X "⊢?" PROMPT	Adjustment to MAXDEV depends whether or not
STO IND 01 FS? 06	copy to END is selected
GTO "LST"	
"COPY TO END?" XEQ "Y"	
FS? 01 GTO 19 1	
FS? 08 ST+ IND 04	
GTO "LST"	
356♦LBL 19	19: Copy to end
RCL 02 STO IND 04	
XEQ "CPY" GTO "LST"	
361♦LBL 18	Storage on card
XEQ "CRD" WDTAX RTN	
365♦LBL "CPY"	<u>CPY</u>: Copies value in year MAXDEV to the end of the
20 FS? 05 12 ENTER↑	series (i.e. to year 12 or 20)
1 E3 / RCL IND 04 +	
1 + STO 00 RCL IND 04	
RCL 04 + 1 + STO 01	
RCL IND 01 STO 02 1	
ST+ 01 GTO 06	

```
388♦LBL 06
RCL 02  STO IND 01  1
ST+ 01  ISG 00  GTO 06
RTN
```

06: Loop to copy

```
396♦LBL "HD"
"INCR RESULT"  FS? 00
"TOT RESULT"  FS? 03
"PROJ "  FS? 04  "NOPR "
FS? 05  "INCR NB "  RTN
```

HD: Selects header in ALPHA register, using flags

```
407♦LBL "CRD"
RCL 04  RCL IND 04  +  1
+  1 E3  /  RCL 04  +  1
-  RTN
```

CRD: Sets X with number of first and last memory registers for Read/Write card

```
420♦LBL "R"
"READY"  PROMPT  STOP
```

R: Prompts 'READY'

```
424♦LBL "Y"
AON  PROMPT  AOFF
ASTO Y  "N"  ASTO X
SF 01  X=Y?  CF 01  RTN
```

Y: Flag 00 is set if input is not N (i.e. if yes)

Flag 00 clear is input is N

```
435♦LBL "N"
FIX 0  CF 29  ARCL X
FIX 2  SF 29  RTN
```

N: Value in X attached to the content of ALPHA register with no decimal digit and no decimal point

```
442♦LBL "V"
SF 21  ARCL X  AVIEW
RTN
```

V: Flag 21 controls printer
Appends content of X to ALPHA register and displays ALPHA register

```
447♦LBL "S"
RCL 01  INT  68  +
STO 04  46  -  STO 03
RTN
```

S: Sets address - See page 2

```
457♦LBL "SN"
FS? 03  ASTO 43  FS? 04
ASTO 20  FS? 05  ASTO 05
RTN
```

SN: Stores name of series, using flags

```
465♦LBL "CN"
FS? 03  ARCL 43  FS? 04
ARCL 20  FS? 05  ARCL 05
FS? 07  ARCL 66  RTN
END
```

CN: Recalls name of series, using flags

Storage Plan

00 to 04	Temporary use for counters/addresses
05	Phasing variable - Name
06	Phasing period (INCR NB) MAXDEV
07-09	Phasing variable (INCR NB - years 0 to 12)
20	Without project - Name
21	Without project max. dev. (NOPR MAXDEV)
22-42	Without project with variable (NOPR - years 0 to 20)
43	With project - Name
44	With project max. dev. (PROJ MAXDEV)
45-65	With project unit variable (PROJ - years 0 to 20)
66	Results - Name
67	Max. dev. of results
68-78	Results of calculation (years 0 to 20)

Requires SIZE 089

Flags used : 00 to 08 - 21 - 24 - 25 - 27 (cf. page 1)

Programme size and global labels

```
            CAT 1
LBL'PHASE
LBL'INP
LBL'LST
LBL'CPY
LBL'HD
LBL'CRD
LBL'R
LBL'Y
LBL'H
LBL'V
LBL'S
LBL'SN
LBL'CN
END       1081 BYTES
.END.     04 BYTES
```

(155 Registers)

User keys

```
            KEYS
USER KEYS:
 21 "PHASE"
 34 SIZE
```

AN EXAMPLE OF MANIP UTILIZATION

The table below has been prepared with the MANIP programme. It is intended to suggest a standardized way of presenting all summary information required to evaluate a farm model.

Section A below includes methodological explanatory notes. Section B describes the content of the MANIP files where the model is stored, to allow utilization by various users. Section C is an annotated listing of the calculations performed by the model.

AN AGRICULTURAL DEVELOPMENT PROJECT
A FARM MODEL
Currency : L.'000

TOTAL INCOME AND CASH FLOW PROJECTIONS

4- 3-1983

Year	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
--TOT.VALUE OF PRODUCTION	4732	5340	6251	7275	8472	9665	9665	9665	9665	9665	9665	9665	9665	9665	9665	9665	9665	9665	9665	9665	9665
--OTHER INCOME	0	150	300	450	600	750	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900
TOTAL INFLOW	4732	5490	6551	7725	9072	10415	10565	10565	10565	10565	10565	10565	10565	10565	10565	10565	10565	10565	10565	10565	10565
--INVESTMENT COSTS	0	5280	0	0	0	0	0	0	0	0	3050	0	0	0	0	0	0	0	0	0	0
--OPERATING COSTS-EX.LAB	450	600	1050	2000	2500	2700	2700	2700	2700	2700	2700	2700	2700	2700	2700	2700	2700	2700	2700	2700	2700
--HIRED LABOUR COSTS	820	840	810	770	760	750	750	750	750	750	750	750	750	750	750	750	750	750	750	750	750
--FAMILY LABOUR COSTS	2200	2230	2170	2060	2020	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000
TOTAL OUTFLOW	3470	8950	4030	4830	5280	5450	5450	5450	5450	5450	8500	5450	5450	5450	5450	5450	5450	5450	5450	5450	5450
NET BEN. BEFORE FIN.	1262	-3460	2521	2895	3792	4965	5115	5115	5115	5115	2065	5115	5115	5115	5115	5115	5115	5115	5115	5115	5115
--FARMER'S SHARE(INVEST)	0	528	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
--LOAN RECEIPTS(INVEST)	0	4752	0	0	0	0	0	0	0	0	2745	0	0	0	0	0	0	0	0	0	0
--DEBT SERV. LONG T. LOAN (1)	0	713	713	1142	1142	1142	1142	1142	1142	1142	412	412	660	660	660	660	660	660	660	0	0
--INTEREST WORK. CAPITAL (2)	95	108	139	208	245	259	259	259	259	259	259	259	259	259	259	259	259	259	259	259	259
NET FINANCING-CURR. COSTS	-95	4459	-852	-1350	-1387	-1401	-1401	-1401	-1401	-1401	2075	-670	-919	-919	-919	-919	-919	-919	-919	-259	-259
NET BENEFIT AFTER FIN.	1167	999	1669	1545	2405	3564	3714	3714	3714	3714	4140	4444	4196	4196	4196	4196	4196	4196	4196	4856	4856
--VALUE SUBSISTENCE CONS.	2500	2650	2900	3200	3350	3350	3350	3350	3350	3350	3350	3350	3350	3350	3350	3350	3350	3350	3350	3350	3350
--FAMILY LABOUR COSTS	2200	2230	2170	2060	2020	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000
NET CASH BALANCE	867	579	939	405	1075	2214	2364	2364	2364	2364	2790	3094	2846	2846	2846	2846	2846	2846	2846	3506	3506
INCR. NET CASH BALANCE	0	-288	72	-462	209	1347	1497	1497	1497	1497	1923	2228	1980	1980	1980	1980	1980	1980	1980	2640	2640
CUMUL. INCR. CASH BAL.	0	-288	-216	-677	-469	879	2376	3873	5370	6868	8791	11018	12998	14978	16957	18937	20917	22897	24876	27516	30155
--DEBT SERV LT (DEFL) (3)	0	636	568	813	726	648	579	517	461	412	368	328	470	419	374	334	298	266	238	0	0
--INT.WORK.CAPITAL(DEFL) (3)	85	96	125	185	218	231	231	231	231	231	231	231	231	231	231	231	231	231	231	231	231
NET CASH BAL.(DEFL)	877	667	1098	757	1518	2736	2955	3017	3073	3122	2861	3206	3064	3115	3160	3200	3236	3267	3296	3534	3534
INCR.NET CASH BAL.(DEFL)	0	-210	221	-120	641	1859	2078	2140	2196	2245	1984	2329	2187	2239	2283	2323	2359	2391	2419	2657	2657
CUMUL.INCR.BAL.(DEFL)	0	-210	11	-109	532	2391	4469	6609	8805	11050	13035	15363	17551	19789	22071	24394	26753	29143	31562	34219	36676

(1) Interest rate 15% p.a. - 2 years grace and 7 years repayment
(2) 15% p.a. on total operating costs for 6 months
(3) Assuming 12% p.a. constant inflation rate

ANNEX 4

A. Specimen Total Income and Cash Flow Projection Table

Explanatory Notes on Methodology

Cash Inflow and Outflow

To be expressed in constant prices.

Total Value of Production

Gross value of farm output, including all production, whether sold or consumed on the farm.

Other Income

To be included only if changes occur because of the project: for example, a reduction in off-farm income because of increased on-farm labour use related to the farm development proposal, or an increase in income resulting from hiring out of oxen purchased as part of the investment programme.

Investment Costs

Includes both initial investment and subsequent replacement costs.

Operating Costs

The distinction between operating costs (excluding labour), hired labour costs and family labour costs, facilitates subsequent economic analysis and adjustments.

Debt Service - Long Term Loan

Interest and capital repayment expressed in current terms (i.e. at the rate of interest prevailing in the country) generally as a capital recovery factor. Footnote assumptions on interest rate, grace and repayment periods.

Interest on Working Capital

Generally to be calculated on the working capital requirements for the estimated period between the time when expenditure is incurred and the time of receipt of payments for sales. The current interest rate is used. Footnote assumptions on proportion of operating costs assumed to be covered by working capital loans, period outstanding and annual interest rate.

Adjustments to Establish True Cash Position

Value of Subsistence Consumption

To be subtracted from Net Benefit after Financing.

Family Labour Costs

To be added to Net Benefit after Financing (figures to correspond with those already shown in cash out flow).

Inflation

Where it is necessary to introduce adjustments to take account of inflation, the following steps are recommended:

Deflating Debt Service on Long-term Financing

The annual debt service on long term financing is deflated at specified rates applied cumulatively from the year in which the financing is first advanced. If a second batch of financing is assumed (as in year 10 of the specimen table), the time at which this is introduced should, of course, again be treated as the inital year for calculating the corresponding deflation factor. However, when a loan is disbursed over more than a year (but is treated as a single credit operation), loan receipts other than the first year should also be deflated. When applying the MANIP programme with the pre-defined operations, users not wishing to utilise this feature, may follow the alternative version of the pre-defined sequences and exclude the deflated lines. Footnote assumed deflation rates.

Deflating Interest on Working Capital

The interest payment (in current terms) should be deflated on a yearly basis by the estimated inflation rate applying to the particular year (i.e. using a simple rather than cumulative deflator). Footnote interest rate assumptions.

B. Use of MANIP for Preparing Cash-flow Analysis Tables

A pre-defined model is available within a set of files called (PROJECT) CASH-FLOW.

It includes:

i) 28 pre-defined variables, as follows:

Variables with their numbers

1. Total Value of Production
2. Other Income
3. Total Inflow
4. Investment Costs
5. Operating Costs - excluding labour
6. Hired Labour Costs
7. Family Labour Costs
8. Interest on Working Capital
9. Total Outflow
10. Net Benefit Before Financing
11. Farmer's Share (Investment)
12. Loan Receipts (Investment)
13. Debt Service Long Term Loan
14. Net Financing - Current Costs
15. Net Financing - Deflated
16. Net Benefit after Financing
17. Value Subsistence Consumption
18. Net Cash Balance
19. Incremental Net Cash Balance
20. Cumulative Incremental Cash Balance
21. Inflator (Cumulative)
22. Farmer's Share Factor
23. Inflator (Annual)
24. Interest on Working Capital (Deflated)
25. Debt Service Long Term (Deflated)
26. Net Cash Balance (Deflated)
27. Incremental Net Cash Balance (Deflated)
28. Cumulative Incremental Balance (Deflated)

Note: For the preparation of the example table (Table 1), all values were set at zero except variables 21 and 23 (set at 12% constant inflation) and variable 22 which was set at 1 in year 1. For variable 21, the rate of inflation has been applied only from year 2 since deflation of 1st year repayment is covered by variable 23.

ii) A pre-defined sequence of operations to perform the necessary calculations (see C. below).

iii) Pre-defined table formats (No. 1 without deflated lines and No. 2 with deflated lines). See Table 1 above.

C. Pre-Defined Sequence for Cash-Flow Projections

Sequence of operations No 1 CASH FLOW

Remarks

1 : Selected operation =AX + BY + C

X = 1 --TOT.VALUE OF PRODUCTION
Y = 2 --OTHER INCOME
Parameter A = 1.00
Parameter B = 1.00
Parameter C = 0.00
Result in variable No. 3 TOTAL INFLOW

2 : Selected operation =Balance & switching values

	Variable	Weight
1	No. 5 --OPERATING COSTS-EX.LAB	0.08
2	No. 6 --HIRED LABOUR COSTS	0.08

Balance in variable No. 8 --INTEREST WORK. CAPITAL
RoR and switching values NOT to be calculated

These weights provide the basis for calculating working capital requirements and must be adjuste depending on interest rate, period and proportio of operating costs

3 : Selected operation =Balance & switching values

	Variable	Weight
1	No. 4 --INVESTMENT COSTS	1.00
2	No. 5 --OPERATING COSTS-EX.LAB	1.00
3	No. 6 --HIRED LABOUR COSTS	1.00
4	No. 7 --FAMILY LABOUR COSTS	1.00

Balance in variable No. 9 TOTAL OUTFLOW
RoR and switching values NOT to be calculated

4 : Selected operation =AX + BY + C

X = 3 TOTAL INFLOW
Y = 9 TOTAL OUTFLOW
Parameter A = 1.00
Parameter B = -1.00
Parameter C = 0.00
Result in variable No. 10 NET BEN. BEFORE FIN.

5 : Selected operation =AX + B

X = 4 --INVESTMENT COSTS
Parameter A = 0.90
Parameter B = 0.00
Result in variable No. 12 --LOAN RECEIPTS(INVEST)

Long term loan assumed to be a given percentage of total investment (to be adjusted by user)

6 : Selected operation =AX + BY + C

X = 4 --INVESTMENT COSTS
Y = 12 --LOAN RECEIPTS(INVEST)
Parameter A = 1.00
Parameter B = -1.00
Parameter C = 0.00
Result in variable No. 11 --FARMER'S SHARE(INVEST)

7 : Selected operation =AXY + B

X = 11 --FARMER'S SHARE(INVEST)
Y = 22 FARMER'S SHARE FACTOR
Parameter A = 1.00
Parameter B = 0.00
Result in variable No. 11 --FARMER'S SHARE(INVEST)

Farmer's share of initial investment equals total investment minus loan and hence factor of 1 should be applied during initial investment period; to exclude farmer's share from calculation of replacement costs, apply factor = 0

8 : Selected operation =AXY + B

X = 12 --LOAN RECEIPTS(INVEST)
Y = 21 INFLATOR (CUMUL.)
Parameter A = 1.00
Parameter B = 0.00
Result in variable No. 12 --LOAN RECEIPTS(INVEST)

Loans received in years beyond year 1 are inflated (variable 21 should be set so that inflation rate is applied only from year 2)

9 : Selected operation =Debt service on X

Debt service calculated on variable No. 12 --LOAN RECEIPTS(INVEST)
Annual amounts treated as separate loans
Loans assumed to be disbursed by beginning of year
Interest rate = 15.00 %
Grace period = 2 years - Interest paid during grace period
Repayment period = 7 years - Equal instalments
Total debt service stored in variable No. 13 --DEBT SERV. LONG T. LOAN
Repayment assumed by end of year

Debt service calculated on inflated loan (loan characteristics to be adjusted by user)

10 : Selected operation =AX/Y + B

X = 12 --LOAN RECEIPTS(INVEST)
Y = 21 INFLATOR (CUMUL.)
Parameter A = 1.00
Parameter B = 0.00
Result in variable No. 12 --LOAN RECEIPTS(INVEST)

Loan receipt back to original values

11 : Selected operation =AX/Y + B

X = 13 --DEBT SERV. LONG T. LOAN
Y = 21 INFLATOR (CUMUL.)
Parameter A = 1.00
Parameter B = 0.00
Result in variable No. 13 --DEBT SERV. LONG T. LOAN

Debt service is now deflated cumulatively from year following the loan (whether for initial loan or for replacement loan)

12 : Selected operation =AX/Y + B

X = 13 --DEBT SERV. LONG T. LOAN
Y = 23 INFLATOR (ANNUAL)
Parameter A = 1.00
Parameter B = 0.00
Result in variable No. 25 --DEBT SERV LT (DEFL)

Debt service now deflated for one more year (non-cumulatively), i.e. from year of loan (this assumes interest paid at end of period, and not in advance)

13 : Selected operation =AX/Y + B

X = 8 --INTEREST WORK. CAPITAL
Y = 23 INFLATOR (ANNUAL)
Parameter A = 1.00
Parameter B = 0.00
Result in variable No. 24 --INT.WORK.CAPITAL(DEFL)

Interest on working capital deflated (non-cumulatively)

14 : Selected operation =Balance & switching values

	Variable	Weight
1	No. 11 --FARMER'S SHARE(INVEST)	1.00
2	No. 12 --LOAN RECEIPTS(INVEST)	1.00
3	No. 24 --INT.WORK.CAPITAL(DEFL)	-1.00
4	No. 25 --DEBT SERV LT (DEFL)	-1.00

Balance in variable No. 15 NET FINANCING - DEFLATED
RoR and switching values NOT to be calculated

15 : Selected operation =Debt service on X

Debt service calculated on variable No. 12 --LOAN RECEIPTS(INVEST)
Annual amounts treated as separate loans
Loans assumed to be disbursed by beginning of year
Interest rate = 15.00 %
Grace period = 2 years - Interest paid during grace period
Repayment period = 7 years - Equal instalments
Total debt service stored in variable No. 13 --DEBT SERV. LONG T. LOAN
Repayment assumed by end of year

Debt service calculated on non-deflated loan amounts

16 : Selected operation =Balance & switching values

	Variable	Weight
1	No. 11 --FARMER'S SHARE(INVEST)	1.00
2	No. 12 --LOAN RECEIPTS(INVEST)	1.00
3	No. 13 --DEBT SERV. LONG T. LOAN	-1.00
4	No. 8 --INTEREST WORK. CAPITAL	-1.00

Balance in variable No. 14 NET FINANCING-CURR. COSTS
RoR and switching values NOT to be calculated

17 : Selected operation =AX + BY + C

X = 10 NET BEN. BEFORE FIN.
Y = 14 NET FINANCING-CURR. COSTS
Parameter A = 1.00
Parameter B = 1.00
Parameter C = 0.00
Result in variable No. 16 NET BENEFIT AFTER FIN.

