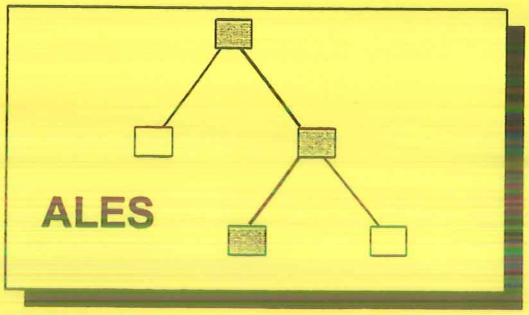
van Engelen

Cornell University, Department of Soil, Crop & Atmospheric Sciences
SCAS Teaching Series No. T93-2
Ithaca, NY USA
May 1993



Automated Land Evaluation System

ALES Version 4 User's Manual

May 1993 Printing

David G. Rossiter
Armand R. Van Wambeke

Correspondence regarding this manual or the ALES program should be addressed to:
ALES Project
Department of Soil, Crop & Atmospheric Sciences (SCAS)
Bradfield & Emerson Halls, 10th floor
Cornell University
Ithaca, NY 14853 USA

Internet: ALES@cornell.edu
Telex: WUI 6713054, Attention: Department of SCAS
FAX: USA (607) 255-8615 or -2644

Portions of this manual have previously been published in these references:

Rossiter, D.G. 1988. The Automated Land Evaluation System: a microcomputer program to assist in land evaluation. 1988. Ph.D. disseration, Cornell University. University Microfilms, Ann Arbor, MI.

Rossiter, D.G. 1988. ALES: a microcomputer program to assist in land evaluation. Proc. International Workshop on Land Qualities in Space and Time, Wageningen, the Netherlands, 22-26 August 1988, Pudoc, Wageningen.

Rossiter, D.G. 1990. ALES: A framework for land evaluation using a microcomputer. Soil Use & Management 6(1): 7-20.

Rossiter, D.G., M. Tolomeo, & A.R. Van Wambeke, 1988. Automated Land Evaluation System Version 1 User's Manual. Department of Agronomy, Cornell University, Ithaca, NY.

Rossiter, D.G., & A.R. Van Wambeke, 1989. Automated Land Evaluation System Version 2 User's Manual. Department of Agronomy, Cornell University, Ithaca, NY.

Rossiter, D.G., & A.R. Van Wambeke, 1991. Automated Land Evaluation System Version 3 User's Manual. Department of Soil, Crop & Atmospheric Sciences, Cornell University, Ithaca, NY.

Table of Contents

10646

1 Introduction

Welcome to ALES (the Automated Land Evaluation System), Version 4. We think ALES is a novel and useful approach to land evaluation, and we hope that ALES will be of use to you in your land evaluation activities.

1.1 Objectves of this manual

This manual has several objectives:

√ *First*, we want you to understand the *objectives of the ALES project* and how it fits in the discipline of land evaluation.

√ *Second*, we want you to be able to *perform land evaluations* using ALES.

√ *Third*, we want you to be able to *integrate ALES into your entire information processing system*, which may include geographic information systems, natural resources databases, word processors, optimization and decision support software, as well as ALES.

√ *Finally*, we want you to be able to *install and maintain the ALES program* on your computer.

Please refer to the table of contents for the organization of this manual.

1.2 How to use this manual

We suggest that you use the manual as follows.

√ Read Chapters 2 and 3 to understand the basic *principles of land evaluation*, and where ALES fits in the land evaluation process. If the material in Chapter 3 ("Land Evaluation") is unfamiliar, you should also study the land evaluation *texts* cited in that chapter.

√ *Install* the ALES software on your computer according to the instructions of Chapter 5.

√ Read Chapter 4 for an explaination of the different categories of ALES users.

√ Read Chapter 6 for an orientation to how to *interact* with ALES.

√ Complete *tutorials* 1 through 4 (Chapter 7). At the same time that you are working on the third and fourth tutorials (which cover model building), consult relevant sections of Chapter 8, which explains the principles of model design.

√ If you have the IDRISI Geographical Information System, follow *tutorial 5* and refer to Chapter 10.

√ If you plan to *integrate* ALES with other software, read Chapter 9, "Interfacing ALES to other systems".

√ The *system administrator* at your location should read Chapter 11, "Maintaining an ALES system" (see Chapter 4 for a discussion of the system manager's role).

1.3 Acknowledgements

David Rossiter was responsible for the requirements analysis and logical design of the system, as well as its implementation, including system design, coding, testing, and documentation. He wrote this manual and supervised the translation into the Spanish language. He also developed the sample evaluations on which Tutorials 1, 2, and 4 are based. Versions 1, 2 and 4 of ALES were developed when he was employed by the International Soils Group of the Department of Soil, Crop, & Atmospheric Sciences at Cornell University. He wrote Version 3 as a private consultant and developed the ALIDRIS interface with the IDRISI geographical system. He has presented training sessions on ALES in Brazil, Venezuela and the Philippines.

Professor **Armand Van Wambeke**, head of the International Soils Group, conceived of the idea that land evaluation software for the microcomputer would be of tremendous benefit to the soil survey and land evaluation community worldwide, and initiated the project that grew into ALES. All through the project, his insistence that the software apply to practical objectives guided many design decisions. He critically read drafts of previous manuals and suggested many changes that made them much clearer. He has publicized the ALES project in the course of his travels to Africa, Europe, and Latin America. He supervised the translation of ALES into the French language.

Professor **Christian Schvartz**, of the Institut Supérieur d'Agriculture of the University of Lille, France, translated into French all texts included in the ALES software.

Dr. **Ngongo Luhembwe** , of the University of Zaire, contributed to the French translation of Chapters 7 and 8 of this manual.

Ing. Agr. **Antonio Jiménez,** of the Venezuelan Ministry of the Environment was mainly responsible for the Spanish translation of the program texts and manual. He also developed new tutorials for the Spanish-language version of ALES.

The **Venezuelan Ministry of the Environment** (Ministerio del Ambiente y de los Recursos Naturales Renovables - MARNR) granted Ing. Jiménez leave with pay and provided him with transportation to Ithaca, so that he might work on the Spanish translation of ALES.

Ing. Agr. **Maria Fernanda Rodriguez,** of the Venezuelan Ministry of the Environment assisted in the Spanish translation.

Professor **Norman Scott**, formerly Director of Research for the New York State College of Agriculture and Life Sciences, encouraged the project and provided the legal support of the College, as well as start-up funds.

The **Cornell Institute for International Food and Agricultural Development** (CIIFAD) supported the development of Version 4. It also supported the translation into Spanish of the V3 manual and the development of the Spanish-language tutorials by Ing Jiménez.

Mr. **Michael Tolomeo**, formerly of the International Soils Group developed the tutorial evaluation that accompanied version 1 and a substantial part of the version 1 ALES User's Manual.

Ms. **Christine DeRoller**, formerly of the International Soils Group, developed several sample evaluations in the course of her MSc. work on land evaluation in Guatemala. Tutorial example 3 is based on her work. She also served as a particularly keen critic of ALES version 2's functionality and user interface. She helped with the initial Spanish translation of the menus.

Dr. **Terence Forbes**, formerly of the International Soils Group, demonstrated linkages between existing soils databases, ALES, and a geographical information system.

Mr. **Tom Fredricks, John Godfrey**, and **John Lewkowicz** from Veterinary Computer Services of the New York State College of Veterinary Medicine served as occasional consultants on programming questions. Tom Fredricks also served as the technical backstop for ALES from April 1991 through August 1992.

Professor **Larry Geohring** of the Department of Agricultural Engineering at Cornell provided cost figures for land drainage that were used in Tutorial 2.

Project Ezra, a cooperative project between Cornell University and the International Business Machines Corporation's "Advanced Education Projects" program, provided the IBM PC/AT system on which ALES was developed.

Ms. **Mary Ellen Niederhofer** of Cornell's Department of Soil, Crop, & Atmospheric Sciences has served as ALES project coordinator since mid 1989. She is responsible for inquiries and sales.

2 A brief introduction to ALES

This chapter provides an overview of the ALES program and the ALES project, and their relation to the discipline of land evaluation.

2.1 The ALES program

The Automated Land Evaluation System, or ALES, is a computer program that allows land evaluators to build *expert systems* to evaluate land according to the method presented in the *Food and Agriculture Organization "Framework for Land Evaluation"* (FAO 1976). It is intended for use in *project* or *regional scale land evaluation*. The entities evaluated by ALES are *map units*, which may be defined either broadly (as e.g. in reconnaissance surveys and general feasibility studies) or narrowly (as e.g. in detailed resource surveys and farm-scale planning). Since *each model is built by a different evaluator to satisfy local needs*, there is no fixed list of land use requirements by which land uses are evaluated, and no fixed list of land characteristics from which land qualities are inferred. Instead, these lists are determined by the evaluator to suit *local conditions and objectives.*

To reiterate the key point:

> **Evaluators build their own *expert systems* with ALES, taking into account local conditions and objectives. ALES is not by itself an expert system, and does not include by itself any knowledge about land and land use. ALES is a *framework* within which evaluators can express their own, local, knowledge.**

ALES has *six components*:

1) a framework for a *knowledge base* describing proposed land uses, in both physical and economic terms;

2) a framework for a *database* describing the land areas being evaluated;

3) an *inference mechanism* to relate these two, thereby computing the physical and economic suitability of a set of map units for a set of proposed land uses;

4) an *explanation facility* that enables model builders to understand and fine-tune their models;

5) a *consultation mode* that enables a casual user to query the system about one land use at a time, and

6) a *report generator.*

The program is highly *interactive*, and takes full advantage of the PC video display and keyboard. It is designed to be self-explanatory, and leads the user through a series of *menus*, data entry *forms*, *'why?'* explanations, and *dialogues*, as well as *context-sensitive help* screens. *Function keys* control most operations.

The program user can choose to interact with ALES in any human language to which the display text has been translated[1]. Special attention has been given to the creation, editing, and display of *decision trees*. In addition, model builders can write their own notes ('annotations') that can be displayed by the model user at appropriate points in the evaluation.

ALES runs on the *IBM PC* microcomputer and its successors, such as the IBM PC/AT and PS/2 series, and on "PC-compatible" machines (i.e. those which are not made by IBM, but which function similarly), under the PC-DOS operating system (or, MS-DOS for compatibles), version 2.3 or later. It requires at least 384 Kb of primary (RAM) memory (preferably 640 Kb) and 3.0 Mb of space on a hard disk for the program, its support files, and a database sufficient for a few evaluations; subsequent evaluations require more disk space. The program is written in the MUMPS programming language, and uses the DataTree MUMPS language and database system (Data Tree Inc., 1986). (See §5.1 for more details on system requirements.)

2.2 Objectives of the ALES project

There is today a high demand in many developing countries for information on the suitability of land for a wide range of land uses. This demand comes from banks and other lending agencies, planning offices, government ministries, and rural and urban development officers. The intent of these agencies is to plan or recommend the uses of land in a rational and equitable way, using the techniques of *land use planning*, primarily for agricultural development. Recommendations and plans must usually be made quickly, in response to actual needs and current conditions.

Land use planning has as its basic purpose to ensure that each area of land will be used so as to provide maximum social benefits, especially including food production, without degradation of the land resource. Planning has two aspects: the *political* and the *rational*. The political process is necessary to initiate and carry out land use planning, to set its objectives, and to arbitrate among competing interests. The rational, or technical, part of planning ensures that plans are feasible, that cost and return estimates are accurate, and that sufficient data have been collected and collated to ensure these. While the political aspect of land use planning is outside the scope of the land evaluator, clearly the expert knowledge of the agronomist, the production agriculture specialist, and other agriculturalists, must form the basis of correct land use planning in its technical aspect.

There are many sources of knowledge about land and land use. Even in less developed countries, there is often adequate information, such as soil surveys, about land resources, and much of this information is of high quality. Also, there is often a lot of information about the relation between land and land use. Much of this information is local, acquired over years of experience by local agriculturalists as well as farmers. Often there has been relevant work at local experiment stations, or in other areas with similar environments. Each of these sources of knowledge is expressed differently, published (or not) in diverse locations, and held by different people.

Our aim is to allow agricultural scientists to present natural resource information in a form that is directly useful to land use planners. Our method was to write a computer program to be used by

[1] As of May 1993, these are English, French and Spanish.

agriculturalists, with the results of the program to be presented to land use planners. The program is designed to allow contributions from all relevant sources of knowledge.

A further objective is to use the large quantity of information that has been recorded to date in soil surveys and other land resource inventories, much of which is sitting *unused* on office shelves. There are a variety of reasons for this disuse, primary of which is that the surveys are not interpreted for various land uses, and secondly that surveys have widely different definitions of map units and land characteristics. So a major design objective of ALES is to allow the use of land data in almost any format, as well as easy interchange of computer-readable data with national soils databases and similar repositories of land data.

This work is especially intended for application in *less developed countries*, since the opportunities for planning are much greater there than in the developed countries. Planning implies a willingness to change, and in countries with developed agriculture, change is usually at the farm level, not involving larger-scale planning. Further, the need for change in agriculture is greater in those countries that can not feed themselves, provide meaningful occupations to their rural populations, or generate sufficient exports to stimulate their economic development.

However, ALES is certainly applicable to many problems in *developed countries*. Examples are defining prime farmland for agricultural land preservation laws or other rural zoning, arriving at assessments of agricultural potential for taxation or land valuation, and performing soil survey interpretations. These countries usually have well-established soils databases, from which data can easily be imported into ALES; conversely, results from ALES can easily be exported to programs that perform further computations such as land valuation.

The ALES project was the outgrowth of the interests of Professor Armand Van Wambeke of the International Soils Group at Cornell, to make soil survey and other natural resource inventory information more useful to those responsible for land use decisions. This followed several years of work in the International Soils Group on the adequacy of soil resource inventories. The FAO land evaluation methodology, which is outlined in Chapter 3, seemed to provide a useful starting point, being based on many years of expert consultation.

However, methods based on the FAO methodology are data-intensive, involve many repetitive calculations or table lookups, and are tedious if many alternatives are to be compared. Manual procedures, both for construction of matching tables or similar devices, and for calculation of suitability, are time-consuming and error prone. Hence an automated procedure seems like a natural development, and indeed there have been a few previous efforts in this direction, most notably the LECS system in Indonesia (Wood & Dent 1983, Purnell 1987). However, this successful system had three limitations for widespread adoption: (1) it is at the lowest, or most specific, level in the three-level hierarchy of the methodology, and thus is applicable to a specific scale and area of the world; (2) it was only available on mainframe computers; (3) it was a non-interactive program. (Since ALES was first written, LECS has been incorporated into the FAO's Agricultural Planning Toolkit, thus removing limitations 2 and 3.)

We have attempted make the FAO methodology more widely applicable by making it available as a microcomputer program to individual development projects. We have also incorporated a definite method of economic land evaluation. Consistent with the 'small is beautiful' philosophy of the project, we hoped to allow evaluators to use locally-available knowledge and data in a flexible model framework. Finally, we wanted a fun, easy-to-use computer program, which would make the task of developing, refining, and testing land evaluation models as enjoyable as possible.

---

3 Land Evaluation

Land evaluation is a vital activity for rural development and planning worldwide. The success of the FAO framework (FAO 1976) and subsequent guidelines for application in diverse types of land uses and land areas (FAO 1983, FAO 1984, FAO 1985, Siderius 1984, Siderius 1986) is an encouraging development. The following paragraph provides a brief review of the specialized terminology of land evaluation. For more information, the reader is encouraged to read the textbook of Dent & Young (1981) or one of the cited FAO bulletins.

√ *Land evaluation* may be defined as 'the process of assessment of land performance when used for specified purposes' (FAO 1984).

√ In a land evaluation exercise, the land use planner matches land areas, termed *land mapping units* (LMU) with land uses, termed *land utilization types* (LUT), determining the relative suitability of each area for each use.

√ Land utilization types are specified by a set of *land-use requirements* (LUR), which are 'the conditions of land necessary for the successful and sustained practice of [a given LUT]' (FAO 1984).

√ Land units are defined by the values of a set of *land characteristics* (LC), which are simple attributes of land that can be measured or estimated. Land characteristic values are combined into levels of *land qualities* (LQ), which are complex attributes of land which influence the suitability of land in a semi-independent manner.

Notice that *land use requirements* express the *demands* of a land use, whereas *land qualities* express the *supply*, i.e., what a particular land area can offer. The distinction is partially semantic but mostly depends on one's point of view. For example, a crop *demands* an adequate water supply: this is a *land use requirement*. On the other hand, a particular land unit can *supply* a certain amount of water: this is a *land quality*. In ALES, the model builder express the LUT in terms of its LUR's, and the system computes the relevant LQ's based on the land unit data.

The actual evaluation of each *land unit* for each *land utilization type* is computed by:

(1) *determining* the actual land characteristic values for the land unit;

(2) *combining* these land characteristic values into land quality values (i.e., *inferring* the LQ from the set of LC's);

(3) *matching* the land quality values with land use requirements;

(4) *combining* these land quality values into composite *suitability classes* (i.e., *inferring* the suitability from the set of LQ's).

There are two kinds of suitability: *physical*, which expresses the degree to which the sustained implementation of the LUT on a certain land unit is feasible without unacceptable risk to the ecosystem, and *economic*, which is based on a calculation of the economic return which may be expected if the LUT in question is implemented on the land unit.

The FAO methodology has three levels of detail:

√ At the highest level is the *Framework*, which is the methodology with no content. ALES is an automated realization of the framework.

√ At the middle level are specific *Guidelines*, which provide suggestions for content.

√ At the lowest level is each separate *evaluation*, which incorporates parts of one or more Guidelines as well as local modifications.

3.1 The structure of an ALES evaluation

In ALES, each evaluation consists of a set of *land utilization types* (LUT), i.e. proposed land uses, and a set of *map units* (LMU), i.e. land areas being considered. Each map unit is evaluated for its suitability for each land utilization type, resulting in a *suitability matrix*. There are two kinds of suitability in the FAO system: physical and economic; ALES can evaluate for both.

3.1.1 Physical evaluation

A physical evaluation indicates the degree of suitability for a land use, without respect to economic conditions. (It certainly can be debated whether this is desirable, or even possible.) A physical evaluation emphasizes the relatively permanent aspects of suitability, such as climate and edaphic conditions, rather than changeable ones, such as prices. It tends to concentrate on *risks* or *hazards*, e.g. to the environment, or *absolute limitations*, e.g. due to climate, of implementing a given land utilization type on a given land area. The idea is that if a use is too risky or physically impossible, no economic analysis can justify it. In ALES, if a land unit is rated physically unsuitable (in the highest-numbered physical suitability class), it will not be evaluated economically; it is automatically in FAO suitability class 'N2' (permanently unsuitable under the assumptions of the LUT).

For land that is not completely unsuitable, a physical evaluation can be used to divide the land into degrees of suitability, based purely on physical conditions. The advantage is that physical suitability doesn't change quickly. The disadvantages are: (1) land use decisions are always based on economics, even if the economics may be distorted by government intervention; (2) it is unclear, in the absence of an economic scale, how to determine degrees of suitability; (3) it is unclear how to compare two land utilization types in strictly physical terms. A physical evaluation can also be used to divide the land units into *management groups*.

3.1.2 Economic evaluation[2]

ALES is intended to provide the land use planner with a realistic estimate of the economic suitability of each land unit for each proposed land use. Economic suitability may be evaluated by several economic metrics, including:

(1) the predicted *gross margin*, based on predicted costs and returns, in units of currency per hectare-year.

---

[2]See Chapter 8 for more information.

(2) the *Net Present Value* (NPV) of a LUT, in units of currency per hectare, over the lifetime of a project, called the *planning horizon*;

(3) the *Benefit/Cost Ratio*, based on the present values of cash in and out, in a dimensionless ratio (<1 : costs outweigh benefits, =1: costs exactly equal benefits; >1 benefits outweigh costs);

(4) the *Internal Rate of Return* (IRR), based on the cash flows, in units of currency per hectare;

Except for the gross margin, these all depend greatly on the *discount rate*, which is the interest rate to be applied to future cash flows. In a commercial non-inflationary environment, this is the rate at which money may be borrowed. In an *inflationary* environment, if it is assumed that costs and prices will rise together with inflation, the discount rate can be set equal to the commercial interest rate, less the inflation rate. For *project analysis*, the discount rate is often set to a lower value than the commercial interest rate, to reflect the social value of the project.

Each of these predictors have their uses.

(1) The *gross margin* is appropriate for LUT's that do *not* involve any land improvements with a one-year or short-term rotation planning horizon. It is much less susceptible to errors due to incorrect assumptions about the discount rate. LUT's with different planning horizons can be compared with this measure, since it is normalized to a per-year basis. The comments of Dent & Young (p. 177 in FAO 1983) are relevant here:

> "Because of the assumptions involved in discounted cash flow analysis, it is better to use gross margins as a basis for land suitability classification where possible, namely, where land improvements are not important or not costly in relation to annual production... Where capital investment is involved, it is still preferable to conduct the land evaluation initially in terms of repayment capacity, as gross margin and net income on an annual basis. This relatively firm figure can later be subjected to the less objective process of discounting."

(2) The *Net Present Value* is, in theory[3], equivalent to the selling price of a unit area of land that is used for the proposed LUT, since the land user would in theory be indifferent to realizing the LUT or receiving the NPV now and investing it. An important caution about the NPV is that it is only valid to compare NPV's for LUT's with the *same planning horizon*: given equal NPV's but different planning horizons, the LUT with the shorter planning horizon would be preferred, because the value of the land is realized sooner.

(3) The *Benefit/cost ratio* expresses the NPV in terms of risk. The higher the B/CR, the less risky the LUT, because a lower-than-expected benefit, or higher-than-expected cost, will still leave the BC/R greater than one. Many industries and lending agencies have guidelines for acceptable BC/R's, e.g. that they be greater than 3.

(4) the *Internal Rate of Return* also expresses the risk: the higher IRR is less risky, because faced with an unexpected rise in discount rate, the LUT with a high IRR will still be profitable. This measure is time-neutral, so can be used to compare LUT's with different planning horizons.

---

[3]Not taking into account social values regarding the desireability of owning land, nor any speculation about the future value of the land.

Land utilization types may have any number of outputs, with any number of crops of each output over the planning horizon. Inputs are allocated to specific years within the plan, or may be related to the level of production of any output.. ALES can analyze rotations, multiple cropping systems, and intercrops. Outputs can have *negative* value (price), so that, for example, loss of topsoil could be reflected in the economic calculation. Environmental degradation outside the project area or other externalities can be treated in the same manner.

Costs are of three types: (1) inherent in the implementation of the LUT, called *S1 costs* in ALES; (2) only incurred on certain land units that have limitations, i.e. non-optimum levels of land qualities, called *additional costs* in ALES, and (3) related to the level of production, called *production-dependent costs* in ALES.

Examples of these three types of costs are: (1) standard cultural practices such as plowing, planting, fertilizing, weeding etc. if they are carried out on all land that is used for the LUT; (2) additional lime needed to bring acid soils up to the proper reaction for a crop: note that this input would *not* be needed on naturally neutral soils, so the cost is only incurred on non-optimum land; (3) manual labor for harvesting, which depends on the amount harvested, not the land area dedicated to the crop.

The model builder may relate *costs of production* to *land use requirements* as follows: an increasing level of limitation (i.e., higher severity level of the corresponding *land quality*) can result in an *increased cost of production* (the additional costs mentioned in the previous ¶), a *decreased yield*, or *both*. Costs of production may be *recurrent* (termed *annual* costs in ALES) or in a *specific year* within the plan.

Costs are expressed by listing the *number of units of inputs* that are required; prices are maintained in a separate table. Thus if the price of an input or product changes, the entire cost of production can be recomputed[4].

With the ease with which ALES computes yields and economic returns, the supposed disadvantage of economic evaluation, namely that economic conditions change quickly, evaporates. Within a few minutes, an ALES user can enter new economic parameters into a model, recompute an evaluation, and print the new results.

3.1.3 Suitability classes

Although ALES expresses its economic results in exact monetary amounts, the FAO Framework defines two *suitability orders* (*S* and *N*) and five *suitability classes* (*S1, S2, S3, N1* and *N2*). For consistency with the Framework, ALES allows the exact economic results to be classified into four *economic suitability classes*: S1, S2, S3 and N1. As in the FAO Framework, class *N2* is reserved for land that is *physically unsuitable*. In ALES this means land that is in the highest-numered *physical* suitability class. Land in this class is not even anlyzed in economic terms, because it has already been assigned to suitability class N2.

[4]See Appendix 1 for a computation diagram.

3.2 Highlights of the ALES approach to land evaluation

This section explains some of the more important features and limitations of the ALES implementation of the FAO Framework.

3.2.1 ALES usually evaluates map units

An important limitation of ALES is that is does not have any *georeferencing* capability and does not itself produce maps[5], so that analyses can not easily take into account proximity or adjacency requirements. ALES usually makes statements about *map units*, i.e. *sets* of delineations on a map, and assumes that the properties of all delineations with the same name are identical, within the precision of the map unit description. Since the delineations of a map unit have diverse locations, it is rarely possible to assign a value to a geographic land characteristic. Map units are typically 'natural' units defined by a soil, climate, geomorphic or physiographic natural resources inventory. Geographical information systems (Burrough 1986) are ideal tools for spatial analysis, and the results of ALES analyses may easily be used as data layers in these systems; see Chapter 9 'Interfacing' and Chapter 10 'ALIDRIS' for more information.

Having said the above, it *is* possible to use ALES to evaluate individual *delineations* or *management units* that have a specific location. It is possible to define *spatial land characteristics* e.g., distance to market, adjacency to a protected area etc. and *spatial land use requirements*, e.g., adjacency, and reason with these in decision trees. Since ALES has no map input or output, the data values for these LC's would have to be obtained from maps or a GIS and entered in the ALES database by hand.[6]

3.2.2 Class-to-class

In ALES, the entities that are used as the basis from which evaluation models are built (i.e. land characteristics) are most conveniently treated as *classified data*. That is, data values are from a *small finite set of possibilities*. These can either be *ordinal* (with an underlying continuous scale), e.g. for slope classes, or *nominal* (with no underlying ordering), e.g. for texture classes.

ALES also allows the model builder to define *continuous* land characteristics, that is, those whose data values are from an *infinite* set of possibilities in some *range* of the real numbers. An example is slope expressed in percent or degrees, as opposed to slope classes. ALES requires that these values be converted into analogous classified data values, by defining so-called *commensurate* land characteristics (i.e., with the same underlying scale of measurement), before land qualities can be determined. The ALES model builder *must* reason with classified data when building decision trees.

There are several reasons for using classified data as the basis for models. Most important is that ALES is evaluating land areas, not single points, so that a single value on a continuous scale is

[5]But see the ALIDRIS interface, Chapter 10 and Tutorial 5.

[6]If these values can be computed in a GIS and written as an ASCII table, ALES can read them from a data file.

not as meaningful as a class. Ordinal classes represent a range in which most of the variability in a map unit is presumed to fall. Nominal data, of course, has no continuous analog.

However, if the model builder wants to combine continuous land characteristic values into a composite value with some well-established formula (e.g. a soil loss equation or a regression model), this is allowed in ALES: a continuous land characteristic may be derived by a *formula* from a set of other continuous land characteristics. The derived characteristic can be used in other formulas or directly converted to a classified LC for use in decision trees.

3.2.3 Decision trees

The way in which the ALES model builder reasons with classified data is to build *decision trees*. Decision trees are hierarchical multiway keys in which the *leaves* are results such as land quality ratings, and the interior nodes (*branch points*) of the tree are decision criteria such as land characteristic values. These trees are constructed by the model builder, and traversed during the computation of an evaluation result, using actual land data. Figure 3.1 shows a simple decision tree, using the logic of Sanchez *et al.* (1982),which would allow ALES to determine the value of the *land quality* 'potential for P-fixation by iron' from the *land characteristics* (1) ratio of free Fe_2O_3 to clay in the topsoil, (2) percentage of clay in the topsoil, (3) hue of the topsoil matrix, and (4) topsoil structure. Note that ALES decision trees allow missing data to be replaced by other criteria, should the tree builder determine this is appropriate.

Figure 3.1

A decision tree to determine whether there is a likelihood of high P-fixation by iron (after Sanchez *et al.* 1982)

» *ratio of free Fe_2O_3 to clay in the topsoil*
[< 0.15] — LOW P-FIXATION
[_ 0.15] » *percentage of clay in the topsoil*
[<35%] — LOW P-FIXATION
[_35%] — HIGH P-FIXATION
[Unknown] » *hue of the topsoil matrix*
[R, 2.5YR, 5YR, 7.5YR] » *topsoil structure*
[granular] — HIGH P-FIXATION
[other] — LOW P-FIXATION
[10YR, Y, G, B] — LOW P-FIXATION
[Unknown] » *hue of the topsoil matrix*
[R, 2.5YR, 5YR, 7.5YR] » *topsoil structure*
[granular]—HIGH P-FIXATION
[other] — LOW P-FIXATION
[10YR, Y, G, B] — LOW P-FIXATION

Discriminant entities are introduced by '»' and *underlined.*
Values of the entities are [boxed].
The level in the tree is indicated by the leader characters, '--'.
Result values are introduced by '—' and written in SMALL CAPITALS.