18 : Selected operation =Balance & switching values

	Variable	Weight
1	No. 16 NET BENEFIT AFTER FIN.	1.00
2	No. 7 --FAMILY LABOUR COSTS	1.00
3	No. 17 --VALUE SUBSISTENCE CONS.	-1.00

Balance in variable No. 18 NET CASH BALANCE
RoR and switching values NOT to be calculated

```
19 : Selected operation =Incremental values over year 0

X = 16 NET CASH BALANCE
Result in variable No. 19 INCR. NET CASH BALANCE


20 : Selected operation =Cumulative values of X

X = 19 INCR. NET CASH BALANCE
INCREMENTAL cum. values
Result in variable No. 20 CUMUL. INCR. CASH BAL.


21 : Selected operation =AX + BY + C

X = 10 NET BEN. BEFORE FIN.
Y = 15 NET FINANCING - DEFLATED
Parameter A =  1.00
Parameter B =  1.00
Parameter C =  0.00
Result in variable No. 26 NET CASH BAL.(DEFL)


22 : Selected operation =Balance & switching values

      Variable                                Weight
 1   No.  26 NET CASH BAL.(DEFL)            1.00
 2   No.   7 --FAMILY LABOUR COSTS-         1.00
 3   No.  17 --VALUE SUBSISTENCE CONS.     -1.00
Balance in variable No. 26 NET CASH BAL.(DEFL)
RoR and switching values  NOT to be calculated


23 : Selected operation =Incremental values over year 0

X = 26 NET CASH BAL.(DEFL)
Result in variable No. 27 INCR.NET CASH BAL.(DEFL)


24 : Selected operation =Cumulative values of X

X = 27 INCR.NET CASH BAL.(DEFL)
INCREMENTAL cum. values
Result in variable No. 28 CUMUL.INCR.BAL.(DEFL)


25 : Selected operation =Display/Print X

X = 19 INCR. NET CASH BALANCE
Printer is TERMINAL
No of decimal digits :     1


26 : Selected operation =Display/Print X

X = 27 INCR.NET CASH BAL.(DEFL)
Printer is TERMINAL
No of decimal digits :     1
```

ANNEX 5

MADS III

USER'S GUIDE

OCT. 1983

CENTRE D'INVESTISSEMENT

ORGANISATION DES NATIONS UNIES POUR L'ALIMENTATION ET L'AGRICULTURE

ROME

MADS III USER'S GUIDE

Table of contents

MADS III

Multipurpose Agricultural Data System (Version III)

User's guide

A. General description

This programme allows the user to create a data base representing an agricultural development project and to perform various calculations on it. The data is structured into various data types, and the programme borrows the basic concepts of data structuring (commodities, activities, plans) from MADS, a package developpped by OECD for several member countries, and various features from MADS II and DASI, two similar programmes developped by FAO (DDC and ESP Divisions).

The data must be prepared in a text file, (this file is called the source file), using a standard text editor. It is then processed by MADS III to create the data base. The programme is organized into five module corresponding to the various functions performed. They are:

- Start: used at the beginning of any run, either to create a new set of data files or to access an old data set created during a previous run. A data set is using several files (five), the names of which are generated from a generic name given by the user (length = 1 to 4 characters). One of these files is the so called output file discussed at the end of this chapter.

- Creating a new data base: to process a given text file where data is specified in text form (such a file is called the source file), thus creating various data files. The process includes various checks on data consistency. The module is called automatically when a new set of data files has been created by the "Start" module. It can be called by the user to replace the data base by a new one, from a different (or after modification of a) source file.

- Listing the data base: to print tables presenting the data.

- Processing a PLAN: to perform calculations on the data and print tables with the results, or copy the results to files that can be accessed afterwards by the MANIP programme (1) for further processing.

- Editing the data base: changes that do not affect the structure of the data base (i.e. changing the value of various parameters) can be done from within the programme. Other changes will require to terminate the programme, modify the Source file with the text editor, enter the programme again, and perform the "create" module with the modified source file.

---

(1) See MANIP user's guide.

After the "Start", the other four modules can be called in any order to perform the various operations. The programme is interactive and the user selects the various options from menus and questions displayed on the terminal.

Four options are available for printing the data base or the results of calculations:

- the terminal CRT (80 characters/line).

- The line-printer (132 characters/line).

- The HP (Hewlett-Packard) printer (227 characters/line).

- The output file (227 characters/line).

The output file is a file created by the programme using the same generic name as the data set, and the suffix T (e.g.: if generic name is TEST, the output file will be TEXT-T: TEXT). Any information printed into the output file can be accessed at the end of a MADS run by using the text editor programme (TED). It can then be looked at, modified, and printed like any other text. Note that the output file is emptied at the begining of a run.

B. Data structuring and other basic concepts

1. General approach

The data representing an agricultural development project can be organized in the form of "variables", i.e. time series of values, for a given number of periods corresponding to the analysis, associated with a name. Usually, the period will be the year, and on option year zero will be included in the series as a way to represent - when it is assumed constant - the without project situation. Each variable represents a cost or benefit item in a project (in quantities or in values), and a farm model or a full project will be no more than the algebric sum of a number of these variables. The MANIP programme is based on this approach: it allows to enter a set of variables in memory and to perform various operations on it. Assuming that the programme includes a sufficient set of functions to perform basic calculations on the variables, this approach proves to be very easy to understand by the user, and very flexible. The draw back is that every step of calculation has to be explicitely specified.

Another possible approach consists in organizing the data (the "variables" of the first approach) depending on the nature of the information represented by this data, so that it becomes possible to pre-define, within the programme, a number of standard calculations and output tables. The most obvious way is a hierarchical structure: the lowest level is the crop, defined by a yield and inputs requirements; crops are combined, together with investments, into farm models, themselves aggregated into sub-areas or project components, and finally at project level. The main benefit from this approach is that the user would specify the data according to categories that correspond to concrete realities. But the power and the flexibility of a programme based on this approach will depend directly on the variety of pre-defined data categories, pre-defined calculations, and pre-defined output tables.

The MADS approach wants to be a compromise between the two above approaches, i.e. between the table manipulation (spreadsheet approach) and the detailed data structuring. The idea is to structure the data to allow pre-defined calculations and output tables, but to limit the number a data types by using a different organization criteria: data is not organized on the basis of what it represents (e.g. crop yield, labour), but on the basis of the type of calculations to be performed on it: all informations requiring the same type of calculations can be groupped within the same data type.

2. The fundamental concepts: commodity, activity, plan

The calculations to be performed to evaluate an agricultural development project all correspond to the following general problem: calculate, at various future dates, a number of costs and benefits. These costs and benefits often correspond to physical quantities multiplied by prices; it can be said that they correspond either to the <u>consumption</u> or to the <u>production</u> of specific <u>commodities</u> (e.g. consumption of fertilizer, production of wheat grain), the only difference between a consumption and a production being the algebric sign. When costs or benefits are specified directly in values, it is always possible to represent them as production or consumption of quantities of an imaginary commodity with a price equal to one. It is thus possible to represent <u>all</u> costs and benefits within a project as consumptions and productions of various commodities.

A commodity is defined by its name and price; more precisely a name, a unit, and a time series of unit prices.

To be able to specify, in our model, consumptions and productions of commodities, we call activity everything that consumes or produces commodities. A typical activity is a crop: 1 ha of sorghum, that produces grain and straw and consumes labour, seeds, land. But the concept is wider: the activity cow consumes feed and labour and produces milk (and meat when it is culled), the activity dam consumes labour and concrete during its construction and produces electricity and water for irrigation.

An activity is defined by a name, a unit, and a list of commodities; for each commodity in the list, one has to specify wether it is produced or consumed, and the corresponding level (a time series of values), e.g. the "level" of the commodity "grain" produced is the yield. The same commodity can be more than once in the definition of an activity, with different levels (e.g. yield and post harvest losses).

Various activities can be combined in a linear way into a new data structure called a plan. A plan will be, for exemple, the combination of various crops and other activities into a farm model. A plan can also be made of other plans, allowing to define a project as a combination of farm models and other activities.

A plan is defined by a name, a unit, and a list of components, with their corresponding level. Components can be activities or plans, but also commodities or the other data types defined in a following section. The level of the activity wheat crop represents the area under that crop. The calculation can be performed in two different ways, annual or phasing, as explained below, and this also needs to be specified for each component as part of the spęcification of a plan.

3. Basic calculation modes: annual and phasing

Let us assume a commodity rice grain, an activity rice, and a plan rice perimeter specifying the number of hectares of rice being put under irrigation during a project, as shown below:

Years	0	1	2	3	4-20
Commodities					
Rice grain (ton)	150	150	150	150	150
Activities					
rice (1 ha)					
Rice grain produced	1.2	2.0	2.5	3.0	3.5
Plans					
Rice perimeter					
Activity rice	0	100	200	300	0

(incremental hectares - phasing calculation)

The price of rice is constant at 150 per ton. The yield is 1.2 ton/ha without project, and would gradually increase to 3.5 t/ha with project. Rice area would be 100 ha in year 1, 200 more in year 2, 300 more in year 3. Rice production can be calculated in two different ways:

i) If the yield depends on the project's year, the yield for a given year is multiplied by the total area. In the above example, production year 3 would be equal to 600 ha x 3 t/ha = 1.800 t. This calculation mechanism is called the annual mode.

ii) If the yield depends on the age of the activity, i.e. on the number of years since a particular unit of the activity got into the project, the calculation wil have to be done a different way, the so called <u>phasing mode</u>. In the above example, production in year 3 would be equal to:

100 ha started in year 1 x 3 t/ha	=	300
200 ha started in year 2 x 2,5 t/ha	=	500
300 ha started in year 3 x 2 t/ha	=	600
Total		1.400 t

This example illustrates several important points:

- When defining a plan, it is necessary to specify the calculation mode for each component. The calculation results will be in total quantities; when the without project situation is not constant, it will have to be defined as a different activity (e.g. "rice without project"), and the incremental values can be calculated by difference (a plan made of a plan "project" at level 1 and a plan "without project" at level - 1 with annual calculation mode).

- In the case of phasing mode, it is necessary to specify wether the level of the component is expressed in total number of units or in incremental numbers (over the previous year). The level will always be in total numbers in the case of annual mode.

- there is no difference, at the level of the specification of an activity, between an annual crop and a perenial crop: the specification of the calculation mode is done at plan level.

- It is important to have a clear understanding of the phasing calculation mechanism, particularly if there are several levels of phasing calculations.

4. <u>More data types</u>

The concepts of commodity, activity and plan, although very powerful, are not sufficient to handle a number of commonly required calculations. Four other data types have thus been introduced, and are presented below.

(a) <u>Investment</u>

The investment type is used both to allow for a distinction between investments and operating costs and to provide additional calculation mechanisms corresponding to some common ways of handling investment costs.

An <u>investment</u> is defined by a name, a unit, and a time series of unit prices, just like a commodity. Five additional parameters are also specified:

- investment life: to calculate automatic replacement and residual value,

- percentage for residual value,

- percentage for contingencies,

- percentage and time lag for maintenance (e.g. maintenance cost for a building are estimated at 5% of the initial cost, and will be applied annually from the fourth year).

The number of units of the investment item is specified by the level of the investment when introduced in a given plan. If an investment is specified directly in value, the price can be set to 1 and the values specified as the level within a plan.

(b) Loans

A credit operation can be defined, based on a data item used to calculate loan amounts and on various parameters on repayment conditions. The base item can be a commodity, an investment, or an aggregate (defined below), with values resulting from the calculations of a plan. When the base item is an investment, only the initial investment per se will be considered (i.e. investment and contingencies), and not the replacement, maintenance costs and residual values. The percentage of the base item to be financed is defined by the level of the loan within a plan.

Credit conditions are defined by six parameters:

- interest rate,

- grace period,

- repayment period,

- interest paid or not during the grace period,

- loan amounts to be considered as a single loan or as separate loans (having the same conditions),

- repayment as constant instalments or equal capital instalments.

(c) Aggregates

Aggregates are sub-totals, defined by a list of components (up to 30), with a weighting parameter for each component. Components can be commodities, investments, loans, or other aggregates. Aggregates are calculated for a given plan by adding up those items in the list of components that are present in the plan. For example if an aggregate cereals is defined as including the three commodities rice, maize and wheat, and a given plan produces only rice and maize, the aggregate will be calculated for these two commodities.

Aggregates can be based on quantities or on values of the components (this changes the effect of the weight). Aggregates are used as such to calculate sub-totals, or as input to loan amounts calculations, or to perform rate of return and sensitivity analysis calculations (see below).

(d) Transfers

Some cost/benefit items of a project cannot be expressed directly using the activity/plan concept, because they are based on results of calculations. For exemple, hired labour will be estimated at farm level as the balance between family labour availability and labour use by the various activities within the farm plan. The same applies to a tax based on production value, or a subsidy based on some input consumption.

Within the so-called transfer mechanism, a commodity, called transfer commodity (TC) can be added/substracted to a model on the basis of the values of another data item (commodity or investment), called the Source item. This "transfer" is based on the sign of the source in a given year, i.e. either on a surplus or on a deficit.

When the effect of the transfer is to compensate say a deficit, this deficit will be offset by a "transfer back" of the source commodity (source can only be a commodity, and not an investment item, if there is transfer back).

Example: in a farm model, the commodity "hired labour" is calculated for the years having a deficit of family labour; this deficit, being compensated, is put back to zero by a "transfer back" of the commodity family labour:

	Year 1	Year 2
Labour consumed (by crops)	150	120
Labour produced (by family)	130	130
Balance before transfer	-20	10
Transfer: hired labour	-20	0
labour (transfer back)	20	0
Balance: labour	0	10
hired labour	-20	0

To summarize, a transfer is defined by a name, a unit, the name of the source item, the name of the transfer commodity, and the type of transfer, i.e. based on surplus or deficit, and with or without transfer back.

The level of a transfer when specified within a plan represents the ratio source item/transfer commodity. The sign of this level indicates wether the transfer commodity is a cost (minus sign) or a benefit to the plan. In the above example, the level would be - 1 (1 man day of hired labour charged as a cost for every man day of deficit of family labour).

5. Standard calculations

For a given plan, the programme will calculate:

- commodities consumed, produced, transferred; commodities specified directly within a plan definition are presented together with commodities transfered, because only their algebric sign specifies wether they are consumed or produced (depending on the sign of the level of the commodity in the plan).

- Investments: value, contingencies, maintenance, replacement, and residual value (at the time of replacement and at the end of project life).

- Loans: amount, debt service, interest and capital components of debt service, outstanding loan at the end of each year.

- Overall plan balance (in value), and on option rate of return on this balance (if year zero is included, or is calculated on incremental values over year zero balance).

- Aggregates.

Results for commodities and aggregates will be in quantities and/or values. Results for aggregates and plan balance are presented with an algebric sign specifying wether they are a cost or a benefit.

Aggregates can be used for rate of return calculation, the programme will then produce present values and switching values (1) for the components of the aggregate.

Any line of results can be copied to a MANIP set of files, for further processing using the MANIP programme.

6. Sub-periods within the year

The programme does not include any specific arrangement to deal with sub-periods within the year. The problem can be handled by defining the data according to the sub-periods: for example, to analyse farm labour on a monthly basis, twelve commodities would be defined instead of one (labour-january, labour-february,....). Total annual labour can then be calculated using an aggregate.

7. Some comments

The flexibility of the programme results from defining the calculations to be performed through the structuration of the data.

The experience shows that the concepts of commodity, activity and plan are very powerfull and allow to describe a wide range of situations, but that this requires sometimes some imagination.

The nature and meaning of the results depends directly on the way the data is specified. Depending for example wether one or several commodities are used to describe several items in a plan, there will be one or several lines in the results. The possibility of calculating a certain project element depends directly on how this element is defined within activities and plans.

A simple example is presented in the chapter G of this manual and in appendix 3, to illustrate how the various components of a project can be translated into the various MADS data structures.

---

(1) The switching value, or switching percentage, of a component is the change in percentage that should be applied to this component, the others being unchanged, to bring the present value of the aggregate to zero (for a given discount rate).

C. Getting started

The programme is available on the DDC-NORSK computer from any terminal.

After completing the log-on procedure, the programme can be called.

Before MADS III can be used, the data must be prepared in a text file, using the text editor (TED - Appendix 1 gives some instructions on the use of TED).

The format instructions are given in chapter D. They are included in a standard file called MADS-SOURCE, stored in the user space PROJECT, that also already includes the various headings for the seven data types. The recommended procedure is as follows:

1. Read with TED the standard file MADS-SOURCE: @TED (PRO)MADS-SOURCE:TEXT.

2. Insert lines with your data, following the specifications included at the end of the MADS-SOURCE text.

3. Write the file with your own file name (within quotes name if such file does not exist yet).

To run MADS III, first call the programme: @MADS.

The text of this user's guide can be displayed on option. Follow the prompts and menus. You will have to give a generic name for your files (different from the source text file), and to give afterwards the name of your source file to create your data base.

D. Creating a data base

1. Preparing the text file (Source file)

Data must be presented in the text file following a certain format. All lines with key words for each data type, and separators (asterisks) must be present. The data for each data item must follow certain rules, as detailed below in the copy of the MADS-SOURCE file (see chapter C above).

Each data item must have a unique name. Names can include spaces, but must not be longer than the allowed length (they can be shorter if you respect the TAB instruction).

Here are the most common beginner's mistakes:

- forgetting unit name,

- forgetting an item type (e.g. C or P),

- putting the wrong number of values in a time series,

- misspelling an item name when used to define another item (e.g. commodity within an activity).

The create module will repport on error and wait for the user to press CR. Some errors will create plenty of artificial errors afterwards; better stop the programme using the ESC key, and correct the source file.

An other type of common mistake, not reported by the programme, relates to errors in data due to mistyping.

Data must always be checked carefully. It is also important to be carefull about the algebric sign of the data:

- commodities included directly into a plan must have a negative level to be counted as a cost,

- the same applies to the level of a transfer into a plan. The resulting transferred commodity will be a cost or a benefit depending on the sign of the level.