In ALES, decision trees are used to determine:

(1) values (so-called *severity levels*) of *land qualities*, from values of *land characteristics*

(2) expected proportional *yields* of outputs, from values of *land qualities*[7]

(3) *physical suitability subclasses*, from values of *land qualities*[8]

(4) values of classified *land characteristics*, from a set of other classified *land characteristics*.[9]

Decision trees have several *advantages* as a method of expressing inferences. First, the model builder and user both have an *explicit* representation of the process used to reach a decision. Trees may be traced by hand or with the aid of ALES' 'why?' screens, so that the model *user* can see exactly how ALES reached a decision, which of course is based on the model *builder's* logic.

> If many factors must be considered, decision trees may become cumbersome, since they grow exponentially with the number of factors. ALES therefore allows the use of multiplicative (proportional) and limiting yield factors, and the maximum limitation method for physical suitability, in addition to decision trees, in appropriate circumstances. In addition, the evaluator may introduce intermediate result in order to break down one tree into several smaller components.

3.3 An annotated bibliography of land evaluation

This is a brief list of texts and reference works on land evaluation. The material presented in these is sufficient to understand the FAO framework, its applications to various land uses, and its relation to other methods of land evaluation.

Beatty, Marvin T., Gary W. Petersen, and Leslie D. Swindale (eds.) (1979) "Planning the uses and management of land", Agronomy Monograph 21, American Society of Agronomy, Madison, WI, USA. Large (1027 page) collection of articles on land use planning from the perspective of natural resource scientists. Emphasis on practical problems of land use in the USA.

Beek, Klaas Jan (1978) "Land evaluation for agricultural development". International Inst. for Land Reclamation and Improvement, Wageningen, publication 23. Subtitled "Some explorations of land-use systems analysis with particular reference to Latin America". This is Beek's doctoral thesis, and is quite wide-ranging.

Bridges, E. M. and Donald A. Davidson (eds.) (1982) "Principles and applications of soil geography". Longman, London. An advanced British undergraduate text, with excellent summary chapters on soil survey, soil classification, automated data handling, agricultural and urban uses of soils information. Good bibliographies.

---

[7]Proportional yield can also be determined by limiting or multiplicative yield factors, see Chapter 8.

[8]Physical suitability subclasses can also be determined by the maximum-limitation method, see Chapter 8.

[9]The use of dervied land characteristics is optional.

Burrough, Peter A. (1986) "Principles of geographical information systems for land resources assessment". Oxford Science Publishers, Monographs on Soil and Resouces Survey No. 12, Clarendon Press, Oxford. An introduction to GIS for natural resources scientists, with some land evaluation examples. Dense but worthwhile.

Davidson, Donald A. (1980) "Soils and land use planning". Longman, London. An advanced British undergraduate text on applications of soils information to planning. Excellent introduction to the literature.

Davidson, Donald A. (1992). "The evaluation of land resources". Longman Scientific, London. Update of the 1980 text, mentions ALES in passing.

Davidson, Donald A. (ed.) (1986) "Land evaluation". Van Nostrand Reinhold/AVI. A collection of original influential papers on land evaluation.

Dent, David, and Anthony Young (1981). "Soil survey and land evaluation". George Allen & Unwin. An advanced British undergraduate text, probably the best introduction to the subjects named in the title.

FAO (1984) "Land evaluation for forestry" FAO Forestry Paper 48, FAO, Rome. Specific guidelines for evaluating land for forestry, based on the FAO framework.

FAO Soil resources development and conservation service (1976) "A framework for land evaluation". FAO Soils Bulletin 32, FAO, Rome. The original statement of the "FAO Framework", this has been superseded by better and more comprehensive explanations in the later soils bulletins on the subject.

FAO Soil resources development and conservation service (1983) "Guidelines: land evaluation for rainfed agriculture". FAO Soils Bulletin 52, FAO, Rome. Specific guidelines for evaluating land for rainfed agriculture, based on the FAO framework.

FAO Soil resources development and conservation service (1985) "Guidelines: land evaluation for irrigated agriculture". FAO Soils Bulletin 55, FAO, Rome. Specific guidelines for evaluating land for irrigated agriculture, based on the FAO framework.

Hudson, Norman. (1981) "Soil conservation", 2nd edition. Cornell University Press. This practical text on soil conservation has a good chapter on the land capability approach to land evaluation. Many examples are from Africa.

van Lanen, H. A. J., Hack-ten Broeke, M. J. D. , Bouma, J., & de Groot, W.J.M. 1992. "A mixed qualitative/quantitative physical land evaluation methodology". **Geoderma** 55: 37. A good example of using the right tool for each job: ALES for a semi-quantitative rapid assessment of problem areas, then a simultion model fo more detailed assessment in critical areas.

van Lanen, H. A. J. and H. Wopereis. 1992. "Computer-captured expert knowledge to evaluate possibilities for injection of slurry from animal manure in the Netherlands". **Geoderma** 54: 107. More from the Staring Centre team.

León Pérez, J. C. 1992. "Aplicación del Sistema Automatizado para la Evaluación de Tierras (ALES), en un sector de la cuenca del río Sinú (Córdoba, Colombia)". **Revista C.I.A.F.** 13(1): 19-42. A physical suitability evaluation for a variety of tropical land uses in a semi-arid river basin. Results were transferred to the ILWIS GIS for spatial analysis and map display. Work of the "Augustín Codazzi" Geographical Institute of Colombia.

Klingebiel, A. A., and P. H. Montgomery (1961) "Land capability classification". USDA Agriculture. Handbook 210, Washington, DC. The source document for the land capability approach. Short and well-stated.

Siderius, W. (ed.) (1984) "Proceedings of the workshop on land evaluation for extensive grazing (LEEG)". International Inst. for Land Reclamation and Improvement, Wageningen, publication 36. Specific guidelines for evaluating land for extensive grazing, based on the FAO framework.

Siderius, W. (ed.) (1986) "Land evaluation for land-use planning and conservation in sloping areas". International Inst. for Land Reclamation and Improvement, Wageningen, publication 40. Specific guidelines for evaluating land in sloping areas, based on the FAO framework.

Sys, C. (1985) "Land evaluation". State University of Ghent (Belgium), International Training Centre for post-graduate soil scientists. Parts I, II, and III. This is the three-volume text used by Prof. Sys in his training course for professional soil scientists. Parts I and II deal with methodology; part III is the only publication where extensive crop requirement tables have been collected.

3.3.1 Availability

The FAO bulletins are available from the FAO publications representatives in each country. In the USA this is: UNIPUB, 4611-F Assembly Drive, Lanham MD 20706-4391, phone (301)-459-7666. Call them for stock numbers, prices, and availability. Some stock numbers are: Soils Bulletin 52: F2649, Soils Bulletin 55: F2798.

FAO has publications representatives in many countries, typically a bookseller in the capital. For a list of the representatives in each country, and a current list of the publications in-print, request the "FAO publications catalog" from: Distribution and Sales section, FAO, Via delle Terme di Caracalla, 00100 Rome, Italy; Telex 610181 foodagri.

UNIPUB is also the US representative for the ILRI publications. ILRI 23 is out of print.

The 3-volume Sys land evaluation text can be obtained from: ABOS, Marsveldplein 5, bus 57, B-1050 Brussel, Belgium.

The Oxford, George Allen & Unwin, VNR, Cornell, and Longmans texts are available through commercial booksellers.

Agronomy Monograph 21 is available from: American Society of Agronomy, 677 S. Segoe Rd., Madison, WI 53711 USA.

USDA Ag Handbook 210 is out of print, but is available in many libraries.

4 Who uses ALES?

This chapter explains the roles of the different people who use ALES or its products. The basic flow of information is as follows. First, *model builders* construct and validate ALES models for the set of Land Utilization Types to be considered in the study, working with a group of *land use experts*. Second, *model users* enter land data and ask the program to compute evaluation results. Third, *evaluation consumers* use the results of the evaluation to make their land use decisions. They might also use ALES in *consultation* mode. All the while, the *system administrator* ensures that the computer system, including ALES, is working correctly.

The following sections explain these roles in more detail.

4.1 Model builder

This is the person who constructs the *expert system* within the ALES framework. This person can be the land evaluation specialist, or a specialist in the use of ALES working with the land evaluation project team. There is no requirement that the model builder be a computer programmer or systems analyst; however, familiarity with the principles of systems analysis will certainly be helpful.

The model builder is often a natural resource scientist (e.g. soil scientist) charged with conducting a land evaluation. Since no scientist is expert in every field, the model builder must build a team of *land use experts* to help make the *decision decisions* as models are built. Since these experts will not be versed in ALES terminology nor land evaluation methods, the model builder must *translate* the expert judgments into a form which ALES can use (e.g., decision trees).

The model builder should be meeting with the eventual *evaluation consumers* from the very beginning of the project, to set objectives and priorities.

Model builders enter ALES with the DOS command **ales**[10]. Entering the program in this mode, they are able to change the model, as well as the database. During validation of the model they spend a great deal of time working in the 'Why?' screens. They will probably *print* reports showing the structure of the evaluation models.

4.2 Land use experts

These are the specialists in different land uses who are interviewed by the model builder to determine the land utilization types to be included in the model, the land use requirements of these, and the set of land characteristics that are needed to evaluate land qualities. They may be agronomists, foresters, soil scientists, hydrologists, economists, sociologists, climatologists etc.

Land use experts are not expected to use ALES directly, but they should review the models on-line with the aid of the model builder.

[10]Tutorials 3 and 4 explain how to use ALES in this mode.

4.3 Model user

Once a model is completed, the model builder can turn it over to the *model user*, who enters land unit definitions and data for all the land units in the study area, using the data entry templates designed by the model builder. The model user then asks ALES to compute the evaluation for all land uses and all land areas, and asks for *printed reports* of the results.

The model user is typically either a high-level clerk or a junior-level natural resource scientist.

Model users enter ALES with the DOS command **evaluate**[11]. Entering the program in this mode, they are *not* able to change the model, only the database and economic parameters such as prices and interest rates. Model users will probably *print* reports of evaluation results and the land unit database.

4.4 Evaluation consumer

The results of the evaluation are presented to the *consumers*. These include land use planners, farmers, farm advisors, in fact, anyone who has can make *decisions* which affect land use, and who has the *mean* to carry out those decisions.

For an evaluation exercise to be successful, it is *critical* that the potential consumers of the evaluation be involved *from the beginning*. They are the ones to set the *priorities* and *objectives* of the evaluation.

Evaluation consumers are not expected to use ALES directly, but the model builder should be prepared to demonstrate the program, including the 'Why?' screens, to the consumers, so that they can see how the models were constructed.

4.5 Consultation mode

An interesting use of ALES is *consultation mode*. This is like a visit to the doctor: the program asks for values of each relevant land characteristic in turn, until it has enough information to determine the suitability of the land unit for a Land Utilization Type.

To consult ALES, you enter with the DOS command **consult.**[12] In this mode, users can not change the model or economic parameters. However, they can record the values of the land characteristics that they entered during the consultation in a map unit in the database for later use, either in this mode or with the **evaluate** command.

4.6 System administrator

This person works with the operating system, performs backups, adds and configures new hardware and programs, and so forth. The system administrator is responsible for installing

[11]Tutorial 2 explains how to use ALES in this mode.

[12]Tutorial 1 explains how to use ALES in this mode.

(Chapter 5) and maintaining (Chapter 11) ALES. They may also be called on to *interface* ALES with other systems (Chapters 9 and 10).

The system administrator enters ALES with the DOS command **ales**, and may also use the DOS command **util** to maintain the MUMPS system. In the last resort they may have to enter the MUMPS system directly.

5 Installing ALES

(Revision date: 20-May-1992)

This chapter explains how to install and configure ALES on your computer system. It has been updated for the newest version (4.3) of DataTree MUMPS, which requires a special PaperKey Certificate for installation, as explained below.

The first section explains what kind of computer hardware and software you need in order to be able to install and run ALES.

The second section explains how to install ALES. We assume that you understand the basics of your operating system (e.g. MS-DOS) and how to switch between disk drives, move in the directory structure, and use the keyboard. For some installation options you must know how to use a text editor.

The third section explains how to customize ALES, including how to run it under the multitasking operating systems DesqView, Windows, and OS/2.

5.1 System requirements

CPU (microprocessor): ALES will run on any processor of the Intel 8086 family (i.e. the processors found in the IBM PC and its successors and compatibles) or functionally-equivalent processor. For reasonable performance we recommend at least a 12MHz 80286. No special features of the 80386 or 80486 processors are used by ALES even if these chips are present.

Coprocessor: A math coprocessor is not required nor used even if installed.

Conventional (DOS) memory: ALES requires a minimum 384Kbytes of free RAM, but will run faster with more free memory up to the DOS limit of 640Kb, since the MUMPS system will use the extra space for more disk buffers.

Expanded and extended memory: ALES does not need nor use these types of memory.

Hard disk: ALES runs only on hard-disk systems. It uses a minimum 3.0Mb of hard-disk space as installed for the MUMPS system, the ALES program and support files, and a database sufficient for about five evaluations. As additional evaluations are added, hard disk space is increased by 0.5Mb at each expansion. A fast hard disk and/or a disk-caching program (e.g., Microsoft's SmartDrive) can dramatically speed up ALES.

Diskettes: ALES only uses diskettes for installation, to save and restore models and databases.

Video: ALES writes to the monochrome (MDA or Hercules) or color (CGA or better) video screen, depending on which is selected in DOS. A graphics adapter (video card) is not required to run ALES, which uses only text mode video. If a color graphics adapter is installed, ALES can display its windows in various user-selected color schemes.

Printer: A printer is not required to run ALES, only to obtain printed results. ALES does not use any special printer features, so any text printer can be used.

Pointing device: ALES Version 3 does not respond to a mouse or other pointing device.

Operating system: ALES requires a computer capable of running one of the Microsoft operating systems MS-DOS or PC-DOS, V2 or later, or another operating system that emulates MS-DOS either directly (e.g. Digital Research's DR-DOS) or in a 'compatibility' mode (e.g. the 'vpix' program running on Unix/386, or OS/2).