The MADS-SOURCE file is copied below.

```
A SAMPLE CASE STUDY PROJECT
MADS TEST DATA SET
'000 P
0 20
13.00
****************************
COMMODITIES
ENDATA
****************************
ACTIVITIES
ENDATA
****************************
INVESTMENTS
ENDATA
```

```
****************************
AGGREGATES
ENDATA
****************************
TRANSFERS
ENDATA
****************************
LOANS
ENDATA
****************************
PLANS
ENDATA
****************************
****************************
FORMAT FOR INPUT :
================

Headings :
  1st line : Project name (max. 50 char.)
  2nd line : Data set name (max. 30 char.)
  3rd line : Currency name (max. 10 char.)
  4th line : Base year (0 or 1) and project LIFE (max. 50 years)
  5th line : Opportunity cost of capital (in percentage)

Data :
set TAB to 30
FOR ALL TYPES :
 1st line : NAME (up to 25 char.) - TAB - UNIT (up to 10 char.)

COMMODITIES : 2nd line : Time series of PRICES

ACTIVITIES : For each commodity in the activity : 2 lines
            1st line : NAME - TAB - TYPE
            2nd line : Time series of COEFFICIENTS
             END on a line at the end of the list of a given commodity
  Possible types : C (for consumed), P (for produced)

INVESTMENTS : 2rd line : Time series of unit prices
               3nd  line  : 5 numbers : INV.LIFE, TIME LAG FOR MAINTENANCE, %
for
                 MAINTENANCE COSTS, % of CONTINGENCIES, % for RESIDUAL VALUE

AGGREGATES : 2 nd line : VALUES or QUANT
             1 line for each item in the aggregate :  NAME - TAB - WEIGHT
               Item can be commodity, investment, aggregate or loan.
               Number of items is limited to 30 per aggregate.
             END on a line at the end of the list of a given aggregate

TRANSFERS : 2st line : TYPE
            3nd line : NAME of SOURCE - TAB - NAME of TRANSFER COMM.
  Possible types : DEFBACK, DEFWITHOUT, SURPBACK, SURPWITHOUT
  If type is DEFWITHOUT or SURPWITHOUT, source can  be  a  commodity  or  an
investment.
  If type is DEFBACK or SURPBACK, source MUST be a commodity.
```

```
LOANS   :  2nd  line  : 3 numbers : GRACE PERIOD, REPAYMENT PERIOD, INTEREST
RATE%
         3rd line : PAID/NOTPAID - TAB - EQUAL.INST/CONST.CAPIT
         4th line : Name of item with LOAN AMOUNTS - TAB - SINGLE/SEPARATE
             Item  can  be  commodity,  investment, aggregate of commodities
and/or
             investments.

PLANS :  For each item in the plan : 2 lines
            1st line : NAME - TAB - TYPE
            2nd line : Time series of COEFFICIENTS
          END on a line at the end of the list for a given plan
  Possible types : A (for annual calc),
                   PT (for phasing calc - nbr of units is total )
                   PI (for phasing calc - nbr of units is incremental )
                   TR if item is a transfer
                   PT and PI can be used only for activities and plans.
  Possible items : anything but an aggregate.

TIME SERIES
===========
On the same line :
 - Type : A(ctual), L(inear interpolation), C(ompound rate)
 - Numbers, depending on type :
A : 1st number : year when values get constant -  next : list of values
Number of values MUST be equal to - 1st number if BASE=1
                                  - 1st number + 1 if BASE=0
L, C : 1st number : Value in base year (0 or 1)
       2nd and 3rd numbers : First and last year of change
       4th number : if L : Value in last years
                    if C : Compound rate in % (negative if decrease)
To separate numbers, use only SPACEs (i.e. blanck characters)
(*************************************************************************)
```

2. <u>Data consistency</u>

When processing the source file, the create module will perform a number of tests on data consistency, and send appropriate messages if errors are found. They are:

(i) Test on loans: the item used to calculate a loan amount can be an aggregate; such an aggregate cannot include a loan (financing of a loan with another loan is not allowed).

(ii) Tests on aggregates: when aggregates are based on quantities, the following restrictions apply:

- all components must have the same unit (be carefull with the spelling) and this unit must be the same as the unit of the aggregate, as it is unlikely to be correct to add quantities that are measured in different units.

- An aggregate based on values only cannot be a component of an aggregate based on quantities.

- A loan cannot be a component of an aggregate based on quantities, because loans are expressed only in values.

(iii) Test on transfers: a commodity which is used as transfer commodity in a transfer definition cannot be used as source item for any transfer in the data base. If this were allowed, the result of transfer operations could vary depending in which order the transfers are calculated.

Some additional tests on data consistency are performed only when the calculations of a plan are executed. Corresponding error messages will be displayed by the programme when the calculations are performed. They are:

(i) Test or transfers: if the investment or commodity item which make up the source of a transfer activity introduced in a given plan is not actually component of that plan, then the value of the transfer commodity will be assumed as zero.

(ii) Tests on loans:

a) The level of a loan within a plan must be positive, as well as the values of the item used to calculate loan amount; negative values will be assumed as zero in the calculations. This is because a loan can finance only costs.

b) If any of the investment or commodity items which make up the loan amount of a given plan are not actually component of that plan, then the loan amount over the entire time series will be assumed as zero since part of the basis for calculation is missing.

c) When the various values in the loan amount lines are to be considered as a single loan, grace period must be sufficient so that the first repayment can take place only after the last loan instalment has been received. If specified grace period is too short, the programme will assume the minimum acceptable value.

d) When a loan amount is made of items which price varies with time, the loan cannot be aggregated into higher level plans using the phasing calculation mode, as such a calculation would give erroneous results. In such a case, loan amount will be assumed as zero.

e) When a loan specified within a plan is also part of an inner plan (i.e. plan A contains loan B and plan C, and plan C also contains loan B), loan will be calculated only for the deepest level (i.e. for plan C; loan B will be ignored if plan A is calculated).

E. Processing a plan

1. Calculations

Calculations performed are listed in section B-5, and illustrated by the small example presented in section G and in Appendix 3.

Calculations on quantities of commodities and investments are performed at once for all the plans defined in the data base. This is done the first time the calculations module is called after creating a data base, or when data related to quantities has been changed with the editing module. If there are many plans (and even more if the system is overloaded), the calculations might take several minutes; don't be surprised.

The first step (by selecting from the menu on screen) is to select a plan. If results are required in values, they are to be requested explicitely (by selecting "calculating values" from the menu).

All results from a given plan can then be printed, except those that are calculated only if the corresponding option is selected from the menu:

- overall balance, and rate of return on this balance,

- calculations of aggregates, including on option ROR and present values/swithching values on the components.

2. Transfer to MANIP

Results of a plan can be transfered to MANIP, i.e. written to a set of disk files that can be accessed later on with the MANIP programme.

Various facilities are available:

- The MANIP files, if they do not exist yet, can be created by the MADS programme, and the necessary information copied into them (Project name, project life, etc.);

- it is possible to use, as a generic name for the MANIP files, either the same name as for the MADS files currently used or any other name supplied by the user;

- it is possible to list variables in the destination (MANIP) files. Lines of results from a MADS plan can be copied in bloc or selectively. They can be either appended to existing data in the MANIP files, or be written starting from a specific variable number. (This feature allows to insert results from MADS into an already defined MANIP model). The names of the lines of data copied from MADS into MANIP can be copied themselves or not;

- The lines of results copied from MADS into MANIP will be named using the name of the corresponding data item, plus a number of suffixes depending on the type of results. The suffixes are the following:

. for commodities:
 -COCQ: Commodity consumed - quantities
 -COPQ: Commodity produced - quantities
 -COTQ: Commodity transferred - quantities
 -COCV: Commodity consumed - values
 -COPV: Commodity produced - values
 -COTV: Commodity transfered - values

. for investment items:
 -VALUE: investment value (i.e. cost)
 -RPCST: replacement cost
 -CONTG: physical contingencies
 -MAINT: maintenance cost
 -RSVAL: residual values

. for loans:
 -VALUE: loan value (amounts lent)
 -DBTSV: total debt service
 -INTCM: interest component of debt service
 -CAPCM: capital component of debt service
 -OUTLN: outstanding capital at end of year

for aggregates:
 -AGGRQ: aggregate in quantities
 -AGGRV: aggregate in values

for balance:
 -PLNBL: plan balance

F. Editing the data base

1. How to do it

Any modification to the data base can be done by modifying the source file (with the text editor), and processing it again to create from scratch a new data base.

This is the only possibility for any change that would affect the structure of the data (e.g. creating or deleting an activity), or that might affect data consistency (e.g. renaming a commodity name, or the definition of a transfer).

changes limited to the value of the various parameters can also be done from within the MADS programme, through the selection of the proper options from menus and prompts. Note that such changes will affect the data within your data files (those corresponding to the generic name you gave at the beginning of the run), but will not modify the content of the source file (text) that was used to create the data base.

Below is the list of the parameters that can be modified from within the MADS programme:

- opportunity cost of capital,
- prices of commodities and investment items,
- investment characteristics (life, contingencies %, etc.),
- loan characteristics (interest rate, repayment, etc.),
- level of a component within an activity or within a plan,
- weight of a component within an aggregate.

2. Effect on stored calculation results

As calculations on quantities of commodities and investments are performed at once for all plans, and kept on file, they will have to be performed again whenever a plan or an activity is edited, but not when only prices or other data are modified. Results in value on the last plan calculated as well as results of calculations of loans and aggregates, will be offset by any editing except the opportunity cost of capital.

G. Examples.

1. A simple example

The simple example presented below is intended to illustrate the various concepts of MADS III. Data organization for the preparation of the source file is explained below. The listing of the source file as well as the various tables (input data and results) produced by MADS from this data are presented in Appendix 3. All names in the example are in French, to illustrate how models can be built in any language.

The first data section, in the source file, includes five lines corresponding to project name, run or data set name, currency name, base year/project life, and opportunity cost of capital.

Example:

```
Mali - Projet d'irrigation
Analyse financière
FCFA
1    20
12.0
```

The next step is to identify, within the project data, the variables corresponding to the various data types.

Let us assume that the project to be analyzed consists of the construction of an irrigation system in an area presently cultivated with flood recession rice, and where it would be possible, with irrigation, to cultivate irrigated rice in some area, in rotation with some rainfed maize, and floating rice in the lower parts of the perimeter. Crop production costs consist of inputs and labour. Project level costs include the investment for land development, maintenance of the perimeter, extension services, and pumping of water for irrigation.

First step will be the definition of commodities, together with their prices: rice, maize, inputs, labour, pumping, extension services. Because in this example inputs are specified directly in value within the crop budget, their price is put to 1000 in the commodity definition, and the values will be specified in 1000 currency units when inputs are introduced into a crop activity.

The commodity section of the source file will be as follows:

```
COMMODITIES
Riz                tonnes
A 1   180000
Maïs             tonnes
A 1   230000
Intrants           1000 FCFA
A 1     1000
Travail            homme-jour
A 1      700
Pompage            m3
A 1        5
Encadrement        millions CFA
A 1  1000000
ENDATA
```

Each crop will be specified in the section defining activities, the yield being a commodity produced (note that a crop could produce more than one commodity: e.g. straw and grain), and inputs and labour being consumed.

It is important to keep in mind that data items must have uniques names. If rice is the name of a commodity, the same name cannot be used for an activity or a plan.

Because some production will disappear with the project, it is necessary to specify a "rice without project" crop.

The activity section is thus as follows:

```
ACTIVITES
Riz irrigué                 Ha
Riz                          P
A   4   2   2.5   3   3.5
Intrants                     C
A   4  35  40    45  52
Travail
A   1   118
END
Riz flottant                 HA
Riz                          P
A   3   1   1.5   2
Intrants                     C
A   3  40   50   60
Travail                      C
A   1  98
END
Riz sans projet
Riz                          P
A   1  0.5
Intrants                     C
A   1  11.4
Travail                      C
A   1  54
END
Maïs (projet)                HA
Maïs                         P
A   3   1   1.5    2
Intrants                      C
A   3  20   30    35
Travail                       C
A   1   118
END
ENDATA
```

The next section in the source file deals with investments:

```
INVESTMENTS
Aménagement       milions CFA
A     1     1000000
50    3     5     10     0
ENDATA
```

Because in this example investment costs will be specified in value and in millions of currency unit (within a plan), the unit price here is one million. Investment life is to be 50 years, maintenance cost will be calculated at 5% of the investment cost, and will apply from the fourth year (i.e. 3 years time lag) after the investment. The percentage is 10% for physical contingencies and zero for residual value at the end of investment life.

The next section, aggregates, will be used to define for example total cereals production as equal to rice plus maize.

It is also used to define an aggregate "perimeter" that includes different elements than the plan "project" defined later on, and will be used to perform an economic analysis: the main model is for financial analysis in the example, but this aggregate will produce an economic analysis on the assummption that credit and labour have no economic cost and that all other items have economic prices identical to their financial prices (if this were not the case, the full source file could be copied under another name, and the prices edited in the commodities section).

The aggregates section reads as fellow:

```
AGGREGATES
Céréales            tonnes
QUANT
Riz                      1
Maïs                     1
END
PERIMETRE
VALUES
Riz                      1
Maïs                     1
Aménagement              1
Intrants                 1
Encadrement              1
Pompage                  1
END
ENDATA
```

Note that when an aggregate is calculated the results are expressed in algebric terms, i.e. for example a commodity consumed will be added with negative values. It would thus be <u>an error</u> to specify the weight of a commodity like "Intrants" with a negative sign because it is a cost.

The mechanism of the transfer will be used, in this example, to calculate pumping costs. This applies because pumping costs (i.e. water consumption) are assumed here to be proportional to crop production. Let us assume that only rice requires pumped water, at the rate of 1.200 m3 per ton of irrigated (dressed) rice, and 2.000 m3/ton of floating rice, at a cost of 5 FCFA per cubic meter. The following transfer can be defined:

```
TRANSFERS
Eau de pompage         m3
SURPWITHOUT
Riz                    Pompage
ENDATA
```

The quantity of pumped water (commodity "Pompage" will be calculated on the basis of rice production (i.e. based on positive values of the commodity "riz" for the plan where the transfer will be introduced, without transfer back of rice). Because we decided to define more than one rice crop to produce the same commodity, and the two crops have got different water requirements, it will be necessary to define two separate plans to calculate "pompage", as shown in the last data section. Another possibility would have been to define two commodities (floating rice and dressed rice, and on aggregate to get total rice production), two transfers to calculate pumping costs, and a single plan.

The next data section allows to specify credit operations. Assuming that inputs would be partly financed on credit, we could specify the following:

```
LOANS
Crédit de campagne      FCFA
0    1    13.5
PAID                      EQUAL.INST
Intrants                  SEPARATE
ENDATA
```

The three figures on the second line mean no grace period, 1 year repayment period, 13.5% interest rate. The next line means that interest is paid during grace period and repayment is in equal instalments (these specifications have no effect in the particular case of an annual loan, but must be present in the text). "Intrants" is the name of the commodity which cost will be used as a basis to calculate the amount of the loan, and SEPARATE means that yearly costs will be considered as the basis for separate loans.

Last comes the plans section:

```
PLANS
Riz dressé                    Périmètre
Riz irrigué                   PT
A    4    0    100    250    450
Eau de pompage                TR
A    1    -1200
END

Riz bas-fonds                 Périmètre
Riz flottant                  PT
A    4    0    75     150    250
Eau de pompage                TR
A    1    -2000
END

Projet                        Périmètre
Riz dressé                   A
A    1    1
Riz bas-fonds                 A
A    1    1
Maïs                         PT
A    5    0    20    60    120    150
Crédit de campagne           A
A    6    0.60    0.60    0.50    0.40    0.30    0.0
Riz sans projet               A
A    1    -200
Encadrement                   A
A    1    -4
Aménagement                  A
A    6    396.6    486.4    177.7    64.7    39.8    0
END
ENDATA
```

As already explained, the plans "Riz dressé" and "Riz bas-fond" are necessary to calculate pumping requirements. The "levels" of the transfer "Eau de pompage" in these two plans, representing the number of cubic meters of water per ton of rice, must be negative if pumping is to be counted as a cost. The parameter PT means that the calculation on rice has to be done following the phasing mode (letter P), and that the number of units (i.e. the rice area) is total (letter T), and not incremental from one year to the next (in which case the parameter would be PI).

The plan "project" includes the two other plans ("riz dressé" and "riz bas-fond") as well as two activities ("maïs" and "riz sans projet", the latter being substracted from the results because the level is negative), one investment (specified in this case in CFA millions), a credit operation (the loan will finance initially 60% of inputs cost, to decrease to zero from year 6), and a commodity (the level, here the cost in CFA millions of extension services, must be negative if it is not to be accounted for as a benefit).

Appendix 3 reproduces the full source file corresponding to this example, including the instructions copied from the MADS-SOURCE file (see sections C and D). It also shows all the data in the form of tables, as produced by MADS, and all the results of the calculations of the plan project, including production of maize and rice, consumption of inputs, labour and pumped water, costs associated with land development, loan and debt service, and aggregates with associated rate of return/present value/sensitivity analysis calculations.

2. The case of labour

As shown by the example in B.4.(d), calculating hired labour based on the calculation of the balance of labour consumed by crops and labour available from the family is an obvious case for using the transfer mechanism.

But there is a problem in this particular case, because a surplus of labour should not result into a corresponding benefit to the farm model. Let us go back to the previous example:

	Year 1	Year 2
Labour consumed (by crops)	150	120
Labour produced (by family)	130	130
Balance before transfer	-20	10
Transfer: hired labour	-20	0
labour (transfer back)	20	0
Balance: labour	0	10
hired labour	-20	0

In year 1, balance of labour is zero (in man-days), so that the cost of family labour would be zero to the model. In year 2, the balance being 10, the mechanism of the calculations would result into an income corresponding to the value of 10 working days, which would be wrong in the case of labour, as unused labour capacity does not correspond to an income. To overcome this problem, one can define another commodity (say "Family labour cost"), and introduce it into a plan in such a way that it would offset the "benefit" from the "labour produced" by the family. This could be done the following way:

- define three commodities: labour, family labour cost and hired labour,

- define an activity "family labour" that produces "labour" (representing family labour availability) and consumes "family labour cost" to offset the value of "labour" produced,

- define crop activities, that include labour requirements (consumption),

- define a transfer to calculate "hired labour" requirements in case of deficit of "labour",

- define a plan including the crop activities, the activity "family labour", and the transfer.

The result would be:

	Year 1	Year 2
Labour consumed (by crops)	150	120
Labour produced (by family)	130	130
Family labour cost	-130	-130
Labour balance before transfer	-20	10
Transfer: hired labour	-20	0
labour (transfer back)	20	0
Balance: labour	0	10
hired labour	-20	0
family labour cost	-130	-130

In values, the balance between the two commodities "labour" and "family labour cost" will correspond to the cost of the labour actually used by the crop activities, e.g. in year 2, minus 130 plus 10 equal 120.