Multitasking operating systems: ALES can run under several popular multitasking operating systems, such as DesqView V2.3 and Windows V3.1, but it is not written to take advantage of them (i.e. it does not use their windowing systems or multitasking). See the final section of this chapter for details.

MUMPS language and database system: ALES is written in the MUMPS programming language (Lewkowicz 1988), and requires Version 4 of the DataTree MUMPS language and database system (DataTree Inc., 1986) in order to run. Version 4.3 of DataTree MUMPS is supplied with the copies of ALES supplied after October 1992. With your ALES license you have also purchased a license from DataTree for their MUMPS system.

5.2 Installation procedure

There are two main steps to the distribution: (1) install the DataTree MUMPS language and database system, and then (2) install the ALES program.

ALES is distributed on two diskettes (3.5" or 5.25" according to customer preference), and comes with a special PaperKey Certificate which is necessary to install MUMPS. The first diskette contains DataTree MUMPS and the MUMPS *installation program* which you use to install MUMPS on your computer system. The second diskette is supplied by Cornell, and contains the ALES program as well as the ALES *installation script* (a batch file) which you use to install ALES, once the MUMPS system has been installated.

The installation diskettes are write-protected; after installation you should *keep them in a safe place*, because you may need them again in a few months or years if you have a catastrophic hard disk failure and need to re-install ALES. You may also want to make a backup copy of the distribution set of diskettes, which will save you the trouble of having to write us for a replacement, should the original set be lost or damaged. *Keep your PaperKey certificate along with the distribution disks* because you will need it if you ever re-install or move your ALES program to another computer or directory.

Keep your distribution disks and PaperKey certificate in a safe place

You must first decide where you want to install ALES in your hard disk directory structure. It is recommended that you install ALES in subdirectory '\dtm' on hard disk drive 'c:', as the installation procedure is designed for this special case. If you do this, the installation procedure will be very easy. If you elect to install on a different hard disk or in a different subdirectory, you will have to perform some additional steps (explained below) not included in the standard installation procedure before starting ALES for the first time.

It is fairly easy to install ALES on a different drive from 'C:' and/or on any path whose last directory is 'dtm'. It is more difficult, but not impossible, to install ALES to a path whose last (lowest-level) directory is not 'dtm'. However, both of these cases are explained below.

5.2.1 Installing DataTree MUMPS

Before you can install ALES, you must install the MUMPS system, as follows.

1. Insert the DataTree distribution disk in a diskette drive.

2. At the DOS prompt, change to that drive (usually 'A:' or 'B:') so that it is the active drive, by typing the drive letter followed by a colon (for example 'A:') and pressing the ENTER key.

3. At the DOS prompt for the diskette drive (typically 'A:>' or 'B:>' depending on which diskette you selected), type the command 'install' and press the ENTER key.

4. Follow the instructions on the screen, answering the questions as follows:

   √ In the *first* screen, press ENTER to continue.

   √ In the *second* screen, accept the default of installing DTM-PC standalone (this option should be checked), by pressing ENTER.

   √ In the *third* screen, answer N when you are asked if you wish to load some DataTree source code files.

   √ In the *fourth* screen, answer Y to continue with the installation.

   √ In the *fifth* screen, select the hard disk drive where MUMPS (and later ALES) should be installed. This should have about 3Mb (3,145,728) free space as a minimum.

   √ In the *sixth* screen, specify the path. If possible, accept the suggested path C:\DTM without changes. If you must change the path, if possible make sure that the final directory name in the path is 'DTM', for example '\MODELS\LECS\DTM'.

   Note: if you really insist on installing MUMPS and ALES to a path that does not end with 'DTM', you will have to edit the installation script INSTALL.BAT on the ALES distribution disk accordingly, changing all occurrences of 'DTM' to the directory name of your choice, before continuing with the installation of ALES.

   √ Ignore the message on the *final* installation screen that tells you how to use DTM-PC in programmer mode. Press ENTER, the installation of MUMPS is complete.

5.2.2 Installing ALES

Once you have installed DataTree MUMPS, you are ready to install ALES V4.

1. *Insert* the ALES distribution disk in a diskette drive.

2. *Change* the DOS prompt (using command 'cd')to that drive.

3. *Run* the command **install** with one argument: the *drive* (and possibly the path) where you installed MUMPS, not including the final (lowest-level) directory name 'DTM'.

If you installed MUMPS to C:\DTM as recommended above, you would simply type:

```
install c:
```

If you installed MUMPS to a drive and/or path other than 'C:\DTM', enter the drive and path *not* including the final '\DTM'. For example, if you installed MUMPS to drive D:, directory \MODELS\LECS\DTM, you would issue the command:

```
install d:\models\lecs
```

As another example, if you installed MUMPS to drive D:, directory \DTM, you would issue the command:

```
install d:
```

Finally, if you installed MUMPS to drive C:, directory \MODELS\LECS\DTM, you would issue the command:

```
install c:\models\lecs
```

If there is a problem during the installation, the script will tell you what went wrong.

If you have installed MUMPS and ALES in directory '\DTM' on hard disk drive 'C:', you should skip the rest of this section.

If you did not install MUMPS and ALES in directory '\dtm' on hard disk drive 'c', you will have to make some changes to the MUMPS database description file, 'mumps.dbd', which was installed in the main subdirectory ('...\dtm') of the MUMPS program, before you can run ALES. As distributed, this file is essentially as follows:

```
;ALES V4/DTM V4.3 MUMPS.DBD - Database description file
;Last update: DGR 20-May-92
;
;datasets
;
;program object code
DATASET C:\DTM\ALES4\ALES4-RT
;program support files, except text
DATASET C:\DTM\ALES4\ALES4-GB
;text databases (1 per language)
DATASET C:\DTM\ALES4\LTX4E-GB
DATASET C:\DTM\ALES4\LTX4F-GB
DATASET C:\DTM\ALES4\LTX4S-GB
;ALES user evaluations
DATASET C:\DTM\EVAL\EVAL-GBL

;namespaces
;program
NSPACE ALES:  ALES4-RT, ALES4-GB
;users
NSPACE EVAL:  ALES4-RT, EVAL-GBL
MAP ^Lform TO            ALES4-GB
MAP ^Lmenu TO            ALES4-GB
MAP ^Ltxt  TO            LTX4E-GB
MAP ^Ltxt("f")           TO    LTX4F-GB
MAP ^Ltxt("s")           TO    LTX4S-GB
MAP ^Lconfig("version")  TO    ALES4-GB
MAP ^Lconfig("mode")     TO    ALES4-GB
END
```

Edit this file with any pure text editor (one that doesn't insert hidden formatting codes), such as the DOS 5 editor or EDLIN, or with a word processor in 'text-only' mode. *Find* the lines that begin with the word 'DATASET'. *Change* the disk and path name prefix in these lines (which as distributed is 'C:\', i.e. the root directory of disk 'C:') to the drive and directory where you actually installed ALES. Make sure to edit the first comment line (beginning with the semicolon ';') to indicate your changes. Don't alter any other lines.

For example, if you elect to install ALES in directory '\MODELS\LECS\DTM' on hard disk 'D:', the database description file would look like this:

```
;ALES V4/DTM V4.3 Database definition - for PC/AT-386#3
;edited by xyz, 30-Jul-1993
;
;datasets
;program object code
DATASET D:\MODELS\LECS\DTM\ALES4\ALES4-RT
;program support files, except text
DATASET D:\MODELS\LECS\DTM\ALES4\ALES4-GB
;text databases (1 per language)
DATASET D:\MODELS\LECS\DTM\ALES4\LTX4E-GB
DATASET D:\MODELS\LECS\DTM\ALES4\LTX4F-GB
DATASET D:\MODELS\LECS\DTM\ALES4\LTX4S-GB
;ALES user evaluations
DATASET D:\MODELS\LECS\DTM\EVAL\EVAL-GBL

... the rest of the file is not changed

END
```

5.2.3 Running ALES for the first time

The very first time you start DataTree MUMPS or any program that uses it (such as ALES) after installating DataTree MUMPS, you have to enter your special PaperKey code. The Paper Key is your proof of ownership of a license from DataTree (for MUMPS) as well as Cornell (for ALES).

You must enter your special PaperKey code the *first time* you start ALES or DataTree MUMPS.

Locate your paper key, which is part of your license document from DataTree. Connect to the directory where you installed MUMPS and ALES, and type the command:

```
ALES
```

and press the ENTER key. You will see the message:

```
Please enter the license registration information exactly
as shown on your PaperKey Certificate:

   KEY:
```

Enter the paper key code **exactly as it appears** including punctuation and spaces.

If you enter the key correctly, MUMPS will next prompt you for the *supplier name*. Enter this as it appears on the paper key certificate. This is usually CORNELL UNIVERSITY SCAS.

If you enter the supplier name correctly, MUMPS will next prompt you for the *end user* name. Enter this as it appears on the paper key certificate. This is usually your organization name.

If you enter all three items correctly, ALES will start. The first time ALES starts up slowly, as it is expanding some of its data structures.

NOTE for those with older PCs: For MUMPS to run, the *correct date and time* must be known to the operating system. On later-model computers, or those with an add-in board containing a real-time clock, this is done automatically at system startup. On older PC's, or on machines will malfunctioning clocks, however, you must *enter the date and time* when prompted by the operating system at system startup. If the operating system thinks that the current date is earlier than the creation date of the MUMPS program, MUMPS can not run, and you will see the message 'license expired' when you try to start ALES.

5.2.4 Moving ALES to another location

If you move ALES to another computer system (for example, a newer computer or a computer at another office) or to another disk or directory on the same system (for example, to rearrange the files on your hard disk), you will have to *re-enter* the PaperKey information the next time you start ALES or MUMPS. You don't have to go through the whole installation procedure, you can copy MUMPS and ALES with the DOS 'xcopy' command, or with a file manager such as XTree, or even with the DOS 'backup' and 'restore' commands, but MUMPS and ALES won't run in the new location until you re-enter the PaperKey information.

Each copy of DataTree MUMPS and ALES is licensed for use on one computer system at a time. You may transfer ALES to another system or even leave it installed in several systems, as long as thre is *no possibility* that more than one copy can be used at the same time.

If you move ALES to another system, you can leave ALES installed on the original system if you *remove the key* so that ALES won't run. You remove the key as follows:

1. Start MUMPS with the command 'MUMPS /KEY'. DTM responds by requesting a new paper key.

2. Enter the word DELETE. DTM responds with the message 'license deleted'. You can now use the key in the new location.

If you move ALES to another disk or directory on the same system, you can leave ALES installed on the original disk or directory and leave the key, so that you can run ALES in either location. This is OK because you can't run both copies at the same time under a single-task operating system.

NOTE *If you move ALES to another disk or directory* on the same or another system, you will have to *edit* 'MUMPS.DBD' as explained in §5.2.2, above, so that the DATASET statements point to the data files in their new location.

5.2.5 Running ALES from DOS

We have provided three one-line batch files, located in the directory where you installed the MUMPS program and ALES. that will start ALES in the various modes mentioned in Chapter 4, as well as perform system maintenance functions. The commands are:

√ **ALES**: all ALES functions are available, including model building. This is intended for the *model builder*.

√ **EVALUATE**: all ALES functions except model building, and the ability to delete models and restore them from a backup, are available. This is intended for the *model user*. Land data can be entered or imported, evaluations can be computed, and reports printed. Economic parameters, including prices and interest rates, can also be edited.

√ **CONSULT**: enters consultation mode directly, to consult one or more ALES models about the suitability of a single tract for a single use. Land data can not be altered. This is intended for the occasional model user and for demonstrations.

√ **UTIL**: runs system maintenance procedures, see Chapter 10 for details.

√ **RMDTM**: delete the MUMPS system.

You may wish to include the main ALES subdirectory on your DOS search path, to allow you to start ALES from any directory. ALES should start within 30 seconds on a basic IBM-PC, and within 10 seconds on a fast IBM-PC/AT.

5.3 Customizing ALES

You can change the way ALES looks or performs, as explained in the following sections.

5.3.1 Changing screen colors

ALES runs on all PC video adapters, including the Monochrome Display Adapter (MDA), Color Graphics Adapter (CGA), Professional Graphics Adapter (PGA), Visual Graphics Adapter (VGA), and the Hercules graphics adapter. On the EGA and subsequent improvements such as VGA, ALES runs in CGA-compatibility mode.

ALES is distributed with a set of pre-defined color schemes, and in Menu 1.1.6 ('Presentation Options'), Options 1-5 can be used to select color schemes for five types of screens:

- the *main* screen
- the *banner* (top) and *annunciator* (bottom or 25th) lines,
- the *help* screens
- *consultation* mode, and
- *annotations*.

If you are running on a *monochrome* or *Hercules* adapter, you should *set all these to the monochrome color scheme* (this is the default in a distribution copy of ALES).

On a *color* monitor, you can select different schemes until you find ones that are pleasing to you. (The color scheme names are place names from central New York state and northern Pennsylvania, USA.)

5.3.2 Changing the human language

ALES *models and databases* can be created in any human language that can be represented by the IBM character set; the language for these is determined by the model builder. The ALES *program* itself contains many texts, such as menus, data entry forms, prompts, and help screens. These were originally written in the English language, but most of them have been translated into French and Spanish. You can change the language in which ALES displays its own texts with Menu 1.2 ('System Options'), Option 6. Any texts that have not yet been translated will be displayed in the original English.

If you don't need one or both of the non-English languages, you can save some hard-disk space by deleting them from your system. To do this:

1. Delete the files 'LTX4F-GB.*' (French) and/or 'LTX4S-GB.*' (Spanish) from the subdirectory 'DTM\ALES4'.

2. Edit the database description file MUMPS.DBD with a plain-text editor, placing a semicolon ';' as the first character of all lines which make reference to the file which you just deleted; this effectively removes the line since all lines beginning with a semicolon are treated as comments by MUMPS and not processed on system startup.

    These lines begin with the words 'DATASET' and 'MAP'.

The next time you start ALES, the language that you deleted will not be available for the program texts. However, any models written in French or Spanish will of course still be in their original language.

5.3.3 Changing the MUMPS system configuration

DataTree MUMPS (and therefore ALES) as delivered is configured to recognize a parallel printer (DOS device PRN), the PC video screen, two serial ports (COM1 and COM2 configured for 8 bit characters, no parity, one stop bit, and flow control enabled, running at 9600 baud), and DOS files. If you have additional peripheral hardware, or want to change the default communications parameters of the pre-configured ports, you must edit the device description file 'MUMPS.DEV'. Follow the instructions in §7.2.5 of the DataTree MUMPS manual; which section is included with the ALES distribution.