TED Beginners guide

In order to use MADS III, it is necessary to prepare the data in the form of a text file, and thus to know how to use some text processing programme. Below are some basic informations on the one called TED, as well as on some basic features of file organization within the DDC NORSD-100 computer. Refer to specific manuals or to your secretary for more informations.

1. File names

The work space is organised into directories and users. Normally a user will be given space in only one directory, and you will not have to worry about directory name. User name is necessary for two things:

- to log-on,

- to access a file in another user space.

Example of a file name: (PROJECT-WORK:PROJECT) MADS-SOURCE:TEXT

Note that there should be no spacing within the file name. In this example, PROJECT-WORK is a directory name and PROJECT a user name (brackets and colon are compulsory). Because a file name can be abbreviated to the shortest non-ambiguous name, these two elements would not be normally necessary, except as already said if you want to access a file from a different user space.

The last part of the full name (:TEXT in the example) is the file type. It is very often unnecessary to specify it, as many commands in the system assume a default type. In particular, the type TEXT is assumed by default whenever a file name is used with TED.

2. Using quote marks with file names

A file name can be specified with or without quote marks, as shown below:

"(PROJECT)MADS-SOURCE"
or (PROJECT)MADS-SOURCE

The general convention for all commands using file names within the system is the following:

- the absence of quote mark implies that the file exists; a command with a file name without quote marks will result into an error message if the file cannot be found.

- The use of quote marks implies that the file does not exist and has to be created; if a file with the given name already exists, the system will send an error message and the command will have no effect.

3. Using files with TED

To read an (existing) file with TED, just type, after the system prompt @ and without space in between, TED <file name> (with a space between TED and the file name).Example: @TED (PROJECT)MADS-SOURCE. The text of the file then appears on the screen.

Write to a file from TED: the current text is stored in a temporary work space, and must be written to a file if you want to keep it. Commands in TED are given when the cursor is in the top line of the screen, not in the text. The cursor is moved to this position with the key . The command to write is W. If you then press the CR key, the current file name will appear after the command (if there is any), and a new CR will execute writing to this file. You can also write a file name and then press return. If you give a name which is not used yet and put it within quote marks (see above), a new file will be created with that name.

4. Writing text

The cursor is moved on the screen by using the arrow keys. The key (Home key) moves the cursor back and forth between the command position and the text. Instructions can be given to TED either from the command position (it is called a home command) or from within the text by using control characters. A control character is produced by pressing simultaneously the CTRL key and another key. Control A will be noted as <A>.

Most useful are the following:

Home commands

H (for HELP) prints on the screen the list of home commands, controls characters, and other instructions (for text formatting).

M: MOVE TO LINE: <line-number> <CR>.

T:HORIZONTAL TAB: give 1 to 8 tab position, then CR, to set TABS.

W: write to file.

Control characters

<A>: erases one character at a time.

<D> <D> (control D twice): deletes the current line.

<L>: creates a blank line above the current line.

<Q>: restores the line deleted last, at the current cursor position (the original one, or another location).

```
****** M A D S III ******

       MULTIPURPOSE AGRICULTURAL DATA SYSTEM
  VERSION 1.0 -- COPYRIGHT FAO INVESTMENT CENTER - ROME                COMMENTS

Do you want a general description of the program? (Y or N)
N

Do you want to use data which have previously been stored             Start
in your set of personal files ?
Y

Give the generic name of your files (1 to 14 characters)
MALI

Are you absolutely certain that the name is correct ?
Y

MALI-T:TEXT          is to be used as OUTPUTFILE
Note : The old content of that file is erased at the start of a new run
Press CR key when you want to continue

Mali - Projet d'irrigation

Current data set is Analyse financiere

Code = 1    Creating a new data base                                   Main menu
Code = 2    Listing the data base
Code = 3    Processing a PLAN
Code = 4    Editing the data base
Code = 5    End of job

Enter a number between  1 and  5
1

Give the name/type of your source file (1 to 21 characters)
MALI-SOURCE

MALI-SOURCE                                                   Creating a data base:
Are you absolutely certain that the name is correct ?         When a Source file is
Y                                                             processed, item names are
                                                              displayed on the terminal,
Mali - Projet d'irrigation                                    allowing to locate errors
Analyse financiere                                            (if any). The data is the
FCFA                                                          same as in appendix 3.
BASE = 1 - LIFE = 20
OPPORTUNITY COST OF CAPITAL = 12.00%
COMMODITIES
COMMODITIES = Riz                           tonnes
COMMODITIES = Mais                          tonnes
COMMODITIES = Intrants                      1.000 FCFA
COMMODITIES = Travail                       homme-jour
COMMODITIES = Pompage                       m3
COMMODITIES = Encadrement                   millions
ENDATA
ACTIVITIES
ACTIVITIES  = Riz irrigue                   Ha
Riz
Intrants
Travail
END
ACTIVITIES  = Riz flottant                  Ha
Riz
Intrants
Travail
END
```

```
2

ACTIVITIES  = Riz sans projet            Ha
Riz
Intrants
Travail
END
ACTIVITIES  = Mais (projet)              Ha
Mais
Intrants
Travail
END
ENDATA
INVESTMENTS
INVESTMENTS = Amenagement                millions
ENDATA
AGGREGATES
AGGREGATES  = Cereales                   tonnes
Riz
Mais
END
AGGREGATES  = PERIMETRE                  U
Riz
Mais
Amenagement
Intrants
Encadrement
Pompage
END
ENDATA
TRANSFERS
TRANSFERS   = Eau de pompage             m3
ENDATA
LOANS
LOANS       = Credit de campagne         fcfa
ENDATA
PLANS
PLANS       = Riz dresse                 Perimetre
Riz irrique
Eau de pompage
END
PLANS       = Riz bas-fonds              Perimetre
Riz flottant
Eau de pompage
END
PLANS       = Projet                     Perimetre
Riz drfonds
Mais (projet)
Credit de campagne
Riz sans projet
Encadrement
A
END
ENDATA
Presshen you want to continue

Mali — Projet d'irrigation

Current data set is Analyse financ
Code = 1    Creating a new data base                         Back to main menU
C = 2   Listing the data base
Code3   ng a PLAN    Editing the data base
Code = 5    End of job

Enter a number between  1 and  5
2
```

```
                    Mali - Projet d'irrigation                       Listing the data base
                       Analyse financiere
                        Currency : FCFA

      PRINTING INPUT DATA

  Code = 1 --- Catalogue
  Code = 2 --- COMMODITIES
  Code = 3 --- ACTIVITIES
  Code = 4 --- INVESTMENTS
  Code = 5 --- AGGREGATES
  Code = 6 --- TRANSFERS
  Code = 7 --- LOANS
  Code = 8 --- PLANS
  Code = 9 --- Full printout
  Code =10 --- Set nbr of dec. digits (current=1)
  Code =11 --- End of print

  Enter a number between 1 and 11
  9

  You have the following options :
  Code = 1     Terminal CRT
  Code = 2     Line-printer (132 chrs/l)
  Code = 3     HP-printer (227 chrs/l)
  Code = 4     Outputfile (227 chrs/l)

  Enter a number between  1 and  4                                    Full listing to outputfile
  4

                    Mali - Projet d'irrigation
                       Analyse financiere
                        Currency : FCFA

      PRINTING INPUT DATA

  Code = 1 --- Catalogue
  Code = 2 --- COMMODITIES
  Code = 3 --- ACTIVITIES
  Code = 4 --- INVESTMENTS
  Code = 5 --- AGGREGATES
  Code = 6 --- TRANSFERS
  Code = 7 --- LOANS
  Code = 8 --- PLANS
  Code = 9 --- Full printout
  Code =10 --- Set nbr of dec. digits (current=1)
  Code =11 --- End of print

  Enter a number between  1 and  11                                   End of listing operations
  11

  Mali - Projet d'irrigation

  Current data set is Analyse financiere

  Code = 1     Creating a new data base
  Code = 2     Listing the data base
  Code = 3     Processing a PLAN                                      Back to main menu
  Code = 4     Editing the data base
  Code = 5     End of job

  Enter a number between  1 and  5
  4
```

```
4

                    Mali - Projet d'irrigation
                       Analyse financiere
                        Currency : FCFA

      EDITING INPUT DATA                                        Editing the data base

  Code = 1 --- COMMODITIES
  Code = 2 --- ACTIVITIES
  Code = 3 --- INVESTMENTS
  Code = 4 --- AGGREGATES
  Code = 5 --- TRANSFERS
  Code = 6 --- LOANS
  Code = 7 --- PLANS
  Code = 8 --- Opportunity Cost of Capital
  Code = 9 --- End of editing

  Enter a number between  1 and   9
  6
          LOANS

  15 Credit de campagne       fcfa
  Select code of item to be modified
  (  14 if no more)

  Enter a number between 14 and  15
  15

  Data on loan Credit de campagne
   Loan based on Intrants

   1 : GRACE PERIOD =    0
   2 : REPAYMENT PERIOD =    1
   3 : INTEREST RATE =  13.50 %
   4 : Interest paid during grace period                        Editing loan data
   5 : Equal instalments
   6 : Annual amounts treated as separate loans

  To change data, enter corresponding line number (0 if no more change)

  Enter a number between  0 and   6
  0
          LOANS

   15 Credit de campagne       fcfa
  Select code of item to be modified
  (  14 if no more)

  Enter a number between 14 and  15
  14
                    Mali - Projet d'irrigation
                       Analyse financiere
                        Currency : FCFA

      EDITING INPUT DATA

  Code = 1 --- COMMODITIES
  Code = 2 --- ACTIVITIES
  Code = 3 --- INVESTMENTS
  Code = 4 --- AGGREGATES
  Code = 5 --- TRANSFERS
  Code = 6 --- LOANS
  Code = 7 --- PLANS
  Code = 8 --- Opportunity Cost of Capital
  Code = 9 --- End of editing

  Enter a number between  1 and   9
  3
```

```
        INVESTMENTS

  11 Amenagement                   millions
 Select code of item to be modified
 ( 10 if no more)

 Enter a number between 10 and 11
 11

 Investment item Amenagement

 Change prices (code=1), other data (code=2) or nothing (code=3) ?

 Enter a number between  1 and   3
 2

 Data on investment item Amenagement

  1 : LIFE =  50
  2 : RESIDUAL VALUE =   0.00 % of investment                    Editing investment data
  3 : % for CONTINGENCIES =  10.00
  4 : % for MAINTENANCE =   5.00
  5 : TIME LAG for MAINTENANCE =   3

 To change data, enter corresponding line number (0 if no more change)

 Enter a number between  0 and   5
 0

 Investment item Amenagement

 Change prices (code=1), other data (code=2) or nothing (code=3) ?

 Enter a number between  1 and   3
 3

        INVESTMENTS

  11 Amenagement                   millions
 Select code of item to be modified
 ( 10 if no more)

 Enter a number between 10 and 11
 10

                    Mali - Projet d'irrigation
                        Analyse financiere
                         Currency : FCFA

        EDITING INPUT DATA

 Code = 1 --- COMMODITIES
 Code = 2 --- ACTIVITIES
 Code = 3 --- INVESTMENTS
 Code = 4 --- AGGREGATES
 Code = 5 --- TRANSFERS
 Code = 6 --- LOANS
 Code = 7 --- PLANS
 Code = 8 --- Opportunity Cost of Capital
 Code = 9 --- End of editing

 Enter a number between  1 and   9
 9
```

```
6

Mali - Projet d'irrigation

Current data set is Analyse financiere

Code = 1    Creating a new data base
Code = 2    Listing the data base                          Back to main menu
Code = 3    Processing a PLAN
Code = 4    Editing the data base
Code = 5    End of job

Enter a number between  1 and  5
3                                                          Calculations:
                                                           Plan names are displayed to
Quantities being calculated. Please wait . . .             keep you waiting while
                                                           quantities are calculated
                                                           for all plans in the data base.
Plan Riz dresse
Plan Riz bas-fonds
Plan Projet

Mali - Projet d'irrigation

Current data set is Analyse financiere

Code = 1    Select a plan
Code = 2    Calculating values
Code = 3    Printing quantities/values
Code = 4    Calculating balance/return on a plan
Code = 5    Aggregates and sensitivity analysis
Code = 6    Transfer results to MANIP
Code = 7    End of plan processing

Enter a number between  1 and  7
1

        PLANS

16 Riz dresse              Perimetre
17 Riz bas-fonds           Perimetre
18 Projet                  Perimetre

Index number of plan to be processed ?
Enter   15 if no plan

Enter a number between 15 and  18
18

Projet                    --- Is it OK ?
Y                                                          A plan must be selected

Mali - Projet d'irrigation

Current data set is Analyse financiere
Last plan processed = Projet

Code = 1    Select a plan
Code = 2    Calculating values
Code = 3    Printing quantities/values
Code = 4    Calculating balance/return on a plan
Code = 5    Aggregates and sensitivity analysis
Code = 6    Transfer results to MANIP
Code = 7    End of plan processing

Enter a number between  1 and  7                           If you want results in values,
2                                                          the calculations must be called.
```

```
Values being calculated. Please wait.....

************************
AMOUNT OF LOAN Credit de campagne
IN PLAN Projet                                      A loan cannot finance a benefit.
IS NEGATIVE IN YEAR      1

---NOTE: LOAN AMOUNT MUST ALWAYS BE POSITIVE!!---

LOAN AMOUNT WILL BE ASSUMED AS ZERO FOR YEAR      1
**********************
Press CR key when you want to continue

Mali - Projet d'irrigation

Current data set is Analyse financiere
Last plan processed = Projet

Code = 1     Select a plan
Code = 2     Calculating values
Code = 3     Printing quantities/values
Code = 4     Calculating balance/return on a plan
Code = 5     Aggregates and sensitivity analysis
Code = 6     Transfer results to MANIP
Code = 7     End of plan processing

Enter a number between  1 and   7                   Printing all results to the
3                                                   outputfile (see copy in
                                                    appendix 3)
Mali - Projet d'irrigation

Current data set is Analyse financiere
Last plan processed = Projet

PRINTING RESULTS OF PLAN CALCULATIONS

Code = 1 --- COMMODITIES
Code = 2 --- INVESTMENTS
Code = 3 --- LOANS
Code = 4 --- Full printout
Code = 5 --- Set nbr of dec. digits (current=1)
Code = 6 --- End of printing

Enter a number between  1 and   6
4

You have the following options :
Code = 1     Terminal CRT
Code = 2     Line-printer (132 chrs/l)
Code = 3     HP-printer (227 chrs/l)
Code = 4     Outputfile (227 chrs/l)

Enter a number between  1 and   4
4

Mali - Projet d'irrigation

Current data set is Analyse financiere
Last plan processed = Projet

PRINTING RESULTS OF PLAN CALCULATIONS

Code = 1 --- COMMODITIES
Code = 2 --- INVESTMENTS
Code = 3 --- LOANS
Code = 4 --- Full printout
Code = 5 --- Set nbr of dec. digits (current=1)
Code = 6 --- End of printing
```

```
8

Enter a number between  1 and   6
6

Mali - Projet d'irrigation

Current data set is Analyse financiere
Last plan processed = Projet

Code = 1     Select a plan
Code = 2     Calculating values
Code = 3     Printing quantities/values
Code = 4     Calculating balance/return on a plan
Code = 5     Aggregates and sensitivity analysis
Code = 6     Transfer results to MANIP
Code = 7     End of plan processing

Enter a number between  1 and   7
4

You have the following options :
Code = 1     Terminal CRT
Code = 2     Line-printer (132 chrs/l)
Code = 3     HP-printer (227 chrs/l)
Code = 4     Outputfile (227 chrs/l)

Enter a number between  1 and   4
4

Mali - Projet d'irrigation

Current data set is Analyse financiere
Last plan processed = Projet

Code = 1     Select a plan
Code = 2     Calculating values
Code = 3     Printing quantities/values
Code = 4     Calculating balance/return on a plan
Code = 5     Aggregates and sensitivity analysis
Code = 6     Transfer results to MANIP
Code = 7     End of plan processing

Enter a number between  1 and   7
4

You have the following options :
Code = 1     Terminal CRT
Code = 2     Line-printer (132 chrs/l)
Code = 3     HP-printer (227 chrs/l)
Code = 4     Outputfile (227 chrs/l)

Enter a number between  1 and   4
1

Last plan processed = Projet

                  Mali - Projet d'irrigation
                     Analyse financiere
                      Currency : FCFA

                 TOTAL BALANCE OVER LIFE OF PLAN
=================================================================
21-10-1983
             Year          1           2           3           4
-----------------------------------------------------------------
Projet  Perimetre -448420000.2 -517381219.7 -118271599.9  83582220.3
=================================================================
Press CR key when you want to continue
```

Calculating balance/rate of return on the current plan, and printing results first to outputfile and second to the terminal.