5.3.4 Running ALES under DesqView

ALES can run under the DesqView[13] multitasking system Version 2.x, which is recommended for the MS-DOS user who wants to run several programs at once, and who has a computer with several megabytes of expanded or extended memory (note that the companion product QEMM is

[13]DesqView and QEMM are registered trademarks of Quarterdeck Office Systems.

required on 386 or better computers to make best use of extended memory). The following explanation is intended for the experienced DesqView user.

You must set up a DesqView menu item for ALES as follows. The program to run is the batch file 'ALES' in directory 'C:\DTM' (or wherever you installed ALES), and the data directory is the same. ALES requires a 384Kb partition, and can use up to 640Kb. It writes text directly to the screen but does not display graphics. It may use serial ports; it does not need diskettes. It does not require or use any extended or expanded memory, and it uses its own color schemes, not DesqView's. On a 80386 processor or better you can elect to virtualize text, so that you can work on ALES in a small window. Otherwise, when the ALES window is made small, you can only scroll its contents, but not type or receive output. ALES can run in the background on a 80386 or better; this is a nice feature for long computations, data import/export, or printed reports.

> ** **Important note** ** *Never try to run more than one ALES session at once under DesqView.* It probably won't work and if it does you run a serious risk of corrupting your evaluations.

5.3.5 Running ALES under Microsoft Windows

ALES can run under the Microsoft Windows system Version 3.1, in either standard or enhanced mode. ALES does not take advantage of any Windows features. The following explanation is intended for the experienced Windows user.

> ** **Warning** ** *Serious system crashes* **have been reported when attempting to run ALES under Windows 3.0 or before, especially on networked computers. Try this at your own risk. Version 3.1 is much more reliable.**

You can start ALES directly from the 'MS-DOS prompt' window, or even by using the 'File, **Run**' command (on 'ales.bat' etc.) from the Program Manager or File Manager. However, if you commonly run ALES under windows, it is worthwhile creating a '.PIF' (configuration) file, and adding an icon to the Program Manager.

The '.PIF' (configuration) file for ALES should have the parameters: Video mode "text", 384 Kb conventional memory, 0Kb XMS memory, Directly modifies "COM1" and "COM2" (the first two serial ports), "Close on exit", "Background". You can choose to run full-screen or in a window (in which case, you might experiment with the 'Fonts' setting on the window's 'control' menu).

ALES can run in the background under Windows enhanced mode (which requires a 80386 or later processor); this is a nice feature for long computations, data import/export, or printed reports.

As with anything having to do with Windows, be prepared to experiment.

> ** **Important note** ** *Never try to run more than one ALES session at once under Windows.* It probably won't work and if it does you run a serious risk of corrupting your evaluations.

5.3.6 Running ALES under OS/2

ALES supposedly runs under the OS/2 operating system Version 2.0 or higher, in the DOS 'compatability box'. The following explanation is intended for the experienced OS/2 user.

DataTree suggests that you adjust the amount of memory available to the DOS box by setting the OS/2 CONFIG.SYS parameter RMSIZE to either 512 or 640. We do not have any experience with this at Cornell, if you try it good luck and let us know how it worked.

> ** **Important note** ** *Never try to run more than one ALES session at once under OS/2.* It probably won't work and if it does you run a serious risk of corrupting your evaluations.

6 Running ALES

ALES has been designed to be easy to operate. The intellectual activity of constructing models and collecting data is difficult enough; interacting with the computer should not cause the land evaluator further difficulties. This chapter presents a brief summary of how the ALES user interacts with the program, and problems that might be encountered..

6.1 Starting ALES from DOS

There are three different ways to start ALES from DOS, each intended for a different type of user. In all three cases, ALES is invoked by typing the name of a DOS a *batch file* in the subdirectory in which ALES is installed. The three programs are:

ALES: all ALES functions are available, including model building, data entry, computation of evaluations, and reports. This is intended for the *model builder*.

EVALUATE: all ALES functions, except model building and the ability to delete models and restore them from a backup, are available. This is intended for the *model user*. Land data can be entered or imported, evaluations can be computed, the evaluation matrix and 'why?' screens can be displayed, and reports printed. Economic parameters, including prices and interest rates, can also be edited.

CONSULT: enters consultation mode directly, to consult one or more ALES models about the suitability of a single tract for a single use. Models can not be modified. Land data from a consultation can be recorded in the database. This is intended as an easy way to consult an ALES model for the *occasional model user*. It provides a simple way to determine the suitability of *one* land area at a time for *one* land use at a time.

Note: if you see the message 'license expired' when trying to start ALES, see §5.2.3.

6.2 Interacting with ALES

The operation of ALES should be fairly self-evident. There are no commands to memorize and type; the entire interaction is controlled by *function* keys (the ten keys marked 'F1' through 'F10'), the *special* keys 'Esc' and 'Return', and the *cursor movement* keys (the arrows and associated keys such as 'Home' located on a separate keypad). The only time you need to use *regular* (typing) keys is when you must enter information, such as a data value.

The screen usually displays multiple overlapping windows. As you make choices and perform actions, new windows appear and remain as long as they are relevant. Parts of all previous windows in which you made a selection are still visible underneath the active window. However, that only the frontmost window is *active*. That is, your actions only affect the window in front of all the others. To show this, the front window's borders are highlighted.

Note: The windows in ALES are not like those in Microsoft Windows or other event-driven environments, because you can not move to any visible window. Instead, you have to complete the actions in the order in which the windows are presented to you.

The order in which windows are displayed may be found in Appendix 1 'Program Structure'.

The basic pattern of interaction with ALES is "*move* and *do*". You *move* the highlight to a selection and then you *press* a function or special key to perform an action.

6.3 Function keys, cursor control keys, typing

At every point in the program, an *annunciator* line at the bottom of the screen shows which keys are active. The meaning of the keys varies with the context, however, certain keys are almost always reserved for certain functions:

F1 and **F2**: *Help*: At almost every point in the program you can receive on-screen help by pressing the 'F1' key, the 'F2' key, or both. Generally, the 'F1' key displays information about the mechanics of the interaction, for example, what keys are active, and how to move the highlight. This is called *form help*. The 'F2' key, by contrast, is usually *content help*: it displays information about what is being requested. In the 'Why?' screens, the 'F2' key is the 'Why?' key.

F9: *Notes*: If the model builder wrote an annotation (note) for the currently-active entity, this key will display it. Items in lists that have associated notes are marked with small 'degree' signs: °.

F10: *Completes* an interaction.

Return: (may be marked **Enter** or ↵.): *Confirms* a selection or a typed entry.

Esc: *Cancels* an interaction, goes back..

The *cursor control keys* are used to move around in selection boxes, among questions in a form, and in the decision tree editor. The *arrow* keys (e.g. '↓') move one line or cell in the appropriate direction. The **Home** and **End** keys move to the first and last items in selection boxes. The **PgUp** and **PgDn** move up and down one screenful in selection boxes that are too long to fit on a screen, and scroll help texts and 'why?' screens in the appropriate direction.

6.4 Leaving ALES

You can only leave ALES, and return to the operating system, from the system (first) menu. You can return from any point in the program to this menu by pressing 'Esc' (to cancel) or 'F10' (to confirm) enough times. Then press 'F10' or 'Esc' and you will leave ALES and return to DOS.

Pressing the 'F10' key enough times will always get you out of ALES.

If you get really stuck, or want a *fast way out of the program*, press **Ctrl-C** (hold the 'Ctrl' key down while you press the 'c' key) at any point in the program. You will be asked if you really want to go back to the opening screen. If you answer 'no' you will be left in the same place in the program as when you pressed Ctrl-C. If you answer 'yes', you will return to the opening screen, from which you can exit the program.

Never, ever, turn off the computer or reboot DOS while ALES is running. This may corrupt the MUMPS database and may force you to *reinstall* ALES from the original distribution diskettes.

6.5 Program errors

You may encounter a *program error* while running ALES. These can be caused by programming errors, by a model which exceeds the capacity of the program (unlikely), or combinations of circumstances unforeseen when the program was designed, as well as by limitations of the computer system, such as a full disk.

If an error occurs, an *error message* will appear in front of the other windows, giving the internal (MUMPS) name of the error and its location in the program. If you wish, you can copy the message and contact the ALES project, along with information about the circumstances in which the error occurred. You can try to get around the error by making sure your model is reasonably complete and contains reasonable values; often problems occur when unforseen data values exceeed the capacity of the system.

Some kinds of errors call for action on your part. In particular, if you see a '*Disk Full*' error, this is because ALES has attempted to allocate more disk space for an evaluation model or data, and was not able to because at least 512Kb was not free. You must free up at least this much disk space before restarting ALES.

After you are done looking at the error message, press any key and you should return to the opening screen, i.e. the list of evaluation models.

In very rare circumstances MUMPS may 'freeze' the computer system. If the computer does not respond to any keystrokes, *wait until all disk activity stops* (i.e. the 'disk active' light is off) for a several seconds and try again.

Usually the 'freeze' is caused by MUMPS expanding or restructuring its database. This occurs at unpredictable times, when the ALES database becomes too big for the existing MUMPS database. Then MUMPS allocated 0.5Mb more disk space and restructures its files. This takes up to one minute on a slow computer. However, during this lengthy process, the *disk-use light* on your computer will be mostly *on*, and after it is done, the ALES program 'unfreezes' and you can continue as normal. If, by contrast, the disk-use light is off for a long time, and the computer still does not respond to any keys including Control-C, reboot and immediately check the dataset integrity, as explained §11.3.

6.6 System integrity

In some circumstances MUMPS may not be able to shut down correctly. This may happen in the 'freeze' situation discussed above, or, more commonly, due to unexpected loss of power to the computer. In these situations, the next time you start ALES, you will see the following message:

```
** Warning: MUMPS not shut down correctly
** Verify dataset integrity before proceeding
```

After a few seconds the opening ALES screen will appear. You can try to use ALES, and if no futher errors occur, the dataset is most likely intact. If you experience further problems, you should immediately exit from ALES and ask the system manager to verify the datasets as explained in §11.3.

7 Learning ALES

This chapter presents four tutorials that have been designed to teach you how to run ALES. All ALES users should start with Tutorial 1, which explains how to interact with ALES. ALES users interested in building or modifying ALES models should continue with Tutorials 2, 3, and 4. Those whose only interest in using ALES is to evaluate using existing models need only to complete Tutorial 2. The first and second tutorials are slow-paced; the last two assume that the reader is comfortable with ALES.

Throughout these tutorials:

√ special keys, such as function keys, which you press are written in **bold** type, for example **F1**.

√ When keys are referred to by way of explanation, they are enclosed in single quotes, for example, the 'F1' key.

√ When you are directed to do something, the verb will be in *italics*, for example, 'now *type* the following:'.

√ Keywords will also be in *italics*, for example, 'the *land map units*:'.

√ Keys other than special keys that you press are written in `Courier` type, for example: `Type This Text.`

7.1 Tutorial 1 - consulting a model

The *objectives* of this tutorial are

(1) to teach you the *mechanics* of using ALES, including the use of the letter, function and cursor control keys, and

(2) to teach you how to *consult* an existing ALES model to determine the suitability of a particular land area for a single land use.

In addition, some of the model builder's reasoning will be explained in notes like this.

Note: This tutorial does not explore all the capabilities of ALES' 'consult' mode.

This tutorial should take about an hour to complete.

In this tutorial *you will learn to*:

- start ALES in "consultation" mode
- interact with ALES using the keyboard
- follow decision trees to obtain land quality values
- view the results of a consultation on the screen
- end the consultation

Before working on this tutorial, *you must know how to*:

(1) *interact* with the microcomputer's *operating system* (usually PC-DOS or MS-DOS);

(2) how to *name files* and *specify disk drives and directories*;

(3) how to use the *keyboard.*

If you can use any other program that runs under DOS, such as a word processor or spreadsheet, you already know everything necessary to start with ALES. Otherwise, you should follow a tutorial on DOS, such as 'Exploring the IBM Personal Computer', which is supplied with the IBM PC, or a tutorial on a specific program.[14]

7.1.1 Starting ALES in 'consultation' mode[15]

To start ALES in consultation mode from DOS, first change to the directory where ALES is installed. Typically this will be in directory '\dtm' on disk 'c:'. Assuming this to be the case, at the DOS prompt *type* `cd c:\dtm` and *press* the **Return** key.[16]

(If you are already working on disk 'c:', you can omit the 'c:' from this entry. If you are all already connected to this directory, you can omit this command altogether.)

Then start ALES itself by running the batch file **consult.bat**: at the DOS prompt *type* `consult` and *press* **Return**.

At this point you should see the *initial* ("welcome") *screen*, which includes the program name, your organization's name, and the serial number. *Press any key*, for example, **Return**, to continue beyond this screen.

You should now be looking at a list of one or more *evaluation models* on the right of the screen and an annunciator line at the bottom of the screen.

7.1.2 The basics of interacting with ALES

Now we must learn the basics of interacting with ALES. Almost everything is done with special keys, that is, keys that aren't used for typing text. These may be considered in three groups:

(1) *Function* keys. These are marked 'F1' through 'F10', and are located either on *top* (IBM extended) of or to the *left* (IBM PC) of the keyboard. These keys are used to perform actions on selected items. Their use changes throughout the program, but one key's action is quite

---

[14]Word processor tutorials, such as 'Learning Microsoft Word: Essentials for Keyboard' are usually very good in this respect

[15]This section assumes that you want to start ALES directly from DOS. You may also have installed ALES under a multitasking operating system such as Microsoft Windows or DesqView, or as an item in a menu system such as the MS-DOS shell, or you may normally start your programs from a shell such as the Norton Commander or Xtree Gold. In this case you already know how to run 'consult.bat'.

[16]This key is variously called **Return** or **Enter**. It may be marked with a square arrow,↵

consistent: the 'F10' key is used to complete an interaction, i.e. when you are *done* at any point in the program.

(2) *Cursor motion* keys. These are marked with directional arrows '←', '→', '↑', and '↓', or with descriptive names such as 'Home' and 'PgUp', and are usually located to the right of the keyboard. On some keyboards the arrow keys are arranged in a T-shape in their own area, and on other keyboards they are part of the numeric keypad.

(3) *Selection* keys. The most important of these is the 'Return' key, which is located at the right of the main keyboard, and may be marked with a backwards L-shaped arrow which looks something like: ↲. This key may also be marked 'Enter'. In general, it is used in ALES to make (*confirm*) a selection or choice. Also used extensively in ALES is the 'Escape' key, whose location on the keyboard is not standard. This may be marked 'Esc'. In general, this key is used in ALES to *cancel* an interaction, i.e. if you don't want to complete the interaction.