On the terminal, tables are 80 characters per line

```
Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

TOTAL BALANCE OVER LIFE OF PLAN
=================================================================
21-10-1983
          Year          5           6           7         8-19
-----------------------------------------------------------------
Projet  Perimetre  151185115.4  233195000.4  249260000.4  247270000.4
=================================================================
Press CR key when you want to continue

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

TOTAL BALANCE OVER LIFE OF PLAN
===========================
21-10-1983
          Year          20
---------------------------
Projet  Perimetre  970292001.5
===========================
Press CR key when you want to continue

Rate of return = 15.3

Press CR key when you want to continue

Mali - Projet d'irrigation

Current data set is Analyse financiere
Last plan processed = Projet

Code = 1     Select a plan
Code = 2     Calculating values
Code = 3     Printing quantities/values
Code = 4     Calculating balance/return on a plan
Code = 5     Aggregates and sensitivity analysis
Code = 6     Transfer results to MANIP
Code = 7     End of plan processing

Enter a number between  1 and   7
5
```

Calculation of aggregates
(see results in appendix 3).

```
Mali - Projet d'irrigation
Analyse financiere

Last plan processed = Projet

Code = 1     Print a specified aggregate
Code = 2     Print all aggregates
Code = 3     End of aggregate calculation/sensitivity analysis

Enter a number between  1 and   3
2

AGGREGATES being calculated. Please wait . . .
You have the following options :
Code = 1     Terminal CRT
Code = 2     Line-printer (132 chrs/l)
Code = 3     HP-printer (227 chrs/l)
Code = 4     Outputfile (227 chrs/l)

Enter a number between  1 and   4
4
```

```
10

Code = 1 --- Results based on QUANTITIES
Code = 2 --- Results based on VALUES
Code = 3 --- Results based on QUANTITIES AND VALUES

Enter a number between  1 and   3
3

Would you like to rate of return and switching values ?
N

Mali - Projet d'irrigation
Analyse financiere

Last plan processed = Projet

Code = 1     Print a specified aggregate
Code = 2     Print all aggregates
Code = 3     End of aggregate calculation/sensitivity analysis

Enter a number between  1 and   3
1

            AGGREGATES

 12 Cereales                     tonnes     13 PERIMETRE               U

Index number of aggregate to be printed ?
Enter     11 if none

Enter a number between 11 and  13
13

PERIMETRE                     ---  Is it OK ?
Y

Code = 1 --- Results based on QUANTITIES
Code = 2 --- Results based on VALUES
Code = 3 --- Results based on QUANTITIES AND VALUES

Enter a number between  1 and   3
2

Would you like rate of return and switching values ?
Y

Mali - Projet d'irrigation
Analyse financiere

Last plan processed = Projet

Code = 1     Print a specified aggregate
Code = 2     Print all aggregates
Code = 3     End of aggregate calculation/sensitivity analysis

Enter a number between  1 and   3
3
Mali - Projet d'irrigation

Current data set is Analyse financiere
Last plan processed = Projet

Code = 1     Select a plan
Code = 2     Calculating values
Code = 3     Printing quantities/values
Code = 4     Calculating balance/return on a plan
Code = 5     Aggregates and sensitivity analysis
Code = 6     Transfer results to MANIP
Code = 7     End of plan processing
```

```
Enter a number between  1 and  7
6

Mali - Projet d'irrigation

Current data set is Analyse financiere
Last plan processed = Projet

Transfer results to MANIP

Code = 1   Use files with same generic name (MALI)
Code = 2   Use files with a different generic name (current = **************)
Code = 3   Select a different generic name
Code = 4   End of transfer to MANIP

Enter a number between  1 and  4
1

Mali - Projet d'irrigation

Current data set is Analyse financiere
Last plan processed = Projet

Selection of results to transfer to MANIP files MALI

Code = 1    COMMODITIES
Code = 2    INVESTMENTS
Code = 3    LOANS
Code = 4    AGGREGATES
Code = 5    PLAN BALANCE
Code = 6    All results
Code = 7    Specify result type (current = VALUES)
Code = 8    End of transfer to MANIP

Enter a number between  1 and  8
7

Code = 1 --- Results based on QUANTITIES
Code = 2 --- Results based on VALUES
Code = 3 --- Results based on QUANTITIES AND VALUES

Enter a number between  1 and  3
3

Mali - Projet d'irrigation

Current data set is Analyse financiere
Last plan processed = Projet

Selection of results to transfer to MANIP files MALI

Code = 1    COMMODITIES
Code = 2    INVESTMENTS
Code = 3    LOANS
Code = 4    AGGREGATES
Code = 5    PLAN BALANCE
Code = 6    All results
Code = 7    Specify result type (current = QUANT & VAL)
Code = 8    End of transfer to MANIP

Enter a number between  1 and  8
6
```

Transfer of all results to files with same generic name where data can be further used with the MANIP programme.

```
Mali - Projet d'irrigation

Current data set is Analyse financiere
Last plan processed = Projet

Selection of location of transfer to MANIP

Code = 1    Append to existing data
Code = 2    Write from a specific variable number
Code = 3    List variables in destination file
Code = 4    Variable names to be copied over (current = YES)
Code = 5    Exit from transfer to MANIP

Enter a number between  1 and   5
2

******************
AGGREGATE PERIMETRE
CAN ONLY BE SPECIFIED IN VALUES!!!
REQUESTS FOR QUANTITIES WILL BE IGNORED
******************
Press CR key when you want to continue

******************
LOANS CAN ONLY BE SPECIFIED IN VALUES!!!
NOTE : ALL REQUESTS FOR LOAN QUANTITIES WILL BE IGNORED
******************
Press CR key when you want to continue

Enter a number between  1 and 1474
1

A block of   27 variables  will be read and added to your
current MANIP dataset

Press CR key when you want to continue

Transfer to MANIP in progress. Please wait . . .

******************
PLAN BALANCE CAN ONLY BE SPECIFIED IN VALUES!!!
NOTE : REQUEST FOR PLAN BALANCE QUANTITIES WILL BE IGNORED
******************
Press CR key when you want to continue

Mali - Projet d'irrigation

Current data set is Analyse financiere
Last plan processed = Projet

Selection of results to transfer to MANIP files MALI

Code = 1    COMMODITIES
Code = 2    INVESTMENTS
Code = 3    LOANS
Code = 4    AGGREGATES
Code = 5    PLAN BALANCE
Code = 6    All results
Code = 7    Specify result type (current = QUANT & VAL)
Code = 8    End of transfer to MANIP

Enter a number between  1 and   8
8
```

The programme will display suitable messages when there are inconsistencies.

```
Mali - Projet d'irrigation

Current data set is Analyse financiere
Last plan processed = Projet

Code = 1      Select a plan
Code = 2      Calculating values
Code = 3      Printing quantities/values
Code = 4      Calculating balance/return on a plan
Code = 5      Aggregates and sensitivity analysis
Code = 6      Transfer results to MANIP
Code = 7      End of plan processing

Enter a number between  1 and  7
7

Mali - Projet d'irrigation

Current data set is Analyse financiere

Code = 1      Creating a new data base
Code = 2      Listing the data base
Code = 3      Processing a PLAN
Code = 4      Editing the data base
Code = 5      End of job

Enter a number between  1 and  5
5

Gosh, what a job !!
But I wont hesitate to work for you some other time
```

```
Mali - Projet d'irrigation
Analyse financiere
FCFA
1 20
12.0
*******************************************
COMMODITIES
Riz                        tonnes
A 1 180000
Mais                       tonnes
A 1 230000
Intrants                   1,000 FCFA
A 1 1000
Travail                    homme-jour
A 1 700
Pompage                    m3
A 1 5
Encadrement                millions
A 1 1000000
ENDATA
********************************************
ACTIVITIES
Riz irrigue                 Ha
Riz                        P
A 4 2 2.5 3 3.5
Intrants                   C
A 4 35 40 45 52
Travail                    C
A 1 118
END

Riz flottant               Ha
Riz                        P
A 3 1 1.5 2
Intrants                   C
A 3 40 50 60
Travail                    C
A 1 98
END

Riz sans projet            Ha
Riz                        P
A 1 0.5
Intrants                   C
A 1 11.4
Travail                    C
A 1 54
END

Mais (projet)              Ha
Mais                       P
A 3 1 1.5 2
Intrants                   C
A 3 20 30 35
Travail                    C
A 1 118
END
ENDATA
*********************************
INVESTMENTS
Amenagement                 millions
A 1 1000000
50 3 5 10 0
ENDATA
*********************************
```

```
AGGREGATES
Cereales                          tonnes
QUANT
Riz                                 1
Mais                                1
END
PERIMETRE                         U
VALUES
Riz                               1
Mais                              1
Amenagement                       1
Intrants                          1
Encadrement                       1
Pompage                           1
END
ENDATA
**********************************
TRANSFERS
Eau de pompage                    m3
SURPWITHOUT
Riz                               Pompage
ENDATA
***********************************
LOANS
Credit de campagne                fcfa
0     1     13.5
PAID                              EQUAL.INST
Intrants                          SEPARATE
ENDATA
***********************************
PLANS

Riz dresse                         Perimetre
Riz irrigue                        PT
A   4    0    100    250           450
Eau de pompage                     TR
A   1    -1200
END

Riz bas-fonds                      Perimetre
Riz flottant                       PT
A   4    0    75    150            250
Eau de pompage                     TR
A   1    -2000
END

Projet                             Perimetre
Riz dresse                         A
A   1    1
Riz bas-fonds                      A
A   1    1
Mais (projet)                      PT
A   5    0    20    60    120    150
Credit de campagne                 A
A   6    0.60   0.60  ,  0.50    0.40    0.30    0.0
Riz sans projet                    A
A   1    -200
Encadrement                        A
A   1    -4
Amenagement                        A
A   6    376.6    486.4    177.7    64.7    39.8    0
END
ENDATA
****************************************
****************************************
```

```
FORMAT FOR INPUT :
================

Headings :
  1st line : Project name (max. 50 char.)
  2nd line : Data set name (max. 30 char.)
  3rd line : Currency name (max. 10 char.)
  4th line : Base year (0 or 1) and project LIFE (max. 50 years)
  5th line : Opportunity cost of capital (in percentage)

Data :
set TAB to 30
FOR ALL TYPES :
  1st line : NAME (up to 25 char.) - TAB - UNIT (up to 10 char.)

COMMODITIES : 2nd line : Time series of PRICES

ACTIVITIES : For each commodity in the activity : 2 lines
               1st line : NAME - TAB - TYPE
               2nd line : Time series of COEFFICIENTS
                END on a line at the end of the list of a given commodity
  Possible types : C (for consumed), P (for produced)

INVESTMENTS : 2rd line : Time series of unit prices
                3nd line : 5 numbers : INV.LIFE, TIME LAG FOR MAINTENANCE, % for
                  MAINTENANCE COSTS, % of CONTINGENCIES, % for RESIDUAL VALUE

AGGREGATES : 2 nd line : VALUES or QUANT
                1 line for each item in the aggregate :  NAME - TAB - WEIGHT
                 Item can be commodity, investment, aggregate or loan.
                 Number of items is limited to 30 per aggregate.
                END on a line at the end of the list of a given aggregate

TRANSFERS : 2st line : TYPE
               3nd line : NAME of SOURCE - TAB - NAME of TIANSFER COMM.
  Possible types : DEFBACK, DEFWITHOUT, SURPBACK, SURPWITHOUT
  If type is DEFWITHOUT or SURPWITHOUT, source can be a commodity or an investment.
  If type is DEFBACK or SURPBACK, source MUST be a commodity.

LOANS  : 2nd line : 3 numbers : GRACE PERIOD, REPAYMENT PERIOD, INTEREST RATE%
           3rd line : PAID/NOTPAID - TAB - EQUAL.INST/CONST.CAPIT
           4th line : Name of item with LOAN AMOUNTS - TAB - SINGLE/SEPERATE
                Item can be commodity, investment, aggregate of commodities and/or
                investments.

PLANS : For each item in the plan : 2 lines
              1st line : NAME - TAB - TYPE
              2nd line : Time series of COEFFICIENTS
            END on a line at the end of the list for a given plan
  Possible types : A (for annual calc),
                     PT (for phasing calc - nbr of units is total )
                     PI (for phasing calc - nbr of units is incremental )
                     TR if item is a transfer
                     PT and PI can be used only for activities and plans.
  Possible items : anything but an aggregate.

TIME SERIES
==========
On the same line :
 - Type : A(ctual), L(inear interpolation), C(ompound rate)
 - Numbers, depending on type :
A : 1st number : year when values get constant - next : list of values
Number of values MUST be equal to - 1st number if BASE=1
                                   - 1st number + 1 if BASE=0
L, C : 1st number : Value in base year (0 or 1)
       2nd and 3rd numbers : First and last year of change
       4th number : if L : Value in last years
                    if C : Compound rate in % (negative if decrease)
To separate numbers, use only SPACEs (i.e. blanck characters)
(*******************************************************************)
*******************************************************************
```

4

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

COMMODITIES

1 Riz	tonnes	2 Mais	tonnes
3 Intrants	1,000 FCFA	4 Travail	homme-jour
5 Pompage	m3	6 Encadrement	millions

ACTIVITIES

7 : Riz irrigue - Ha
1 COMM P Riz 2 COMM C Intrants
3 COMM C Travail

8 : Riz flottant - Ha
1 COMM P Riz 2 COMM C Intrants
3 COMM C Travail

9 : Riz sans projet - Ha
1 COMM P Riz 2 COMM C Intrants
3 COMM C Travail

10 : Mais (projet) - Ha
1 COMM P Mais 2 COMM C Intrants
3 COMM C Travail

INVESTMENTS

11 Amenagement millions

AGGREGATES

12 Cereales tonnes 13 PERIMETRE U

TRANSFERS

14 Eau de pompage m3

LOANS

15 Credit de campagne fcfa

PLANS

16 : Riz dresse - Perimetre
1 ACTI PT Riz irrigue 2 TRAN TR Eau de pompage

17 : Riz bas-fonds - Perimetre
1 ACTI PT Riz flottant 2 TRAN TR Eau de pompage

18 : Projet - Perimetre
1 PLAN A Riz dresse 2 PLAN A Riz bas-fonds
3 ACTI PT Mais (projet) 4 LOAN A Credit de campagne
5 ACTI A Riz sans projet 6 COMM A Encadrement
7 INVE A Amenagement

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

PRICES OF COMMODITIES

7-10-1983

	Year	1-20
Riz	tonnes	180000.0
Mais	tonnes	230000.0
Intrants	1.000 FCFA	1000.0
Travail	homme-jour	700.0
Pompage	m3	5.0
Encadrement	millions	1000000.0

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

COMPONENTS OF ACTIVITY Riz irrigue

7-10-1983

	Year	1	2	3	4-20
Riz	PROD	2.0	2.5	3.0	3.5
Intrants	CONS	35.0	40.0	45.0	52.0
Travail	CONS	118.0	118.0	118.0	118.0

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

COMPONENTS OF ACTIVITY Riz flottant

7-10-1983

	Year	1	2	3-20
Riz	PROD	1.0	1.5	2.0
Intrants	CONS	40.0	50.0	60.0
Travail	CONS	98.0	98.0	98.0

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

COMPONENTS OF ACTIVITY Riz sans projet

7-10-1983

	Year	1-20
Riz	PROD	0.5
Intrants	CONS	11.4
Travail	CONS	54.0

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

COMPONENTS OF ACTIVITY Mais (projet)

7-10-1983

	Year	1	2	3-20
Mais	PROD	1.0	1.5	2.0
Intrants	CONS	20.0	30.0	35.0
Travail	CONS	118.0	118.0	118.0

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

UNIT PRICES OF INVESTMENTS

7-10-1983

	Year	1-20
Amenagement	millions	1000000.0

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

DATA ON INVESTMENTS

NAME	UNIT	LIFE	%CONTING	%MAINTEN	LAG.MAINT	%RES.VAL
Amenagement	millions	50	10.0	5.0	3	0.0

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

DEFINITION OF AGGREGATES

Cereales tonnes based on QUANT

NBR	NAME	WEIGHT	NBR	NAME	WEIGHT
1	Riz	1.00	2	Mais	1.00

PERIMETRE U based on VALUES

NBR	NAME	WEIGHT	NBR	NAME	WEIGHT
1	Riz	1.00	4	Intrants	1.00
2	Mais	1.00	5	Encadrement	1.00
3	Amenagement	1.00	6	Pompage	1.00

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

TRANSFER ACTIVITIES

NAME	UNIT	SOURCE	TYPE	TRANSFER COMMODITY
Eau de pompage	m3	Riz	SURPWITHOUT	Pompage

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

DATA ON LOANS

NAME	UNIT	GRACE	REPAY	RATE%	IDG	INSTALM	BASE ITEM	SEP/SGL
Credit de campagne	fcfa	0	1	13.50	PAID	EQUAL	Intrants	SEPART

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

COMPONENTS OF PLAN Riz dresse

7-10-1983

	Year	1	2	3	4-20
Riz irrigue	AC-PT	0.0	100.0	250.0	450.0
Eau de pompage	TRANS	-1200.0	-1200.0	-1200.0	-1200.0

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

COMPONENTS OF PLAN Riz bas-fonds

7-10-1983

	Year	1	2	3	4-20
Riz flottant	AC-PT	0.0	75.0	150.0	250.0
Eau de pompage	TRANS	-2000.0	-2000.0	-2000.0	-2000.0

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

COMPONENTS OF PLAN Projet

7-10-1983

	Year	1	2	3	4	5	6-20
Riz dresse	PL-A	1.0	1.0	1.0	1.0	1.0	1.0
Riz bas-fonds	PL-A	1.0	1.0	1.0	1.0	1.0	1.0
Mais (projet)	AC-PT	0.0	20.0	60.0	120.0	150.0	150.0
Credit de campagne	LO-A	0.6	0.6	0.5	0.4	0.3	0.0
Riz sans projet	AC-A	-200.0	-200.0	-200.0	-200.0	-200.0	-200.0
Encadrement	CO-A	-4.0	-4.0	-4.0	-4.0	-4.0	-4.0
Amenagement	IN-A	396.6	486.4	177.7	64.7	39.8	0.0

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

RESULTS OF COMMODITIES CONSUMED FOR Projet (QUANTITIES)

7-10-1983

	Year	1	2	3	4	5	6	7-20
Intrants	1,000 FCFA	-2280.0	4620.0	15120.0	30570.0	36170.0	39820.0	41370.0
Travail	homme-jour	-10800.0	10710.0	40480.0	80960.0	84500.0	84500.0	84500.0

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

RESULTS OF COMMODITIES CONSUMED FOR Projet (VALUES)

7-10-1983

	Year	1	2	3	4	5	6	7-20
Intrants	1,000 FCFA	-2280000.0	4620000.0	15120000.0	30570000.0	36170000.0	39820000.0	41370000.0
Travail	homme-jour	-7560000.0	7497000.0	28336000.0	56672000.0	59150000.0	59150000.0	59150000.0

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

RESULTS OF COMMODITIES PRODUCED FOR Projet (QUANTITIES)

7-10-1983

	Year	1	2	3	4	5	6	7-20
Riz	tonnes	-100.0	175.0	637.5	1337.5	1650.0	1875.0	1975.0
Mais	tonnes	0.0	20.0	70.0	160.0	240.0	285.0	300.0

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

RESULTS OF COMMODITIES PRODUCED FOR Projet (VALUES)

7-10-1983

	Year	1	2	3	4	5	6	7-20
Riz	tonnes	-18000000.0	31500000.0	114750000.1	240749999.9	297000000.1	337500000.0	355500000.0
Mais	tonnes	0.0	4600000.0	16100000.0	36800000.0	55200000.0	65550000.0	69000000.0

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

RESULTS OF COMMODITIES TRANSFERRED FOR Projet (QUANTITIES)

7-10-1983

	Year	1	2	3	4	5	6	7-20
Pompage	m3	0.0	-390000.0	-1035000.0	-2015000.0	-2460000.0	-2770000.0	-2890000.0
Encadrement	millions	-4.0	-4.0	-4.0	-4.0	-4.0	-4.0	-4.0

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

RESULTS OF COMMODITIES TRANSFERRED FOR Projet (VALUES)

7-10-1983

	Year	1	2	3	4	5	6	7-20
Pompage	m3	0.0	-1950000.0	-5175000.0	-10075000.0	-12300000.0	-13850000.0	-14450000.0
Encadrement	millions	-4000000.0	-4000000.0	-4000000.0	-4000000.0	-4000000.0	-4000000.0	-4000000.0

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

RESULTS OF INVESTMENT ACTIVITIES FOR Projet (QUANTITIES)

7-10-1983

	Year	1	2	3	4	5	6-20
Amenagement	millions	396.6	486.4	177.7	64.7	39.8	0.0

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

RESULTS OF INVESTMENT ACTIVITIES FOR Projet (VALUES)

7-10-1983

	Year	1	2	3	4	5	6	7	8-19	20
Amenagement	millions	396600000.0	486399999.8	177699999.9	64700000.0	39800000.0	0.0	0.0	0.0	0.0
REPLACEMENT COSTS		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CONTINGENCIES		39660000.0	48640000.0	17770000.0	6470000.0	3980000.0	0.0	0.0	0.0	0.0
MAINTENANCE COSTS		0.0	0.0	0.0	19830000.0	44150000.0	53035000.0	56270000.0	58260000.0	58260000.0
RESIDUAL VALUE		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	723022801.4

SULTS OF LOAN TRANSACTIONS FOR Projet (VALUES)

7-10-1983

Year		1	2	3	4	5	6-20
Credit de campagne	Fcfa	0.0	2772000.0	7560000.0	12228000.0	10851000.0	0.0
DEBT SERVICE		0.0	3146220.0	8580600.0	13878780.0	12315885.0	0.0
INTEREST COMPONENT		0.0	374220.0	1020600.0	1650780.0	1464885.0	0.0
CAPITAL COMPONENT		0.0	2772000.0	7560000.0	12228000.0	10851000.0	0.0
OUTSTANDING LOAN		0.0	0.0	0.0	0.0	0.0	0.0

Mali - Projet d'irrigation
Analyse financiere

Last plan processed = Projet

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

TOTAL BALANCE OVER LIFE OF PLAN

7-10-1983

Year		1	2	3	4	5	6	7	8-19	20
Projet	Perimetre	-448420000.2	-517381219.7	-118271599.9	83582220.3	151185115.4	233195000.4	249260000.4	247270000.4	970292001.5

Rate of return = 15.3

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

RESULTS OF AGGREGATES FOR PLAN Projet (QUANTITIES)

7-10-1983

Year		1	2	3	4	5	6	7-20
Cereales	tonnes	-100.0	195.0	707.5	1497.5	1890.0	2160.0	2275.0

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

RESULTS OF AGGREGATES FOR PLAN Projet (VALUES)

11- 1-1984

Year		1	2	3	4	5	6	7	8-19	20
Cereales	tonnes	-18000000.0	36100000.0	130850000.0	277550000.0	352200000.0	403850000.0	424500000.2	424500000.2	424500000.2
PERIMETRE	U	-455980000.0	-509509999.8	-88914999.9	141905000.1	211800000.3	292345000.3	308410000.3	306420000.3	1029442001.5

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

RESULTS OF AGGREGATES FOR PLAN Projet (VALUES)

7-10-1983

Year		1	2	3	4	5	6	7	8-19	20
Cereales	Tonnes	-100.0	195.0	707.5	1497.5	1890.0	2160.0	2275.0	2275.0	2275.0
PERIMETRE	U	-455980000.0	-509509999.8	-88914999.9	141905000.1	211800000.3	292345000.3	308410000.3	306420000.3	1029442001.5

Mali - Projet d'irrigation
Analyse financiere
Currency : FCFA

RESULTS OF AGGREGATES FOR PLAN Projet (VALUES)

7-10-1983

Year		1	2	3	4	5	6	7	8-19	20
PERIMETRE	U	-455980000.0	-509509999.8	-88914999.9	141905000.1	211800000.3	292345000.3	308410000.3	306420000.3	1029442001.5

Rate of return = 19.2

Switching valses for opportsnity cost of capital = 12.0 %

Current AGGREGATE being analysed is PERIMETRE

		PRESENT VALUES	SWITCHING VALUES PER CENT
ITEM	Riz	1777013616.07	-33.0
ITEM	Mais	334749954.84	-175.3
ITEM	Amenagement	-1209560992.66	48.5
ITEM	Intrants	-211457067.82	277.5
ITEM	Encadrement	-29877774.38	1963.7
ITEM	Pompage	-74160651.09	791.1
WITHOUT	PROJECT BALANCE	0.00	
	NET BALANCE	586707085.18	

MALI

SITUATION ET PERSPECTIVES DE L'ELEVAGE

STRATIFICATION DE LA PRODUCTION DE VIANDE - MODELE DE TROUPEAU SAHELIEN (1)

I. INTRODUCTION - LE MODELE

Afin de tester les effets de la transformation d'un troupeau sahélien en troupeau naisseur, un modèle dynamique de troupeau a été construit en utilisant un programme de calcul disponible sur l'ordinateur (modèle HP9845B) du Centre d'investissement de la FAO.

Le troupeau est divisé en 11 classes d'âge. Les mâles reproducteurs sont traités à part, leur nombre est déterminé par un rapport fixe avec le nombre de femelles, et ajusté par des achats et/ou des ventes (équivalent à un transfert à l'intérieur du troupeau si le prix est le même que celui des animaux d'élevage).

Les caractéristiques du troupeau sont empruntées à un rapport de l'IER sur la 5ème région et correspondant à un troupeau transhumant de 1.000 têtes originaire de la zone sèche.

On a supposé que tous les paramètres techniques restaient constants en l'absence de tout projet, sauf le taux de réforme des femelles de plus de 3 ans, qui doit augmenter afin de stabiliser la taille du troupeau lorsque celui-ci astteint la limite fixée à 1.100 unités animales, soit 25% de plus que la taille actuelle (les unités animales sont calculées en affectant un poids 0,3 au nombre de veaux et 0,6 aux animaux 1-2 ans).

Les caractéristiques du modèle sont résumées au Tableau 1. Les projections sont présentées au Tableau 2, et la valeur correspondante des ventes au Tableau 3.

II. CHANGEMENT DU MODE D'EXPLOITATION

Tout le reste étant constant, on a changé le taux d'exploitation des mâles, de façon progressive sur une période de six ans, avec les valeurs suivantes (en %):

	Années						
	0	1	2	3	4	5	6
Mâles 1-2 ans	12	20	30	50	70	90	100
Mâles 2-3 ans	8	20	30	50	70	90	100
Mâles 3-5 ans	20	30	50	70	90	100	100
Mâles 3-5 ans	30	50	70	90	100	100	100

Les effets sur la structure et la production du troupeau sont montrés par le Tableau 4.

Entre les deux situations, le nombre de vaches de plus de trois ans passe de 34% à 44% du nombre d'animaux et de 42% à 59% du nombre d'unités animales. Le taux d'exploitation stabilisé passe de 11% à 15%.

(1) Reproduced from Ref. 46.

La valeur de la production du troupeau ainsi transformé est calculée dans le Tableau 5. La part du lait dans le revenu total passe de 34% à 40%.

L'effet net de ces changements (différence entre les deux situations) est montré par les Tableaux 6 et 7. Le revenu du troupeau est augmenté pendant les années 1 à 4, mais est ensuite inférieur pendant 6 ans, avant de remonter à nouveau.

Une fois la croissance terminée et les troupeaux stabilisés, la transformation en troupeaux naisseurs prend tout son sens et le modèle montre que le revenu est augmenté de 18% par rapport au système de gestion antérieure.

La valeur actualisée de la production sur 20 ans n'augmente que de 8 à 9% (selon le taux d'actualisation), ce qui, vu les variations de revenu, n'est pas très attrayant.

Si la transformation en troupeau naisseur est plus rapide (3 ans au lieu de 6), les résultats sont pires:

- Baisse du revenu des années 1 à 10.

- Augmentation de la valeur actualisée de la production (à 12%) de 7,3%. Si la transformation en troupeau naisseur est plus lente (10 au lieu de 6), la baisse de revenu est beaucoup moins marquée et dure 5 ans (des années 5 à 9), mais l'augmentation de production (valeur actualisée) n'est toujours que de 9%, comme le montrent les Tableaux 8 à 11.

III. CHANGEMENT DU PRIX DES JEUNES MALES

On a combiné au changement de mode d'exploitation exposé ci-dessus un changement de prix des mâles de 1 à 2 ans, le faisant passer de 30.000 FM/tête à 32.000 en année 2, 35.000 en année 3, 38.000 en année 4 et 40.000 à partir de l'année 5, soit une augmentation d'un tiers.

Dans l'hypothèse de base (changement des taux de réforme en six ans), le changement parallèle de prix des jeunes mâles limite la baisse de revenu aux années 5 à 9, et fait passer l'augmentation de revenu sur 20 ans à 14% (taux d'actualisation de 12% ou de 20%).

La combinaison d'un changement de taux de réforme étalé sur 10 ans avec une augmentation du prix des jeunes mâles donne la projection de revenu montrée par le Tableau 12 qui fait presque totalement disparaître la baisse transitoire de revenu due à la transformation du troupeau (elle persiste en année 6), comme le montre le Tableau 13.

L'augmentation de revenu sur 20 ans est dans ce cas de 13 à 14%. Et si l'on compare le revenu, une fois la croissance achevée et les troupeaux en équilibre, le troupeau naisseur donne un revenu augmenté de 27% par rapport au système de gestion antérieure.

MALI

SITUATION ET PERSPECTIVES DE L'ELEVAGE

Modèle de Troupeau Sahélien
Caractéristiques présentes

Age	Composition en année zéro		Taux de réforme (%)		Mortalité (%)		Prix de vente (milliers de FM/tête)	
	Mâles	Femelles	Mâles	Femelles	Mâles	Femelles	Mâles	Femelles
0-1	74	86	-	-	30	21	12	15
1-2	55	75	12	3	13	13	30	35
2-3	43	67	8	1	)		45	50
3-4	37	)	20	)	)		55	)
4-5	29	) 442	20	) 10	) 3		65	) 55
> 5	70	)	30	)	)		75	)
Mâles reproducteurs	22	-	10	-	)		75	-

Taux de fécondité : 59 %

Age du premier vélage : > 3 ans

Age de la réforme des femelles excédentaires : > 3 ans

% des femelles exploitées pour le lait : 50

Rapport mâles reproducteurs/femelles : 5 %

Capacité de charge maximum : 1 100 U.A.

Vente de lait : 20 000 FM/femelle exploitée/an

ANNEXE 6
Tableau 1

MALI - SITUATION ET PERSPECTIVES DE L'ELEVAGE TROUPEAU NAISSEUR

ANNEXE 6
Tableau 2
page 1

HERD DEVELOPMENT COMPOSITION — WITHOUT PROJECT SITUATION

YEAR NO OF ANIMALS (BEG. OF YR)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0-1 YR MALE	74	133	134	133	141	149	144	139	138	130	137	137	137	137	137	137	137	137	137	137	137
0-1 YR FEMALE	06	133	134	133	141	149	144	139	130	138	137	137	137	137	137	137	137	137	137	137	137
1-2 YR MALE	55	52	93	94	93	99	104	101	97	97	96	96	96	96	96	96	96	96	96	96	96
1-2 YR FEMALE	75	68	105	106	105	112	117	114	110	109	109	107	109	109	109	109	107	107	107	107	107
2-3 YR MALE	43	42	40	71	72	71	76	80	77	75	74	74	74	74	74	74	74	74	74	74	74
2-3 YR FEMALE	67	63	57	09	89	09	94	99	96	93	92	92	92	92	92	92	92	92	92	92	92
3-4 YR MALE	37	30	38	35	64	64	64	68	71	69	67	66	66	66	64	66	66	66	66	66	66
4-5 YR MALE	29	27	30	27	27	49	50	49	52	55	53	52	51	51	51	51	51	51	51	51	51
OVER 5 YR MALE	70	70	70	71	71	69	85	96	104	111	118	122	123	123	123	123	123	123	123	123	123
OVER 3 YR FEM.	442	450	454	451	479	504	400	472	469	467	466	466	466	466	466	466	466	466	466	466	466
BREEDING MALE	22	22	23	23	24	25	24	24	23	23	23	23	23	23	23	23	23	23	23	23	23
TOTAL AN. UNIT	836	667	910	968	1029	1087	1100	1100	1100	1099	1100	1100	1100	1100	1100	1100	1100	1100	1100	1100	1100
TOT. HERD SIZE	1000	1101	1176	1234	1306	1379	1390	1301	1377	1375	1374	1374	1374	1374	1375	1375	1374	1374	1374	1374	1374
BREEDING FEM.	442	450	454	451	479	504	400	472	469	467	466	466	466	466	466	466	466	466	466	465	466
MILK. FEMALE	221	225	227	226	239	252	244	236	235	234	233	233	233	233	233	233	233	233	233	233	233

MORTALITIES - WITHOUT PROJECT SITUATION

YEAR	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0-1 YR MALE	22	40	40	40	42	45	43	42	42	41	41	41	41	41	41	41	41	41	41	41	41
0-1 YR FEMALE	18	20	28	20	30	31	30	29	29	27	29	27	29	29	27	29	27	29	29	29	29
1-2 YR MALE	7	7	12	12	12	13	14	13	13	13	13	13	13	13	13	13	13	13	13	13	13
1-2 YR FEMALE	10	9	14	14	14	15	15	15	14	14	14	14	14	14	14	14	14	14	14	14	14
2-3 YR MALE	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2-3 YR FEMALE	2	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3-4 YR MALE	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
4-5 YR MALE	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2
OVER 5 YR MALE	2	2	2	2	2	2	3	3	3	3	4	4	4	4	4	4	4	4	4	4	4
OVER 3 YR FEM.	13	14	14	14	14	15	15	14	13	14	14	14	14	14	14	14	14	14	14	14	14
BREEDING MALE	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
TOT. MORTALITY	78	105	115	117	123	129	129	126	124	124	124	124	124	124	124	124	124	124	124	124	124
MORTALITY RATE	8	10	10	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9

CULLING-SALES - WITHOUT PROJECT SITUATION

YEAR	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0-1 YR MALE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0-1 YR FEMALE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1-2 YR MALE	6	5	10	10	10	10	11	11	10	10	10	10	10	10	10	10	10	10	10	16	10
1-2 YR FEMALE	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2-3 YR MALE	3	3	3	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	5	[illegible]
2-3 YR FEMALE	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3-4 YR MALE	7	7	7	7	12	12	12	13	14	13	13	13	13	13	13	13	13	13	13	13	[illegible]
4-5 YR MALE	6	6	6	6	5	10	10	10	10	11	10	10	10	10	10	10	10	10	10	10	13
OVER 5 YR MALE	20	20	20	21	21	20	25	28	30	32	34	35	36	36	36	36	35	36	36	35	[illegible]
OVER 3 YR FEM.	43	44	44	44	46	86	92	84	80	76	75	74	74	74	74	74	74	74	74	74	74
BREEDING MALE	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
TOT. CULLING	90	90	96	98	106	150	162	158	157	154	154	154	154	154	154	154	154	154	154	154	154
OFF-TAKE RATE	9	0	8	0	8	11	12	11	11	11	11	11	11	11	11	11	11	11	11	11	11

ANNEXE 6
Tableau 2
page 2

PURCH. OF BR. MALE - WITHOUT PROJECT SITUATION

YEAR	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	3	3	3	4	4	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3

MALI

SITUATION ET PERSPECTIVES DE L'ELEVAGE

Annex 6
Tableau 3

VALEUR DE LA PRODUCTION

ECONOMIC/FINANCIAL ANALYSIS - WITHOUT PROJECT SITUATION

YEAR	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0	19.0	20.0
SALES OF AN. — WITHOUT PROJECT SITUATION																					
0-1 YR MALE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0-1 YR FEMALE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1-2 YR MALE	172.3	162.2	291.2	293.5	291.0	309.8	325.9	315.6	305.3	303.3	302.0	301.4	301.4	301.4	301.4	301.4	301.4	301.4	301.4	301.4	301.4
1-2 YR FEMALE	68.5	62.1	95.8	96.6	96.1	102.0	107.3	103.9	100.5	99.8	99.4	99.2	99.2	99.2	99.2	99.2	99.2	99.2	99.2	99.2	99.2
2-3 YR MALE	150.2	147.0	138.5	248.5	250.5	249.1	264.4	278.2	269.4	260.6	258.9	257.8	257.3	257.3	257.3	257.3	257.3	257.3	257.3	257.3	257.3
2-3 YR FEMALE	32.5	38.7	27.8	42.9	43.3	43.0	45.7	48.1	46.5	45.0	44.7	44.5	44.4	44.4	44.4	44.4	44.4	44.4	44.4	44.4	44.4
3-4 YR MALE	394.0	407.4	400.9	377.6	677.7	603.2	679.3	721.0	750.5	734.6	710.5	706.0	703.0	701.5	701.5	701.5	701.5	701.5	701.5	701.5	701.5
4-5 YR MALE	365.7	362.1	375.5	367.7	346.3	621.5	626.5	623.0	661.2	695.6	673.7	651.6	647.5	644.7	643.3	643.3	643.3	643.3	643.3	643.3	643.3
OVER 5 YR MALE	1527.0	1520.5	1524.1	1539.2	1539.0	1510.1	1060.1	2104.5	2265.6	2426.5	2501.0	2657.9	2679.9	2609.3	2691.9	2691.9	2691.9	2591.8	2691.8	2691.4	2671.8
OVER 3 YR FEM.	2358.1	2401.9	2421.1	2407.3	2555.2	4726.9	5049.9	4619.0	4405.2	4107.4	4102.0	4002.1	4071.7	4071.7	4071.7	4071.7	4071.7	4071.7	4071.7	4071.7	4071.7
BREEDING MALE	160.1	163.8	165.1	164.1	174.2	183.3	177.5	171.7	170.6	169.9	169.5	169.5	169.5	169.5	169.5	169.5	169.5	169.5	169.5	169.5	169.5
TOTAL	5229.0	5267.6	5440.6	5537.6	5974.2	8428.8	9136.6	8905.7	8902.9	8922.7	8943.5	8970.1	8973.9	8979.0	8900.3	8908.2	8980.2	8988.2	8980.1	8981.1	8980.1
OTHER INCOMES - WITHOUT PROJECT SITUATION																					
LAIT	4420.0	4502.1	4530.1	4512.4	4789.6	5038.4	4080.0	4720.0	4690.0	4670.0	4660.0	4660.0	4660.0	4660.0	4660.0	4660.0	4660.0	4660.0	4660.0	4660.0	4660.0
TOTAL	4420.0	4502.1	4538.1	4512.4	4789.6	5038.4	4080.0	4720.0	4690.0	4670.0	4660.0	4660.0	4660.0	4660.0	4660.0	4660.0	4660.0	4560.0	4660.0	4660.0	4660.0

INCREMENTAL VALUE OF HERD AT YEAR 20.0 : 14989 - WITHOUT PROJECT SITUATION

SITUATION ET PERSPECTIVES DE L'ELEVAGE

ANNEXE 6
Tableau 4
page 1

TROUPEAU NAISSEUR 2
(Changement sur 6 ans)