(4) The *typing* keys, i.e. the ones with letters and numbers on them.

With this background, let's go through the first interaction. This is selecting the evaluation model we want to consult. The right side of the screen should now contain a selection box on the right, showing several lines[17]. Each line consists of an *evaluation code* (on the left) and the *descriptive name* which applies to the evaluation code (on the right). At the bottom of the screen is the *annunciator line*. This shows the active keys, along with a brief mnemonic to show their action.

Notice on the *annunciator* (bottom) line of the screen is that the 'F1' and 'F2' keys can be used to get help. This is true at almost every point in the program.

√ 'F1' is used to explain the mechanics of an interaction; i.e., how to interact with ALES. In ALES this is called *form help*.

√ 'F2' is used to explain what is required, i.e. what the program is asking for. In ALES this is called *content help*.

Press **F1**. You should now be looking at an explanation of the keys that you can use. Notice that the annunciator line has changed to show a new set of active keys, this time for the help interaction itself. Also notice that the lower right corner of the help display says '...<more>...'; this indicates that there are more lines that can't fit in the box. You can display these lines by using the down arrow or 'PgDn' keys; move back up with the up arrow or 'PgUp'.

Now leave the help and return to the original interaction by *pressing* either **F10** (since you're 'done' looking at the help) or **Return** (since you're 'confirming' the help) or **Esc** .(since you're 'canceling' the help request).

Now *press* **F2** to see the content help. This text all fits in one display, so only the 'Return', 'F10', and 'Esc' keys are active. *Press* any one of them to return to the list of evaluation models.

---

[17]If there is only one line, you will not be able to move the cursor, only select the single evaluation.

For this exercise we will use the model named 'Tutor1', which is a very simple evaluation with only one Land Utilization Type.[18] The aim here is to lead you as easily as possible through a sample ALES evaluation, not to provide a comprehensive evaluation.

Find the line in the list of evaluations with the code (on the left) 'Tutor1'. Notice that it has a descriptive name (on the right) of 'ALES V4 Tutorial 1 : Simple Consultation'. This is the line we want to select.

> If 'Tutor1' does not appear in the list of evaluation models, you must first leave consultation mode (press **Esc**), and *load* it into your list of ALES models. The saved model 'tutor1.als' is included in the archive 'examples.zip' on the distribution diskette. Extract it to any convenient place on your hard disk with the 'unzip' program (also found on the diskette), start ALES with the DOS command 'ales', and then load 'tutor1.als' into the list of evaluations by using 'F8' when that list is displayed.

There are several ways to select one item out of the list; to begin with we'll use the simplest way, namely, moving the highlight with the cursor motion keys. *Press* ↓ , and you will see the highlight move from the first line to the second. Press↑ and you will see the highlight move back to the first line. Now *press* ↑ and ↓ as necessary to position the highlight on the line with 'Tutor1' as its code.

Throughout ALES, the model builder is able to leave *annotations*, or notes, for model users, such as you, to look at. These explain the evaluation as a whole or any part of it. Whenever a note is present, you will see the degree sign, '°', to the right of the code. Looking at the list of evaluations, you should see a '°' to the right of the code 'Tutor1'. So the model builder has left a note for you. Let's look at it now. Notice that on the annunciator line shows that 'F9' is marked 'Note'; also when we pressed 'F1' above to get form help, the help text told us that 'F9' could be used to read annotations. So, *press* **F9** now to read the note.

You should now be looking at the note. It gives some background about the entire evaluation model. When you're done reading it, *press* one of the **Return, F10**, or **Esc** keys to return to the list of evaluation models. The highlight should still be on 'Tutor1'.

Now press **Return** to select the highlighted evaluation, namely 'Tutor1'.

The list of evaluations should now disappear, and the code of the evaluation model you selected (namely, 'Tutor1') should appear in the upper right corner of the screen (the banner area). The upper right corner of the screen has the land utilization type code, in this case 'ccc' (which stands for conventional continuous corn)[19] In the center of the screen should be a *dialog box*, asking if you want to start with data from an existing map unit definition. The default answer, 'No' is *highlighted. Press* **Return** to confirm the 'No' answer, in other words, to enter all the data for the consultation directly from the keyboard (You would answer 'Yes' if you had saved land unit data at the end of a previous consultation.)

---

[18]This example is a simplification of the more realistic New York State example used in Tutorial 2.

[19]In this example we have only defined one land utilization type; if there had been more, you would have been asked to select one.

The dialog box should now disappear, and the first decision tree box should appear.

7.1.3 Following decision trees to obtain land quality values

When ALES computes the suitability of a map unit, it begins by computing *land quality* values corresponding to the set of *land use requirements* that the model builder defined for a land utilization type. Corresponding to each land use requirement is a *land quality*, which is the attribute of the land that is to be matched with the requirement.[20]

In consultation mode, you follow this computation one step at a time. ALES asks you for the data that it needs in order to compute the first land quality value, then the second, and so forth until all land quality values have been computed. Once ALES has determined all the land quality values, it computes overall suitability ratings.

The way in which ALES asks you for data in consultation mode is to lead you through each *severity level decision tree* in turn, one for each land use requirement. On the screen now is a large window with the title "LQ e (erosion hazard)". ALES is now going to determine the severity of the erosion hazard, on a scale of 1 to 4[21], for your particular land area.

In consultation mode, you only see as much of the decision tree as necessary. Right now you are looking at the *top level* of the tree, i.e. the *land characteristic* that the model builder thought was most important to determine the severity level of the land quality 'erosion hazard'. This first land characteristic is the slope of the land, and as you can see, it is measured in five classes: A, B, C, D, and EF[22]; the highlight is by default on the 'unknown' choice in the tree, the branch marked **?**, since you haven't yet selected a slope class. Once you have selected a slope class, other factors may be considered.

> Slope affects erosion hazard because the energy of runoff is greater on steeper slopes.
>
> The model builder constructed the tree considering local conditions and the availability of data, while trying to keep the tree as simple as possible. When we consider how many factors can affect erosion hazard, we see that the tree could become excessively complicated. However, when the evaluation is restricted to a certain area, much of the variability is implicit in the context, and need not appear in the tree. For example, in clayey soils, the type of clay (kaolinitic, montmorillonitic, illitic etc.) can have a great effect on erodibility, but in the area of New York State for which this model was constructed, clays are predominantly illitic; hence the type of clay need not be included in the tree, as it is a constant, not a variable. This theme is expanded in Chapter 8.

Press **F1** to bring up a help text explaining how to select one of the branches of the tree. As you can see, the arrow keys can be used to move to the correct branch. *Press* **Return** when you're done looking at the help.

---

[20]This is just a semantic difference: a land *use* has *requirements*, whereas a land *area* has *qualities* which match the requirements.

[21]The fact that there are four severity levels of erosion hazard is part of the model, not of ALES.

[22]Again, this is part of the model, not of ALES

Suppose that the land we are evaluating for corn production is sloping, i.e. in slope class C. Move the highlight to the correct branch by *pressing* the up arrow, '↑', three times. Branch 3 ('C' slopes) should be highlighted. *Press* **Return** to confirm this value.

Now the slope classes disappear, and next to the 'slope' name is the class we selected, namely, 'C'. ALES is now considering a sloping land unit. Under the header is the name of the next land characteristic to be considered, namely, 'perm', the permeability of the upper subsoil, which as you can see is measured in seven classes. That is, the model builder decided that, on sloping lands, the permeability of the upper subsoil should be the second factor to consider when determining the erosion hazard.

> Permeability of the upper subsoil affects erosion hazard, because it determines how fast water will infiltrate into the profile. If infiltration is slow, water will pond on the surface and run off. In this area of New York State, infiltration rates are usually limited by the subsoil, not the surface infiltration.

Suppose that the land area of interest has a moderately permeable subsoil. So, the highlight has to be moved to branch 4, 'm'. This time we'll use another method of moving the highlight: typing the branch number. *Type* the number 4; the highlight should move to the correct branch. *Press* **Return** to confirm this value.

Now the permeability classes disappear, and next to the 'perm' name is the class we selected, namely, 'm'. ALES is now considering sloping maps units with moderately permeable subsoils. Under the header is the name of the next land characteristic to be considered, namely, 'text-A', the USDA texture class of the surface soil, which as you can see is measured in the standard 12 classes.

> Texture of the surface soil affects erosion hazard, because certain particle sizes are easier to detach by the impact of rain or runoff and to transport by runoff. In general, silts and very fine sands are the most susceptible.

Supposing that in this land area the soils have a silt loam texture; *move* the highlight to the appropriate branch (number 5 for class 'sil', you can use either the arrows or type the class number). *Press* **Return** to confirm this value.

Now the texture classes disappear, and next to the 'text-A' name is the class we selected, namely, 'sil'. Under this is the *result*, in this case the severity level, or land quality value, for the land quality 'erosion hazard'. In the situation we have just described to ALES, namely, a silt loam with moderately permeable upper subsoil on a C slope, the resultant erosion hazard is in class 3 (of 4), namely, a 'moderate' erosion hazard.,

What does it mean for the erosion hazard to be 'moderate'? To answer this question, the model builder has included an explanation. The result is marked with the note sign, '°', and the annunciator line indicates that if we press 'F9', we can read the note. So, *press* **F9** and the note will appear. Read the explanation of the land use requirement and its severity levels; when you are done, *press* **Return** to remove the note and return to the results screen.

Now that we're done with this land quality, *press* **Return** to confirm it and continue. The window is replaced with a similar one for the land quality 'm' (moisture availability), as you can see from the new title at the top of the window. The first land characteristic to be considered when evaluating for this quality, namely, 'dbr' (depth to bedrock or root-impenetrable layer), is displayed, along with its three classes.

> This land characteristic helps determine the potential rooting volume, which directly affects moisture holding capacity.

Suppose the land to be evaluated has a dense fragipan ('Bx' horizon) at 40cm. As you can see from the class names, this is considered 'shallow'. The highlight should be on choice **?**, 'unknown'; *move* the highlight (with the arrow keys or by typing the class number) until it is on 's' (shallow) and *press* **Return** to confirm it.

ALES now displays the result value, namely 'severe stress', of the land quality 'moisture availability'. After considering only one factor, ALES has concluded that corn will suffer severe moisture stress on these soils. This illustrates how decision tree paths can be of different lengths, depending on which branch is taken. Here the shallow depth is enough, by itself, to allow the model builder to conclude that moisture stress is severe.

Suppose that we made a mistake, and that the fragipan is really a bit deeper, at 75 cm. So we see that we must revise our original choice. Fortunately we can still do this. Notice that the annunciator line says that the left arrow, '←', can be used to go 'back to the last choice'. *Press*← now. The result is erased, and the list of choices for 'dbr' is re-displayed. *Move* the highlight to the second branch, which is the correct value (50-100cm), and *press* **Return** to confirm it.

Now we see that ALES can't make a decision yet on moderately deep soils. It is asking for the subsoil texture

> Subsoil texture is an indicator of water-holding capacity because of the correlation between particle size and pore size. Small pores retain water against drainage, but very small pores retain water against the efforts of plant roots to extract it. Medium textures are a good compromise. Note that this correlation must be calibrated in each general kind of soil (structure and clay mineralogy).

Suppose the area to be evaluated has a silty clay loam subsoil. *Move* the highlight to branch 9, corresponding to 'sicl', and *press* **Return** to confirm it.

Now ALES is asking for another land characteristic value, May-to-September (i.e., growing season) precipitation, 'pptMS', which is measured in four classes.

> This *land* characteristic isn't a *soil* characteristic as are texture and permeability., but rather a *climatic* characteristic. Rainfall affects moisture supply in the obvious way. These classes were the ones delineated on climate maps for the survey area in central New York state.

Suppose that the area receives substantial rainfall, between 450 and 500mm, in the growing season. *Move* the highlight to branch 3, corresponding to '450-500mm', and *press* **Return** to confirm it. ALES now displays the final result: 'moderate' moisture stress, class 2 on a scale of 1 to 3. There is a note for this land quality; read it if you wish.

Press **Return** to complete the interaction for moisture availability. The window is replaced with a similar one for the land quality 'me', conditions for mechanization. The first land characteristic to be considered when evaluating for this quality, namely, 'cfv-A' (volume of coarse fragments in the surface soil), is displayed, along with its four classes.

> If coarse fragments are too numerous or too large, they interfere with mechanical land preparation.

Suppose that there are no significant coarse fragments. *Move* the highlight to branch 1, and *press* **Return** to confirm it.

The next land characteristic to be considered when determining the conditions for mechanization, 'slope', is displayed.

> If slopes are too steep, farm equipment can not safely be operated.

Notice that the highlight is already on branch 3, 'C' (sloping). Recall that this is the value of this land characteristic that we entered in the first land quality box ('e', erosion hazard). Notice that the annunciator line is different: the arrow keys are no longer active. Since you have already entered a value for this land characteristic, it can't be altered now (to do so would be inconsistent). ALES still shows you this step in the decision tree, however. You can only press 'Return' to confirm this value, or back up to the previous land characteristic. *Press* **Return** now. ALES displays the final result: no limitation to mechanization, class 1 on a scale of 1 to 3. There is a note for this land quality; read it if you wish.

Press **Return** to complete the interaction for mechanization limitations. The window is replaced with one for the land quality 'pl', planting conditions.

> The earliness of spring planting greatly affects corn yields in this region, because the yields are generally limited by the available heat units. If plants are well-developed by the time hot weather begins,

The first land characteristic to be considered when evaluating for this quality, namely, 'dc' (soil drainage class), is displayed, along with its six classes.

> Wet areas can not be prepared or planted by machine until they dry out, thus planting is considerably delayed. Note that the soil even in well-drained areas is usually saturated at the end of winter in central New York.

Suppose the land area to be evaluated is well drained; *move* the highlight to branch 2, 'wd', and *press* **Return** to select it.

Now the drainage classes disappear, and next to the 'dc' name is the class we selected, namely, 'wd'. Under this is the name of the next land characteristic to be considered on well-drained soils, namely, 'fpsc', the family particle size class (from USDA Soil Taxonomy), which as you can see is measured in eleven classes.

> There are more classes in Soil Taxonomy, but only these eleven occur in the evaluation area. The family particle size class is used here as an indication of the moisture retention of the entire soil profile.

We have already told ALES that this land has a silt loam topsoil over a silty clay loam subsoil, so it most probably has a fine-loamy particle size class.[23] *Move* to branch 7, 'fil', and *press* **Return** to select it.