HERD DEVELOPMENT COMPOSITION — WITH PROJECT SITUATION

YEAR	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
NO OF ANIMALS (BEG. OF YR)																					
0-1 YR MALE	74	133	134	133	141	149	155	162	169	177	185	194	193	192	192	192	192	192	192	192	172
0-1 YR FEMALE	86	133	134	133	141	149	155	162	167	177	105	194	193	192	192	172	192	192	192	192	192
1-2 YR MALE	55	52	93	94	93	99	104	108	113	119	124	130	136	135	134	134	134	134	134	134	134
1-2 YR FEMALE	75	60	105	106	105	112	117	122	120	134	140	146	153	153	151	151	151	151	151	151	151
2-3 YR MALE	43	42	36	57	41	24	9	0	0	0	-0	-0	-0	-0	-0	-0	-0	-0	0	-0	0
2-3 YR FEMALE	67	63	57	07	89	09	94	99	103	100	113	110	123	129	129	128	120	120	120	120	120
3-4 YR MALE	37	38	33	24	27	12	2	-0	0	0	0	0	0	0	0	0	0	0	0	0	0
4 5 YR MALE	29	29	26	16	7	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OVER 5 YR MALE	70	70	53	28	7	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OVER 3 YR FEM.	442	450	454	451	479	584	525	549	574	601	628	657	655	650	650	650	650	650	650	650	650
BREEDING MALE	22	23	23	23	24	25	26	27	29	30	31	33	33	33	33	33	33	33	33	33	33
TOTAL AN. UNIT	036	857	801	807	879	873	002	911	953	976	1042	1009	1100	1097	1070	1097	1097	1097	1097	1097	1097
TOT. HERD SIZE	1000	1101	1148	1153	1156	1165	1188	1230	1206	1345	1407	1471	1486	1403	1401	1480	1400	1480	1480	1480	1484
BREEDING FEM.	442	450	454	451	479	584	525	549	574	601	628	657	655	650	650	650	650	550	650	650	650
MILK. FEMALE	221	225	227	226	239	252	263	274	207	300	314	320	320	325	325	325	325	325	325	325	325

MORTALITIES — WITH PROJECT SITUATION

YEAR	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0-1 YR MALE	22	46	40	40	42	45	46	49	51	53	56	50	58	58	58	58	50	58	58	58	58
0-1 YR FEMALE	18	20	20	20	30	31	33	34	36	37	39	41	41	40	40	40	40	40	40	40	40
1-2 YR MALE	7	7	12	12	12	13	14	14	15	15	16	17	18	10	17	17	17	17	17	17	17
1 2 YR FEMALE	10	9	14	14	14	15	15	16	17	17	18	19	20	20	20	20	20	20	20	20	20
2-3 YR MALE	1	1	1	2	1	1	0	0	0	0	-0	0	-0	-0	-0	-0	-0	-0	-0	-0	-0
2 3 YR FEMALE	2	2	2	3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	4	4
3-4 YR MALE	1	1	1	1	1	0	0	-0	0	0	0	0	0	0	0	0	0	0	0	0	0
4-5 YR MALE	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OVER 5 YR MALE	2	2	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OVER 3 YR FEM.	13	14	14	14	14	15	16	16	17	10	19	20	20	20	20	20	20	20	20	20	20
BREEDING MALE	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
TOT. MORTALITY	78	105	114	114	118	123	127	133	139	145	152	159	160	160	159	159	159	159	159	159	159
MORTALITY RATE	0	10	10	10	10	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11

SITUATION ET PERSPECTIVES DE L'ELEVAGE

CULLING-SALES - WITH PROJECT SITUATION

YEAR	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0-1 YR MALE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0-1 YR FEMALE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1-2 YR MALE	6	9	24	41	57	77	91	94	99	103	100	113	110	110	117	117	117	117	117	117	117
1-2 YR FEMALE	2	2	3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	4	4	4
2-3 YR MALE	3	5	10	27	28	21	0	0	0	0	-0	0	-0	-0	-0	0	-0	-0	-0	-0	-0
2-3 YR FEMALE	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3-4 YR MALE	7	11	16	17	24	12	2	-0	0	0	0	0	0	0	0	0	0	0	0	0	0
4-5 YR MALE	6	8	13	11	6	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OVER 5 YR MALE	24	34	36	25	7	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OVER 3 YR FEM.	43	44	44	44	46	49	51	53	56	58	61	95	104	105	104	103	103	103	103	103	103
BREEDING MALE	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
TOT. CULLING	90	117	149	170	174	169	159	154	161	169	177	216	230	231	229	228	228	220	220	220	228
OFF-TAKE RATE	9	11	13	15	15	14	13	13	13	13	13	15	15	16	15	15	15	15	15	15	15

PURCH. OF BR. MALE WITH PROJECT SITUATION

YEAR	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	3	3	3	4	4	4	5	5	5	5	5	4	4	4	4	4	4	4	4	4	4

ANNEXE 6
Tableau 4
page 2

MALI

SITUATION ET PERSPECTIVES DE L'ELEVAGE

VALEUR DE LA PRODUCTION
(changement sur six ans)

ECONOMIC/FINANCIAL ANALYSIS - WITH PROJECT SITUATION

YEAR	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0	19.0	20.0
SALES OF AN. - WITH PROJECT SITUATION																					
0-1 YR MALE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.0	0.0	0.0	0.0
0-1 YR FEMALE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1-2 YR MALE	172.3	270.4	727.9	1222.9	1702.4	2323.3	2715.5	2830.0	2950.1	3095.3	3236.6	3304.3	3539.1	3530.3	3503.3	3503.3	3503.3	3503.3	3503.3	3503.3	3503.3
1-2 YR FEMALE	68.5	62.1	95.8	96.6	96.1	102.0	107.3	111.8	116.8	122.3	127.8	133.7	139.8	139.4	138.4	138.4	138.4	138.4	138.4	138.4	138.4
2-3 YR MALE	150.2	367.6	472.1	1235.7	1245.6	955.4	375.6	0.0	0.0	0.0	-.0	-.0	-.0	-.0	-.0	-.0	-.0	-.0	-.0	-.0	-.0
2-3 YR FEMALE	32.5	38.7	27.8	42.9	43.3	43.0	45.7	48.1	50.1	52.4	54.8	57.3	59.9	62.6	62.5	62.0	62.0	62.0	62.0	62.0	62.1
3-4 YR MALE	374.8	614.2	871.6	914.2	1318.5	632.9	125.9	-.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4-5 YR MALE	365.7	543.1	321.4	699.4	404.2	167.9	.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
OVER 5 YR MALE	1527.8	2547.5	2722.6	1846.0	534.5	50.3	.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
OVER 3 YR FEM.	2358.1	2481.9	2421.1	2407.3	2555.2	2688.0	2801.3	2928.1	3063.9	3203.0	3350.0	5244.5	5716.7	5740.3	5731.3	5679.4	5679.4	5679.4	5679.4	5679.4	5679.4
BREEDING MALE	150.1	163.8	165.1	164.1	174.2	183.3	191.0	199.6	208.9	218.4	228.4	230.9	230.3	236.4	236.4	236.4	236.4	236.4	236.4	236.4	236.4
TOTAL	5229.8	7881.1	8325.4	8629.3	8073.9	7146.0	6362.2	6117.5	6377.8	6692.2	6997.6	9050.5	9693.7	9717.0	9671.9	9619.5	9617.5	9619.5	9619.5	9619.5	9619.5
OTHER INCOMES - WITH PROJECT SITUATION																					
LAIT	4420.0	4502.1	4530.1	4512.4	4789.6	5038.4	5250.7	5488.4	5743.0	6005.3	6279.2	6566.4	6550.0	6500.0	6500.0	6500.0	6500.0	6500.0	6500.0	6500.0	6500.0
TOTAL	4420.0	4502.1	4538.1	4512.4	4789.6	5028.4	5250.7	5488.4	5743.0	6005.3	6279.2	6566.4	6550.0	6500.0	6500.0	6500.0	6500.0	6500.0	6500.0	6500.0	6500.0

INCREMENTAL VALUE OF HERD AT YEAR 20.0 : 12217 - WITH PROJECT SITUATION

ANNEXE 6
Tableau 5

MALI

SITUATION ET PERSPECTIVES DE . ƎLE .GE

VALEUR DE LA PRODUCTION (changement sur 6 ans)

ECONOMIC/FINANCIAL ANALYSIS

Incremental

YEAR	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	0.0	9.0	10.0
SALES OF AN. - Incremental											
0-1 YR MALE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0-1 YR FEMALE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1-2 YR MALE	0.0	190.2	436.6	929.4	1410.6	2013.5	2309.6	2514.3	2652.8	2792.0	2934.6
1-2 YR FEMALE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.9	16.4	22.4	20.4
2-3 YR MALE	0.0	220.6	333.6	507.1	995.0	706.3	111.2	-278.2	-269.4	-260.6	-258.9
2-3 YR FEMALE	0.0	9.9	0.0	0.0	0.0	0.0	0.0	0.0	3.5	7.3	10.0
3-4 YR MALE	0.0	294.7	470.7	536.6	640.7	-50.3	-553.4	-721.0	-758.5	-734.6	-710.5
4-5 YR MALE	0.0	131.0	445.5	331.7	57.9	-453.6	-626.5	-623.0	-661.2	-675.6	-673.7
OVER 5 YR MALE	0.0	1019.0	1190.4	306.8	-1004.5	-1459.8	-1060.1	-2104.5	-2265.6	-2426.5	-2501.8
OVER 3 YR FEM.	0.6	0.0	0.0	0.0	0.0	-2038.9	-2243.7	-1691.7	-1341.4	-983.6	-752.8
BREEDING MALE	0.0	9.0	0.0	0.0	0.0	0.0	13.5	28.0	30.3	40.6	58.9
TOTAL	0.0	1733.5	2085.4	3091.7	2077.7	-1202.8	-2774.5	-2868.1	-2585.1	-2230.5	-1745.8
OTHER INCOMES	0.0	9.0	0.0	0.0	0.0	0.0	370.7	768.4	1053.0	1335.3	1619.2

YEAR	11.0	12.0	13.0	14.0	15.0	16.0	17.0	10.0	17.0	20.0
SALES OF AN. - Incremental										
0-1 YR MALE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0-1 YR FEMALE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1-2 YR MALE	3002.9	3237.7	3228.9	3201.9	3201.9	3201.9	3201.9	3201.9	3201.9	3201.9
1-2 YR FEMALE	34.5	40.6	40.2	39.2	39.2	39.2	39.2	39.2	37.2	37.2
2-3 YR MALE	-257.0	-257.3	-257.3	-257.3	-257.3	-257.3	-257.3	-257.3	-257.3	-257.3
2-3 YR FEMALE	12.7	15.4	10.2	18.0	17.6	17.6	17.6	17.6	17.6	17.6
3-4 YR MALE	-706.0	-703.0	-701.5	-701.5	-701.5	-701.5	-701.5	-701.5	-701.5	-701.5
4-5 YR MALE	-651.6	-647.5	-644.7	-643.3	-643.3	-643.3	-643.3	-643.3	-643.3	-643.3
OVER 5 YR MALE	-2657.9	-2679.9	-2689.3	-2691.9	-2691.9	-2691.9	-2691.8	2691.8	-2691.8	-2671.8
OVER 3 YR FEM.	1162.4	1645.1	1676.6	1659.7	1607.7	1607.7	1607.7	1607.7	1607.7	1607.7
BREEDING MALE	69.3	68.7	66.9	66.9	66.9	66.9	66.9	66.9	66.9	66.9
TOTAL	80.5	719.8	738.0	671.6	639.2	639.3	639.3	639.3	639.3	639.5
OTHER INCOMES	1906.4	1070.0	1040.0	1040.0	1840.0	1840.0	1840.0	1016.0	1040.0	1840.0

INCREMENTAL VALUE OF HERD AT YEAR 20.0 : -2772 - Incremental

ANNEXE 6
Tableau 6

MALI - SITUATION ET PERSPECTIVES DE L'ELEVAGE

ANNEXE 6
Tableau 7

COSTS/BENEFITS FLOW - Incremental (changement sur 6 ans)

YEARS	TOTAL COSTS	GROSS REV.	BALANCE
0	0	0	0
1	0	1734	1734
2	0	2885	2885
3	0	3092	3092
4	0	2100	2100
5	0	-1283	-1283
6	0	-2404	-2404
7	0	-2100	-2100
8	0	-1532	-1532
9	0	-895	-895
10	0	-327	-327
11	0	1995	1995
12	0	2610	2610
13	0	2570	2570
14	0	2532	2532
15	0	2479	2479
16	0	2479	2479
17	0	2479	2479
18	0	2479	2479
19	0	2479	2479
20	0	-293	-293

NET PRESENT VALUES FOR AN OPPORTUNITY COST OF CAPITAL OF 12.0 %

WITHOUT PROJECT SITUATION : 92781.6
WITH PROJECT SITUATION : 100372.8
INCREMENTAL VALUE : 7591.2
PERCENTAGE CHANGE : 8.2 % OVER WITHOUT PROJECT SITUATION

NET PRESENT VALUES FOR AN OPPORTUNITY COST OF CAPITAL OF 20.0 %

WITHOUT PROJECT SITUATION : 57644.8
WITH PROJECT SITUATION : 62974.9
INCREMENTAL VALUE : 5330.1

ANNEXE 6
Tableau 8

MALI

SITUATION ET PERSPECTIVES DE L'ELEVAGE

Modèle de troupeau sahélien

Changement de taux de réforme des mâles sur dix ans

Classe d'âge	Année 0	1	2	3	4	5	6	7	8	9	10
0-1	0	0	0	0	0	0	0	0	0	0	0
1-2	12	15	20	30	40	50	60	70	80	90	100
2-3	8	15	20	30	40	50	60	70	80	90	100
3-4	20	30	40	50	60	70	80	90	100	100	100
4-5	20	30	40	50	60	70	80	90	100	100	100
> 5	30	40	50	60	70	80	90	100	100	100	100
Mâle reproducteur	10	10	10	10	10	10	10	10	10	10	10

TROUPEAU NAISSEUR 2

(Changement sur 10 ans)

HERD DEVELOPMENT COMPOSITION — WITH PROJECT SITUATION

YEAR NO OF ANIMALS (BEG. OF YR)	0	1	2	3	4	5	6	7	0	9	10	11	12	13	14	15	16	17	18	17	20
0-1 YR MALE	74	13?	134	133	141	149	155	162	169	177	185	174	193	192	172	192	172	192	192	192	192
0-1 YR FEMALE	66	133	134	133	141	149	155	162	167	177	105	174	193	192	172	192	172	172	172	152	192
1-2 YR MALE	55	??	93	94	93	99	104	108	113	119	124	130	136	135	134	134	134	134	134	134	134
1-2 YR FEMALE	75	63	105	106	105	112	117	122	128	134	140	146	153	153	151	151	151	151	151	151	151
2-3 YR MALE	43	42	30	65	57	49	43	36	20	20	10	-	-0	-0	-0	-0	-0	-0	-0	-0	-0
2-3 YR FEMALE	67	43	57	07	89	07	94	99	103	108	113	110	123	129	129	128	128	128	120	128	128
3-4 YR MALE	37	30	35	30	44	33	24	17	11	5	2	-0	0	0	0	0	0	0	0	0	0
4-5 YR MALE	27	27	26	20	14	17	10	5	2	0	-0	0	0	0	0	0	0	0	0	0	0
OVER 5 YR MALE	70	70	60	44	27	13	8	3	0	0	0	0	0	0	0	0	0	0	0	0	0
OVER 3 YR FEM.	442	450	454	451	479	504	525	549	574	601	628	657	655	650	650	650	650	650	650	650	651
BREEDING MALE	22	23	23	23	24	25	26	27	29	30	31	33	33	33	33	33	33	33	33	33	33
TOTAL AN. UNIT	836	067	872	721	938	946	955	971	994	1021	1054	1037	1100	1097	1070	1097	1077	1097	1077	1097	
TOT. HERD SIZE	1000	1101	1159	1107	1216	1238	1261	1290	1327	1370	1419	1471	1486	1403	1481	1480	1480	1480	1400	1480	1430
BREEDING FEM.	442	450	454	451	479	504	525	549	574	601	628	657	655	650	650	650	650	650	650	650	650
MILK. FEMALE	221	225	227	226	239	252	263	274	287	300	314	328	328	325	325	325	325	325	325	325	325

MORTALITIES — WITH PROJECT SITUATION

YEAR	0	1	2	3	4	5	6	7	0	9	10	11	12	13	14	15	16	17	10	17	20
0-1 YR MALE	22	40	40	40	42	45	46	49	51	53	56	50	58	50	58	58	58	58	59	58	50
0-1 YR FEMALE	18	20	28	28	39	31	33	34	36	37	39	41	41	40	40	40	40	40	49	40	40
1-2 YR MALE	7	7	12	12	12	13	14	14	15	15	16	17	18	10	17	17	17	17	17	17	17
1-2 YR FEMALE	10	9	14	14	14	15	15	16	17	17	18	17	20	29	20	20	20	20	20	20	20
2-3 YR MALE	1	1	1	2	2	1	1	1	1	1	0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
2-3 YR FEMALE	2	2	2	3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	4	4
3-4 YR MALE	1	1	1	1	1	1	1	1	0	0	0	-0	0	0	0	0	0	0	0	0	0
4-5 YR MALE	1	1	1	1	0	1	0	0	0	0	-0	0	0	0	0	0	0	0	0	0	0
OVER 5 YR MALE	2	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
OVER 3 YR FEM.	13	14	14	14	14	15	16	16	17	18	19	20	20	20	20	20	20	20	20	20	20
BREEDING MALE	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
TOT. MORTALITY	78	105	115	115	120	125	130	135	140	146	152	159	160	160	159	159	159	159	159	159	159
MORTALITY RATE	0	10	10	10	10	10	10	10	11	11	11	11	11	11	11	11	11	11	11	11	11

ANNEXE 6
Tableau 9
page 1

CULLING-SALES - WITH PROJECT SITUATION

YEAR	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0-1 YR MALE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0-1 YR FEMALE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1-2 YR MALE	5	7	16	24	32	43	54	66	79	93	108	113	118	110	117	117	117	117	117	117	117
1-2 YR FEMALE	2	2	3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	4	4	4
2-3 YR MALE	3	6	7	17	22	24	25	25	22	17	10	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
2-3 YR FEMALE	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3-4 YR MALE	7	11	13	14	26	23	10	15	10	5	2	-0	0	0	0	0	0	0	0	0	0
4-5 YR MALE	6	8	10	10	8	12	8	4	2	0	-0	0	0	0	0	0	0	0	0	0	0
OVER 5 YR MALE	20	27	29	26	18	10	7	3	0	0	0	0	0	0	0	0	0	0	0	0	0
OVER 3 YR FEM.	43	44	44	44	46	47	51	53	56	50	61	95	104	105	104	103	103	103	103	103	103
BREEDING MALE	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
TOT. CULLING	90	103	126	143	159	166	169	172	176	181	180	216	231	231	229	228	228	228	220	228	228
OFF-TAKE RATE	9	10	11	12	13	13	13	13	13	13	13	15	15	16	15	15	15	15	15	15	15

PURCH. OF BR. MALE WITH PROJECT SITUATION

YEAR	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	17	20
	3	3	3	4	4	4	5	5	5	5	5	4	4	4	4	4	4	4	4	4	4

MALI

SITUATION ET PERSPECTIVES DE L'ELEVAGE

ANNEXE 6
Tableau 10

VALEUR DE LA PRODUCTION (changement sur 10 ans)

ECONOMIC/FINANCIAL ANALYSIS

Incremental

SALES OF AN. - Incremental

YEAR	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0
0-1 YR MALE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0-1 YR FEMALE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1-2 YR MALE	0.0	40.6	194.1	440.3	681.0	900.9	1303.4	1665.3	2061.2	2402.4	2934.6
1-2 YR FEMALE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.9	16.4	22.4	20.4
2-3 YR MALE	0.0	120.7	155.9	590.8	745.9	812.4	862.4	828.1	718.8	514.1	191.4
2-3 YR FEMALE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.5	7.3	10.0
3-4 YR MALE	0.0	204.7	339.9	415.3	720.6	557.3	327.6	80.5	-196.4	-441.7	-608.5
4-5 YR MALE	0.0	131.0	201.6	269.3	199.1	138.8	-139.0	-363.3	-557.1	-695.6	-673.7
OVER 5 YR MALE	0.0	599.5	667.7	390.3	-162.9	-726.9	-1364.4	-1914.7	-2233.3	-2426.5	-2581.8
OVER 3 YR FEM.	0.0	0.0	0.0	0.0	0.0	-2038.9	-2240.7	-1671.7	-1341.4	-783.6	-752.8
BREEDING MALE	0.0	0.0	0.0	0.0	0.0	0.0	13.5	28.0	30.3	48.6	50.9
TOTAL	0.0	1064.5	1679.3	2122.0	2191.7	-284.3	-1245.2	-1359.8	-1472.0	-1472.4	-1393.4
OTHER INCOMES	0.0	0.0	0.0	0.0	0.0	0.0	370.7	768.4	1053.0	1335.3	1619.2

YEAR	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0	19.0	20.0
0-1 YR MALE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0-1 YR FEMALE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1-2 YR MALE	3002.9	3237.7	3220.9	3201.9	3201.9	3201.9	3201.9	3201.9	3201.9	3201.9
1-2 YR FEMALE	34.5	40.6	40.2	39.2	39.2	39.2	39.2	39.2	39.2	39.2
2-3 YR MALE	-257.0	-257.3	-257.3	-257.3	-257.3	-257.3	-257.3	-257.3	-257.3	-257.3
2-3 YR FEMALE	12.7	15.4	10.2	18.0	17.6	17.6	17.6	17.6	17.6	17.6
3-4 YR MALE	-706.0	-703.0	-701.5	-701.5	-701.5	-701.5	-701.5	-701.5	-701.5	-701.5
4-5 YR MALE	-651.6	-647.5	-644.7	-643.3	-643.3	-643.3	-643.3	-643.3	-643.3	-643.3
OVER 5 YR MALE	-2657.9	-2679.9	-2689.3	-2691.9	-2691.9	-2691.9	-2691.8	-2691.8	-2691.8	-2691.8
OVER 3 YR FEM.	1162.4	1645.1	1676.6	1657.7	1607.7	1607.7	1607.7	1607.7	1607.7	1607.7
BREEDING MALE	69.3	68.7	66.9	66.9	66.9	66.9	66.9	66.9	66.9	66.9
TOTAL	83.5	719.8	730.0	691.6	639.2	639.3	639.3	639.3	639.3	639.5
OTHER INCOMES	1906.4	1890.0	1840.0	1840.0	1840.0	1840.0	1840.0	1840.0	1840.0	1840.0

INCREMENTAL VALUE OF HERD AT YEAR 20.0 : -2772 - Incremental

MALI

SITUATION ET PERSPECTIVES DE L'ELEVAGE

ANNEXE 6
Tableau 11

COSTS/BENEFITS FLOW - Incremental (changement sur 10 ans)

YEARS	TOTAL COSTS	GROSS REV.	BALANCE
0	0	0	0
1	0	1065	1065
2	0	1679	1679
3	0	2122	2122
4	0	2192	2192
5	0	-284	-284
6	0	-875	-875
7	0	-591	-591
8	0	-439	-439
9	0	-137	-137
10	0	226	226
11	0	1995	1995
12	0	2610	2610
13	0	2578	2578
14	0	2532	2532
15	0	2479	2479
16	0	2479	2479
17	0	2479	2479
18	0	2479	2479
19	0	2479	2479
20	0	-293	-293

NET PRESENT VALUES FOR AN OPPORTUNITY COST OF CAPITAL OF 12.0 %

WITHOUT PROJECT SITUATION : 92781.6
WITH PROJECT SITUATION : 101098.5
INCREMENTAL VALUE : 8316.9
PERCENTAGE CHANGE : 9.0 % OVER WITHOUT PROJECT SITUATION

NET PRESENT VALUES FOR AN OPPORTUNITY COST OF CAPITAL OF 20.0 %

WITHOUT PROJECT SITUATION : 57644.8
WITH PROJECT SITUATION : 62887.7
INCREMENTAL VALUE : 5242.9
PERCENTAGE CHANGE : 9.1 % OVER WITHOUT PROJECT SITUATION

VALEUR DE LA PRODUCTION
(changement sur 10 ans et augmentation de prix)

ECONOMIC/FINANCIAL ANALYSIS - Incremental

YEAR	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0
SALES OF AN. - Incremental											
0-1 YR MALE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0-1 YR FEMALE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1-2 YR MALE	0.0	40.6	226.5	562.6	940.4	1411.2	1046.5	2325.7	2050.0	3411.0	4013.5
1-2 YR FEMALE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.9	16.4	22.4	28.4
2-3 YR MALE	0.0	128.7	195.9	598.8	745.9	812.4	862.4	828.1	718.8	514.1	191.4
2-3 YR FEMALE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.5	7.3	10.6
3-4 YR MALE	0.0	284.7	339.9	415.3	728.6	557.3	327.6	80.5	-196.4	-441.7	-600.5
4-5 YR MALE	0.0	101.0	281.6	269.3	199.1	130.8	-139.0	-363.3	-559.1	-695.6	-673.7
OVER 5 YR MALE	0.0	509.5	667.7	398.3	-162.9	-726.9	-1364.4	-1914.7	-2233.3	-2426.5	-2501.8
OVER 3 YR FEM.	0.0	0.0	0.0	0.0	0.0	-2030.9	-2240.7	-1691.7	-1341.4	-903.6	-752.8
BREEDING MALE	0.0	0.0	0.0	0.0	0.0	0.0	13.5	28.0	30.3	40.6	58.9
TOTAL	0.0	1064.5	1711.6	2244.3	2451.1	145.9	-702.1	-699.5	-703.2	-543.7	-314.5
OTHER INCOMES	0.0	0.0	0.0	0.0	0.0	0.0	370.7	768.4	1053.0	1335.3	1619.2

YEAR	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0	19.0	20.0
SALES OF AN. - Incremental										
0-1 YR MALE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0-1 YR FEMALE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1-2 YR MALE	4211.0	4417.4	4405.6	4369.7	4369.7	4369.7	4369.7	4369.7	4369.7	4369.7
1-2 YR FEMALE	34.5	40.6	40.2	39.2	39.2	39.2	39.2	39.2	39.2	39.2
2-3 YR MALE	-257.0	-257.3	-257.3	-257.3	-257.3	-257.3	-257.3	-257.3	-257.3	-257.3
2-3 YR FEMALE	12.7	15.4	18.2	18.0	17.6	17.6	17.6	17.6	17.6	17.6
3-4 YR MALE	-706.0	-703.0	-701.5	-701.5	-701.5	-701.5	-701.5	-701.5	-701.5	-701.5
4-5 YR MALE	-651.6	-647.5	-644.7	-643.3	-643.3	-643.3	-643.3	-643.3	-643.3	-643.3
OVER 5 YR MALE	-2657.9	-2679.9	-2689.3	-2691.9	-2691.9	-2691.9	-2691.3	-2691.8	-2691.8	-2691.8
OVER 3 YR FEM.	1162.4	1645.1	1676.6	1659.7	1687.7	1687.7	1687.7	1687.7	1687.7	1687.7
BREEDING MALE	69.3	68.7	66.9	66.9	66.9	66.9	66.9	66.9	66.9	66.9
TOTAL	1216.5	1099.5	1914.7	1857.3	1887.0	1887.0	1887.0	1887.1	1887.1	1887.1
OTHER INCOMES	1906.4	1890.0	1840.0	1840.0	1840.0	1840.0	1840.0	1840.0	1840.0	1840.0

INCREMENTAL VALUE OF HERD AT YEAR 20.0 : -1430 - Incremental

Annexe 6
Tableau 12

MALI

SITUATION ET PERSPECTIVES DE L'ELEVAGE

ANNEXE 6
Tableau 13

COSTS/BENEFITS FLOW - Incremental
Changement sur 10 and et augmentation de prix

YEARS	TOTAL COSTS	GROSS REV.	BALANCE
0	0	0	0
1	0	1065	1065
2	0	1712	1712
3	0	2244	2244
4	0	2451	2451
5	0	146	146
6	0	-331	-331
7	0	67	67
8	0	350	350
9	0	791	791
10	0	1305	1305
11	0	3123	3123
12	0	3790	3790
13	0	3755	3755
14	0	3699	3699
15	0	3647	3647
16	0	3647	3647
17	0	3647	3647
18	0	3647	3647
19	0	3647	3647
20	0	2217	2217

NET PRESENT VALUES FOR AN OPPORTUNITY COST OF CAPITAL OF 12.0 %

WITHOUT PROJECT SITUATION : 92781.6
WITH PROJECT SITUATION : 105452.2
INCREMENTAL VALUE : 12670.6
PERCENTAGE CHANGE : 13.7 % OVER WITHOUT PROJECT SITUATION

NET PRESENT VALUES FOR AN OPPORTUNITY COST OF CAPITAL OF 20.0 %

WITHOUT PROJECT SITUATION : 57644.8
WITH PROJECT SITUATION : 65005.3
INCREMENTAL VALUE : 7360.5
PERCENTAGE CHANGE : 12.8 % OVER WITHOUT PROJECT SITUATION

SELECTED BIBLIOGRAPHY

I. Project Analysis Methodology

(a) Reference Publications

1. LITTLE I.M.D. and MIRRLEES J.A., Project Appraisal and Planning for Developing Countries; Hainemann Educational Book Limited, London, 1974.

2. BRIDIER M. et MICHAILOF S. - Guide pratique d'analyse de projets - Economica, Paris, 1980.

3. PRICE GITTINGER J. - Economic Analysis of Agricultural Projects - The Johns Hopkins University Press, Baltimore, 1972.

4. BROWN M.L. Farm budgets - From Farm Income Analysis to Agricultural Project Analysis - World Bank staff occasional paper no. 29 - The Johns Hopkins University Press, Baltimore, 1979.

5. SQUIRE L. and VAN DER TAK G. - Economic Analysis of Projects - A World Bank Research Publication - The Johns Hopkins University Press, Baltimore, 1975.

6. Guide to the Economic Evaluation of Irrigation Projects by H. BERGMANN and J.M. BOUSSARD - OECD - Paris 1976.

7. UNIDO: Guidelines for Project Evaluation; Project Formulation and Evaluation Series no. 2; United Nations, New York, 1972.

8. Charles PROU and Marc CHERVEL, Etablissement des programmes en économie sous-développée, tome 3, l'étude des grappes de projets, Paris, Dunod, 1970.

(Each of these publications contains a bibliography).

(b) FAO Investment Centre Guidelines (FAO/World Bank Cooperative Programme) (1)

9. Guidelines for the Preparation of Agricultural Investment Projects, FAO/WB CP, August 1977 (E + F + S). Revised August 1982.

10. Guideline for the Preparation of Feasibility Studies, Rural Development Projects, FAO/WB CP, March 1975 (E + S).

11. Guidelines for the Preparation of Feasibility Studies, Agricultural Credit Projects, FAO/WB CP, October 1972 (E).

12. Guidelines for the Preparation of Feasibility Studies, Livestock Projects, FAO/WB CP, March 1971 (E + F + S).

---

(1) E = in English - F = in French - S = in Spanish.

13. Guideline for the Preparation of Feasibility Studies, for Irrigation and Drainage Projects, FAO/WB CP, December 1970 (E + F + S).

14. Outlines for Projects to be Presented for Financing FAO/WB CP, September 1967 (E + F + S).

15. The Project Cycle, A.M. EID, Near-East Workshop on Agricultural Investment Projects, January/February 1978 (E + F + Arabic).

16. Guidelines for Basic Data Collection and Analysis for the Preparation of Integrated Rural Development Projects, FAO/WB CP, October 1982 (E + F + S).

(c) Other FAO and World Bank Publications

17. Directives générales applicables à l'analyse des projets de production agricole - Etudes sur la planification agricole No. 14 - FAO - 1972.

18. Projets d'investissement agricole: textes pour le Proche-Orient (Choix de documents présentés au stage sur les projets d'investissement agricole à l'intention de la région du Proche-Orient, Rome, 9 janvier-10 février 1978) - Etudes sur la planification agricole No. 20 - FAO - 1980.

19. VIII Curso de planificaciõn del desarrollo agrĩcola - Selecciõn de conferencias - IRYDA, Madrid, y FAO, Roma, 1976.

20. Investment Criteria in Economic Analysis of Projects - Central Projects Note 2.01 - World Bank, Oct. 1978.

21. Social Cost-Benefit Analysis: A Guide for Country and Project Economist, to the derivation and application of economic and social accounting prices, by Colin Bruce - WB Staff Working Paper no. 239, August 1976.

22. The "Effects Method" of Project Evaluation - WB Staff Working Paper no. 231 - March 1976 - (analyse critique de la méthode des effets, proposant de modifier la méthode en utilisant certains prix de référence).

23. REUTLINGER S., Techniques for Project Appraisal under Uncertainty. World Bank Staff Occasional Papers no. 10. Washington, D.C., World Bank, 1970.

24. Risk and Sensitivity Analysis in the Economic Analysis of Projects - Central projects note 2.02 - World Bank - Oct 1978.

25. POULIQUEN L.Y., Risk Analysis in Project Appraisal. World Bank Staff Occasional Papers No. 11, Washington. D.C., World Bank, 1970.

II. Analysis and Programming

(a) General Publications - Computer Languages

26. WIRTH, N. Algorithms + Data Structures = Programs - Prentice Hall, 1976.

27. KNUTH, D. E. The Art of Computer Programming (3 vol.), Reading Addison - Wasley Publishing Company, 1968.

28. GABRINI, P. - Les structures de données et leurs représentations, Gaëtan Morin, 1982.

29. WIRTH, N. (traduction O. LECARME) - Introduction à la programmation systématique - Masson, 1977.

30. Structured System Programming - J. WELSH and M. McKEAG - Prentice-Hall International - 1980.

31. WIRTH, N. and JENSEN, K.: Pascal, User Manual and Report - Springer Verlage, 1978.

32. WIRTH, N. et JENSEN, K.: Pascal, Manuel de l'utilisateur. Traduit de l'anglais par P. KRUCHTEN et J.-A. HERNANDEZ - Eyrolles, 1981.

33. LIGNELET, P. - Pascal - T1: Manuel de base - T2: Techniques de programmation et concepts avancés - MASSON, 1980.

34. FINDLAY, W. and WATT, D.A.: Pascal, An Introduction to Methodical Programming - PITMAN, 1981.

35. HEWLETT-PACKARD - SYSTEM 45 DESKTOP COMPUTER=
- Operating and programming
- Mass storage techniques.
(These manuals contain the description of the BASIC language used for some of the programmes mentioned).

36. PLOVIN M. - Programmer en BASIC - P.S.I. Paris, 1980.

37. BOISGONTIER J. - Le BASIC et ses fichiers - Méthodes pratiques - P.S.I. Paris, 1980.

(b) Manuals of the Programmes Mentioned

38. CB Display (Users Guide to CB Display) - Economic and Policy Division, Agriculture and Rural Development Department, the World Bank, April 4, 1979. - Programme con[u par G. Temple - Programmation par E. RAMISCAL, T. KLEIN, P. GUTTERMAN.

39. Farm Model Processor (FARMOD) - External Characteristics - by V. SAHASRABUDME and G. TEMPLE - The World Bank, October 30, 1981.

40. The Project Cost Table System (COSTAB) - Users Guide (draft) - Compass Users Guide Series, World Bank, June 1982.

41. COMPASS PROCEDURES and EDITOR - Users Guide - Compass Processor Series - World Bank, Nov. 1981.

42. Notice d'emploi du programme MADS - OCDE - Service de la coopération technique, Direction de l'agriculture, Paris, 17 Mai 1979.

43. MADS II - Manual - FAO Investment Center - October 1980.

44. Data Analysis and Simulation Programme (DASI) - User's Manual, by C. Cappi - FAO Development Policy Studies and Training Service, April 1982.

III. Reports on Projects mentioned

45. MALI - Projet de développement de la zone lacustre.

- Rapport de reconnaissance - FAO CP - Mars 1980.

- Aménagement de mares (Document de travail) - FAO CP - Juin 1980.

46. MALI - Situation et perspectives de l'élevage - Banque mondiale - Mission régionale en Afrique de l'Ouest, Abidjan, avril 1980.

47. NEPAL - Community Forestry Development Project - Preparation Report - FAO CP - April 1979.

48. PARAGUAY - Agricultural Credit Project - Preparation Report FAO CP - October 1982.

49. MAROC - Projet de développement de la petite et moyenne hydraulique agricole - Rapport de préparation (2 volumes) FAO CP - Mai 1982.

50. SENEGAL - Projet de développement de l'élevage au Sénégal oriental et en Haute Casamance - FAO CP - Août 1982.

51. MALI - Fonds de développement associatif dans la région de Ségou - Préparation pour le FIDA - VILLARET/DE WALCK/SON - FAO ISP - 21 août 1981.

M-63
ISBN 92-5-102255-0