The result for this land quality, namely 'early' planting class 1 on a scale of 1 to 4, should be displayed. *Press* **Return** to complete the interaction for planting conditions. This was the last land quality in the model, so no more LQ boxes will be displayed. Instead, the central part of the screen is cleared, and a 'Working...' message flashes while ALES computes overall physical and economic suitabilities.

7.1.4 Viewing the results of the consultation on the screen

When the 'Working...' message disappears, the screen is filled with a summary of the land use requirements and the computed severity levels of their corresponding land qualities.

At the top of the list are the four land qualities in the model, corresponding to the four decision tree boxes we just completed. The land area being evaluated has no planting condition or mechanization limitations, but is moderately limiting for both erosion hazard and moisture availability.

Below the land quality ratings are the final physical and economic *suitabilities*:

1. the *physical suitability subclass*

2. the *economic suitability* based on the *present value* (per unit land area)of the cash flow

3. the *economic suitability* based on the *gross margin* (per unit land area & year),

For both types of economic analysis, ALES computes an *economic suitability class* using the FAO system of three suitable classes (S1: very suitable, S2: suitable, and S3: marginally suitable), and two unsuitable (N1: economically unsuited, and N2: physically unsuited)[24]

As you can see, this land area is *suitable* for the proposed land use. Although it is in physical suitability class 3, due to erosion hazard, the gross margin and net present value are sufficient to place it in the second-highest economic suitability class.

At this point, we could follow the 'Why?' screens with 'F2' to see how the composite suitability values were obtained (this is explained in the next tutorial), or *re-consult* the same model, with different land data, with 'F3'. To keep this tutorial short, however, we'll just end the consultation. So, when you are all done looking at the results, *press* **Return**.

You will now be asked if you wish to save the land data from this consultation. During the consultation, ALES made a temporary list of the data values you entered. These could be saved for starting another consultation with pre-defined data values, or for use in a more comprehensive

[23]ALES isn't able to detect contradictions between different land characteristics, for example, if you stated that the soil was very fine, which would be impossible, given the textures, ALES wouldn't detect the error.

[24]The terminology and methods of economic suitability analysis are explained in Chapter 8.

evaluation. In the present case, the data we entered were hypothetical and not from a real map unit, so there is no point in saving them. Notice that the highlight is already on the default answer, 'No', so just *press* **Return** to confirm.

7.1.5 Ending the consultation

ALES now returns to the list of evaluation models, with the one we've just consulted, namely 'Tutor1', highlighted. If you wanted to re-consult the same model, you would press 'Return' to select the model. However, at this point we'll leave ALES. *Press* **F10** to tell ALES that you're done. The screen should clear and you will be looking at the DOS prompt again.

Congratulations, you've finished the first tutorial! With this practice in interacting with ALES, and with some idea of how ALES models can be used to evaluate a land areas, it's time to move on to a more comprehensive example.

7.2 Tutorial 2 : entering data and running an evaluation

The objective of this tutorial is to teach you the mechanics of computing evaluation results on a set of land units, using an existing ALES model. It is intended for the *model user*, as well as the *model builder*.[25] Once you finish this tutorial, you will be able to define map units, enter their data, calculate physical and economic evaluation results, and print reports, all on *existing* models that were built by a land evaluator.

In this tutorial *you will learn to*:

- start ALES in *evaluation mode*
- enter and edit *land unit descriptions* from the keyboard
- enter and edit *land unit data* from the keyboard
- have ALES *compute evaluation results* for the land units
- *view* evaluation results as a *matrix* on the screen
- use the *'Why?'* screens to understand how results were computed
- *edit* economic parameters and ask ALES to *re-compute* an evaluation
- *print* evaluation results
- *delete map units* from the database

Before working on this tutorial, you should know the basics of interacting with ALES, such as moving the highlight in selection boxes, making selections, using function keys, and getting on-line help. These were all explained in the first tutorial.

This tutorial should take about three hours to complete.

[25] You will build your own evaluation model in the following tutorial.

7.2.1 Starting ALES in 'evaluation' mode

To start ALES in *evaluation mode* from DOS, first change to the directory where ALES is installed. Typically this will be in directory '\dtm' on disk 'c:'. Assuming this to be the case, at the DOS prompt *type* `cd c:\dtm` and *press* the **Return** key.[26]

(If you are already working on disk 'c:', you can omit the 'c:' from this entry. If you are all already connected to this directory, you can omit this command altogether.)

Then start ALES itself by running the batch file **evaluate.bat**: at the DOS prompt *type* `consult` and *press* **Return**.

At this point you should see the *initial* ("welcome") *screen*, which includes the program name, your organization's name, and the serial number. *Press any key*, for example, **Return**, to continue beyond this screen.

An initial menu will now appear:

ALES [Evaluate Only]	
1	**Evaluate**
2	**System Options**
3	**About ALES**

In this tutorial we will only explore item 1, 'Evaluate'.

> You use item 2 'System Options' to change system characteristics, such as screen colors, the human language (English, Spanish or French), and default directories. You use item 3 'About ALES' to obtain license, version and maintenance information.

The highlight should already be on item 1, 'Evaluate' (if not, *move* to item 1 by typing '1'). *Press* **Return** to select this item. You should now be looking at a *list of evaluation models* in the center of the screen and an annunciator line at the bottom of the screen.

For this exercise we will use the model named 'Tutor2', which is a simplification of an evaluation for common field cropping systems in Cayuga County, New York State, USA.

> If 'Tutor2' does not appear in the list of evaluation models, you must first *load* it into your list of ALES models. The saved model 'tutor2.als' is included in the archive 'examples.zip' on the distribution diskette. Extract it to any convenient place on your hard disk with the 'unzip' program (also found on the diskette), start ALES with the DOS command 'ales', and then load 'tutor2.als' into the list of evaluations by using 'F8' when that list is displayed.

Move the highlight to this line in the choice box. *Press* **F9** to view the explanatory note, and when you are done reading the note, *press* **F10** to go back to the menu. Now *press* **Return** to select 'Tutor2'.

---

[26]This key is variously called **Return** or **Enter**. It may be marked with a square arrow,↵

You will now see the *main menu*, number 1, with six choices. Choice 3, 'Evaluate' is highlighted, i.e. is the default choice, since the most common use is to compute and view evaluations. When you start ALES with the 'evaluate' command, you will be able to enter and edit land unit descriptions and data, but you won't be able to alter the models, except for economic parameters such as prices and interest rates.

Since we will start out by entering land unit definitions and data, *move* the highlight to item 2 'Data' and *press* **Return** to select it. Menu 1.3, 'Data', with nine items, will now appear.[27]

7.2.2 Defining a homogeneous land unit

Before we can enter land data, we must first define the map units we want to evaluate. We will work from the *keyboard*, hence we'll use choice 1, to define land units, and then choice 4, to enter data for these land units.

The highlight should already be on choice 1, 'Definitions, from keyboard, Enter or edit' (if not, *move* the highlight by *typing* 1) and press **Return** to select it. A list of the *map units* that have been defined for this evaluation will now appear. These map units are all from the Cayuga County (NY) soil survey. We will add four new map units to this list.

> Notice that the list is longer than the screen; the usual cursor motion keys can be used to see the hidden part of the list.

Notice that the annunciator line shows that 'F3' is to be used for a *new* map unit; so *press* **F3** now. An *entry box* should appear, asking for the code for the map unit.

Land Mapping Unit code? ______________________

This is a kind of interaction we haven't seen before. Occasionally ALES must ask you to type, rather than press function keys. This happens when ALES needs new or revised information. In the present case, we have told ALES (by using the 'F3' key) that we want to define a new map unit; therefore, ALES now needs a code for the new unit, so it is asking us for this code.

Map unit codes are typically found on the map legend for a natural resource inventory such as a soil survey[28], and is a mnemonic for the map unit name. Let's enter the map unit 'Schoharie silt loam, 2-6% slopes'.

Type the code `SeB` in the entry box. *Press* **Return** to confirm this code. Since there isn't already a map unit with this code, ALES will add it to the list.

---

[27]You may have noticed that menu 1.3 appears when you select item 2. This discrepancy is due to the numbering of menus in model-building mode.

[28]But ALES map units can represent individual delineations, either of natural resources or of management units. See Chapter 8 for details.

If you make a mistake while typing the code, erase the incorrect characters from the *end* of your entry with the 'Backspace' key (often marked with a left-pointing arrow, not to be confused with the 'left arrow' key).

First, however, ALES needs some information about the map unit. ALES now displays a three-item *data entry form*, asking for (1) the map unit name, (2) whether it is homogeneous or compound, and (3) the number of hectares it occupies in the survey area.

The first question in the form is the *descriptive name* of the map unit.

Notice that this question is marked with an *asterisk* '*', which means that it is *mandatory*, i.e., you *must* answer it and can not leave it blank.

Type `Schoharie loam, 2-6%` and *press* **Return** to move to the next question, type of map unit (*homogeneous* or *compound*).

This soil map unit is *homogeneous*, i.e, ALES will assume that it has the same values of all land characteristics over all its extent. Land characteristic values are typically measured in ranges, and in a homogeneous map units, these ranges cover all the variability in the map unit.

Type `h` (which is the abbreviation of 'homogeneous') and *press* **Return** to move to the next question, *area*.

Notice that this question is *not* marked with an asterisk '*', which means that it is *not* mandatory. We will enter a value nonetheless. In the study area, map unit 'SeB' occupied 550 acres[29].

Type `550` and *press* **Return** to confirm. Notice that the highlight move back to the top of the form and the first question.

If you make typing mistakes while entering items in the form, you can move between questions with the arrow keys, or just use the 'Return' key to cycle between questions. Within a question, arrow keys can be used to move around, and the 'Backspace' and 'Del' keys can be used to erase mistakes. You can always press 'F1' to see how the keys work.

When you've filled in all three questions of the form, *press* **F10** to confirm all the answers at once. ALES will now record the new map unit in the database. Notice that the list of map units now includes 'SeB', and in fact the highlight is on it.

Once you've created the map unit definition, you can't change the code, or whether the map unit is homogeneous or compound, but you *can* change the descriptive name or area. To practice editing in a data entry from, let's change the descriptive name of map unit 'SeB'. Notice that the annunciator line shows that 'F5' or 'Return' may be used to edit the definition. *Press* **F5**. The

[29]Please accept our apologies for the US customary units that are used throughout this tutorial. Unfortunately, the United States of America has not decided to join the modern world with respect to the use of SI or even metric units. Since the knowledge sources (maps, production costs, yields etc.) are expressed in US customary units, we were forced to use these for our tutorial.

data entry form for the definition of map unit 'SeB' appears, but because this is an *existing* (as opposed to *new*) map unit, the form has only two items, and your original answers appear as *defaults*, i.e., if you don't change them, ALES will keep their existing values.

Move to the last question (the area) by pressing **Return**, and change the answer from '550' to '650'. The easiest way to edit answers in a data entry form is to press **F4** to clear the current answer, and then type the new answer, i.e. `650`. You can also backspace one character at a time through the current answer until it is erased, and then type the new answer. Press **F10** to record your changes. You will be returned to the list of map units. Press **F10** again to return to the 'Data' menu (1.3).

7.2.3 Entering and editing land unit data from the keyboard

Before ALES can evaluate a map unit, it must know the data values for each land characteristic that the model builder has included in the evaluation. In other words, we must *fill in the data base* for this map unit.

To enter map unit data, *move* the highlight to menu item 4, 'Data, from keyboard, enter or edit', and *press* **Return** to select it.

Data values are entered using one or more *data entry templates*. These are *lists* of the *land characteristics* which are included in the database. A selection box with the names of two templates, namely 'cl' (climatological LC's) and 'ss' (Cayuga County soil survey data), should be displayed. Model builders usually group related data items (e.g. that appear on the same paper form or in the same survey report) into templates, so that the model user (you) can enter data separately for each group. In the present case, there are two sources of data: (1) climate maps (Cornell University 1987) and (2) the soil survey report (Hutton 1971). The model builder created two templates, one for each source.

> The model builder can leave notes explaining how data items in the template can be obtained, or other information about the template. Notice that both templates in this list have associated notes, as indicated by the degree signs, '°', to the right of each code. If you wish, you can *press* **F9** to read the note for 'cl' (this is the highlighted code), and then *press* **Return** when you're done reading the note.

Start by entering the climate data. The highlight should already be on the correct template, 'cl' (if not, move it), and *press* **Return** to select it. The list of *homogeneous map units* should now be displayed.

This is a good place to learn another method of moving the highlight in selection boxes, which is especially useful in long lists such as this one. You can use the *typing keys* to make a choice by typing a unique prefix of your answer. A *prefix* of a string of characters is a group of characters that begin the string, and a *unique* prefix of one string in a set of strings is a prefix that distinguishes the string from all the others.

> For example, in the set of strings: 'Example', 'Exercise', 'Explanation', the unique prefixes would be 'Exa', 'Exe', and 'Exp', respectively. Of course there may be strings with no unique prefix. For example, if 'Ex' were added to the set, it wouldn't have one.

Now you can see the most efficient way to move the highlight to the map unit we just defined: type the shortest unique prefix of the code ‘SeB’. In this list, there are no other map unit codes that begin with ‘S’, so all we need to *type* is the single letter `S` (remember to use the ‘Shift’ key, since the case must match). You will see the highlight move to ‘SeB’. *Press* **Return** to select this map unit.

You will now see a data entry form with three items, one for each land characteristic in the template. Each question must be answered with one of the *class codes* for the LC. The highlight is on the first data item, which is repeated at the bottom of the form.

The easiest and most reliable way to enter data is to use ‘F3’ to *pop up*[30] a list of the possible choices (i.e. the class codes), and then use pick the correct choice our of the pop-up list. Press **F3** now, and you will see a new window appear, with the five choices for the first data item, ‘ffs’, which stands for ‘frost-free season’.

Choices for ‘ffs (frost-free season)’

110-130	**[110-130 days]**
130-150	[130-150 days]
150-170	[150-170 days]
<110	[0-110 days]
>170	[170-190 days]

Schoharie soils occur in an area with 150 to 170 frost-free days, so *move* the highlight to ‘150-170’ and *press* **Return** to select it. You will see ‘150-170’ appear in the answer space in the form. The highlight and answer box move to the next question, ‘gdd50’, which stands for ‘Growing degree days, base 50°F’.

Just for practice, answer this question with another method, namely, *typing* the answer in the answer box. The correct answer here is ‘2.4-6k’, but let’s deliberately make a mistake and see what ALES does. *Type* `2.4-6` (i.e. omit the ‘k’ at the end of the class name), and then press **Return** to confirm it. An *error message* should appear in the lower right hand corner of the screen, telling you that the data value just entered is not one of the correct values. *Press* any key to remove the error message, and you will be back at the question, still showing the incorrect answer. The cursor is already at the end of the entry, so you can just type the missing letter, `k`, and then press **Return** to confirm it. Now you can see why the ‘F3’ method is preferable; you only need to type a unique prefix of the answer, not the entire thing, and it’s impossible to make a typing mistake. Also, you don’t have to remember the class codes.

Now the highlight and answer box are on the last question, ‘pptMS’, which stands for ‘May-September precipitation’; *answer* this with ‘<400’ using either method. *Press* **F10** to confirm all the answers in the form, and return to the list of map units.

[30] The term ‘pop-up’ comes from the fact that this list just appears ‘out of nowhere’, at the position of the highlight.

Now the climatological data has been entered for this map unit, but the soil survey data remain to be entered. These data are entered with a different template. To change templates, *press* **F10** to leave the map unit list and return to the template list. *Move* the highlight to template 'ss' ('Cayuga County soil survey') and *press* **Return** to select it. Now you'll see the list of map units again. *Move* the highlight to 'SeB' and press **Return** to select it.

A data entry form with fifteen items, one for each land characteristic in the template, should now be displayed. *Fill it in* according to the following table; when the form is filled in, *press* **F10** to file your answers. Items marked '*[not applicable]*' should be left blank, but all other questions should be answered exactly as they appear in this table. You can use the 'F3' method to answer each question, or, since the answers are all short codes, you might choose to type the answers in directly (remember to press 'Return' to move to the next question).

1.	parent material	:	`lac`	(lacustrine)
2.	limestone	:	`d`	(dominant)
3.	slope	:	`B`	(gently sloping)
4.	previous erosion	:	`m`	(moderate)
5.	depth to bedrock	:	`d`	(deep)
6.	drainage class	:	`mwd`	(moderately well drained)
7.	flooding	:	`n`	(never)
8.	reaction	:	`mial`	(mildly alkaline)
9.	family particle size class	:	`f`	(fine)
10.	texture of surface soil	:	`sil`	(silt loam)
11.	sand modifier, surface soil	:	*[not applicable]*	
12.	type of coarse fragments	:	*[not applicable]*	
13.	volume of coarse fragments	:	`n`	(not enough to be named)
14.	texture of subsoil	:	`sic`	(silty clay loam)
15.	sand modifier, subsoil	:	*[not applicable]*	

You should now be looking at the list of map units. *Press* **F10** *twice* to get back to the 'Data' menu.

7.2.4 Copying land units and editing land unit data

Now we want to define and enter data for the two other members of the Schoharie-Odessa-Lakemont drainage sequence. Odessa and Lakemont soils resemble Schoharie for the most part, and only differ with respect to their drainage class, previous erosion, and surface texture. The climate is the same. So, we can save a lot of work by taking advantage of the ability of ALES to *copy* entities in almost any list of definitions.

First, *move* the highlight to main menu (1a) item '2 Data' and *press* **Return** to select it. Menu 1.3 'Data' should appear. *Select* item '1 Definitions, from keyboard, enter or edit' to see a list of map unit definitions. *Move* the highlight to 'SeB' (i.e., the map unit we just defined and for which we just entered data). Notice that the annunciator (bottom line of the screen) shows that the 'F6' key is to be used to *copy* land unit definitions; so *press* **F6**.

An *answer box* should now be displayed, asking for the new map unit code, just as if you had pressed 'F3' to define a new map unit. However, the box also shows that the new map unit will be copied from 'SeB'. *Type* in the map unit code, `OdA` (this is the Odessa map unit) and press

Return to confirm it. Now ALES will *copy* all information from 'SeB' to 'OdA', including data values.

Once the data has been copied, the same two-item data entry form as for a new map unit will be displayed, asking for (1) the map unit name, and (2) the area it occupies in the survey area. Since this map unit was copied from another, the answers are filled in with the values of the original map unit. The first question, the descriptive name, is of course incorrect. Clear the entire answer by *pressing* **F4**, and then *type* in the correct answer, `Odessa silt loam, 0-2%`. Move to the next question by *pressing* **Return**, and *enter* any value for the area. *Press* **F10** to confirm all the answers.

Now you will be back to the list of map units, with the highlight on the newly-defined map unit, 'Oda'. Using the same techniques as explained in the previous paragraphs, *make a copy* of this map unit as 'Lc', *change* its descriptive name to 'Lakemont silty clay loam', and *return* to the list of map units.

Now you have to change a few data values for the two new map units. *Press* **F10** to return to the 'Data' menu, and *select* item 4, 'Data, from keyboard, enter or edit', and press **Return** to select it. *Move* the highlight to 'ss' in the list of templates, and *press* **Return** to select it. (Remember, the climate is the same for all three members of the drainage sequence, so we don't have to edit anything in template 'cl'.) In the list of map units, *move* the highlight to the first new map unit, 'OdA', and *press* **Return** to select it. The 15-item data entry form should now be displayed, and the data values we entered for 'SeB', from which map unit 'OdA' was copied, should be displayed in the form. Map unit 'OdA' differs from 'SeB' only in 3 respects: slope, previous erosion, and drainage class, so you need only change the answers to questions 3, 4, and 6.

When the data entry form first appears, question 1 is the active question. *Move* to question 3 by *pressing* **Return** or '↓' *twice*. Now you should see the previous value, 'B', in the answer box. *Change* this to 'A', either by popping up a list of the answers with 'F3' and typing 'A', or by backspacing over the incorrect answer and typing 'A'. *Press* **Return** or '↓' to move on to question 4, and *change* the answer to 'n' (no previous erosion). *Move* down two more questions to number 6, and *change* the answer to 'spd' (somewhat poorly drained). The rest of the form is correct, so *press* **F10** to confirm the entire form.

Now you will be back at the list of map units. *Select* the next new map unit, 'Lc', and *edit* its values in the same manner as the previous paragraph, changing the slope to 'A', previous erosion to 'n', and the drainage class to 'pd' (poorly drained).

Lakemont soils have a slightly heavier topsoil than the better-drained members of the drainage sequence, so we must change the answer to question 10 (texture of surface soil) from silt loam to silty clay loam. Let's practice another way of moving among questions in a data entry form. You could press 'Return' or '↓' enough times to arrive at question 10, but a quicker way is to *press* **F5** to *jump* to another question. A small answer box will appear, asking which question to which to jump. *Type* `10` and *press* **Return** to confirm it; the answer box will disappear and the highlight will move to question 10.

Now change the answer to 'sicl' (silty clay loam), either using the 'F3' (pop-up) method or typing the answer `sicl`, and *press* **F10** to confirm the entire form. You have now entered correct data for all three members of the drainage sequence. *Press* **F10** *twice* to return to the 'Data' menu.

7.2.5 Defining a compound land unit

Now you will practice entering another kind map unit: a *compound* map unit, namely, one that is made up of two or more *homogeneous constituents*. You will define an *association* of soil units that has as constituents the homogeneous map units you just defined. ALES will evaluate each constituent separately, and then combine the results.

> Compound map units are commonly used for smaller-scale maps, such as a general soils map of a county or region. At these scales, the individual constituents can't be mapped separately, but since they are present in regular and predictable patterns on the landscape, a compound unit which contains the homogeneous soils can be defined and mapped.
>
> In this case, you will define a drainage association of heavy-textured soils on recent lacustrine sediments: the imperfectly-drained Schoharie silt loam, the somewhat poorly drained Odessa silt loam, and the poorly drained Lakemont silty clay loam. Also included in this association are some topographically-higher areas of fine-textured glacial till that were not covered by lacustrine sediments; these are the Cazenovia silt loams.

To define this compound map unit, you first must *select* option '1 Definitions, from keyboard, enter or edit' from the 'Data' menu, i.e., and *press* **Return** to select it. The list of map units should now appear.

Start out as you did with the homogeneous map unit: when the list of map units is active, *press* **F3** to start a new map unit. Next *type* the map unit code: in this case we will use the code `A13`[31]. *Press* **Return** and you will again see the information form for the map unit. *Fill in* the form as follows: (1) the name is `Schoharie-Odessa association`, (2) it is compound (enter `c`), and (3) its extent is `45000` acres. *Press* **F10** when you are all done with the form.

Now you will see, at the upper right of the screen, a window for the *list of the constituents* of this compound map unit. We told ALES that 'A13' was compound, so now ALES needs to know which homogeneous map units are included in the compound unit. At first, the list is empty, containing only the dummy item 'Add a new item'. Looking at the annunciator, you can see that 'F3' is again used to enter a new item, in this case, to add a homogeneous map unit to the list of constituents.

Press **F3** to add the first constituent. A selection list will now appear on the right of the screen, showing all the *homogeneous* map units.

> Notice that this list is *not* the same as the choice box in which we added the 'A13' map unit. That choice box is still visible in the middle of the screen (although it is partially covered by the more recently-active windows), and you can see that it includes both compound and homogeneous map units. That choice box was used to define the *compound map unit*, whereas the currently-active selection list is used to define the homogeneous *constituents* of the compound unit.

---

[31] 'A13' stands for 'association number 13'; this is the number used in the Cayuga County soil survey.

You must pick the first homogeneous constituent out of the selection list. *Move* the highlight in this list just as in a choice box (remember, you can always press 'F1' to see which keys are active) to the Schoharie soil, with code 'SeB', and *press* **Return** to select it.

You will see 'SeB' added to the list of constituents; in fact it is the only item so far. Now *repeat* the selection process to add three more constituents: 'OdA' (Odessa) and 'Lc' (Lakemont), and 'CeB' (Cazenovia). Notice how items that you select as constituents are removed as choices for further selections. Once you added these four items to the list, *press* **F10** to confirm it.

Now ALES knows what constituents are included in the 'A13' association, but it also needs to know their *proportions* within the compound map unit. These proportions are used to compute weighted averages for crop yields, costs, and economic suitability. You will now see another data entry form, of four items, namely, the proportion of each of the four homogeneous map units. The default values of 25% are shown for each one.

> The default of 25% in this case is because there are *four* units and ALES assumes, in the absence of more specific information, that they occupy *equal proportions* of the association.

In fact, according to the soil survey, the true proportions are: 40% Schoharie, 30% Odessa, 15% Lakemont, and 15% Cazenovia. The first item is 'CeB' (since the constituents are arranged alphabetically by code); erase the default 25% answer by *pressing* **F4**, and then *enter* the correct value of `15`. *Press* **Return** to move to the next item, 'Lc'. *Correct* this to a value of '15', and then *change* the entry for 'OdA' to a value of '30' and for 'SeB' to a value of '50'.

You will note we deliberately created proportions that sum to 110%. Let's see what ALES thinks of this! *Press* **F10** to confirm the form; ALES now notices the mistake and puts up an *error alert* in the lower right hand corner, telling us that the proportions do not sum to 100%. *Press* any key to get past the error message. Now the form is re-displayed, again with the default proportions. Starting from the top again, *enter* the correct proportions, which sum to 100%, namely: 'CeB': 15, 'Lc': 15, 'OdA': 30, 'SeB': 40. Now *press* **F10** to confirm the form; this time everything is fine and ALES files the information for the new map unit, which appears in the list of map units.

Now we're done entering map unit definitions, so *press* **F10** to remove the list of map units and return to the 'Data' menu. *Press* **F10** again to return to the main menu.

> Note that we can't enter data for a compound map unit; this is because the data is associated with each of the homogeneous constituents. The only 'data' for a compound map unit is its list of constituents and their proportions.

7.2.6 Getting ALES to compute evaluation results for the land units

Now we've defined some map units and entered data values, and the model builder has defined some land utilization types. Let's see how well-suited our land is for the various land uses, in other words, it's time to compute an evaluation. From the main menu, *move* the highlight to choice 3, 'Evaluations', and *press* **Return** to select it.

You will see another menu, number 1.4, the 'Evaluations' menu[32]. This menu contains the actions that you will take to compute an evaluation, view the results on the screen, and print the results for the output consumer. The first thing, of course, is to compute. The highlight should already be on option 1, 'Compute an evaluation' (if not, *move* the highlight there), and *press* **Return** to select it.

This is your first encounter with yet another interaction type, the *'set' box*. In this interaction, you can select as many items out of this list as you wish, and the action to be performed will be done on all the selected items, and on no others. Here, as you can see, the list is of the *land utilization types*, that is, the land uses for which we are evaluating the land. The list has three land uses, with codes 'ccc', 'ccc-d', and 'ttpp'. We could choose to compare only the two corn production systems, for example, but for now let's compare all the land uses.

To select *all* items in a 'set' box, *press* **F6**[33]. Notice that all the items in the list are now *marked* with a diamond, '♦'. *Press* **F10** to complete this interaction.

Now another set box should appear, namely, a list of the *map units* that could be evaluated. We must tell ALES which map units to evaluate. Let's select only the association we just defined, namely 'A13'. The highlight should already be on 'A13' (if not, *move* the highlight there) and *press* **F3** to select it[34]. Notice that this item, and only this item, is now marked with the diamond, '♦', and that the highlight has moved down to the next item, so that it could be selected next. At this point,. we could select other map units with 'F3'; however, in this example we only want to evaluate map unit 'A13', so *press* **F10** to complete the selection.

Now ALES begins to compute; this will require from 20 seconds to 3 minutes, depending on your computer[35]. The annunciator line is replaced with a progress report, showing the land utilization type and map unit code that ALES is currently evaluating. A message at the lower right informs us that we can press the 'Esc' key during the computation to interrupt it. This is in case we asked for a large evaluation and decide that it's taking too long.

> The amount of time that ALES requires to compute an evaluation depends directly on the product of the number of land utilization types and map units selected. In the present case, we selected three land utilization types and five map units (the association and, implicitly, its four constituents), so ALES must compute 15 pairs. The computation time also depends on the complexity of the model (especially the number of decision trees and the path length in each tree), and, of course, the processor speed. Disk speed is only a minor factor, due to the fact that MUMPS saves copies of recently-used disk blocks in primary memory. As a rough estimate, an average-sized physical

---

[32]You may have noticed that menu 1.4 appears when you select item 3. Again, this discrepancy is due to the numbering of menus in model-building mode.

[33] Notice that in the annunciator line, 'F6' is marked 'all'.

[34] Notice that in the annunciator line, 'F3' is marked 'select'.

[35]If you get bored waiting for the computation to finish, you might try to compute the physical and economic suitability of five map units for three land utilization types by hand.

and economic model on a 12MHz IBM PC/AT takes about 6 seconds per LMU/LUT combination, on an original IBM PC/XT about 15 seconds.

You will notice the progress report changing as the evaluation proceeds. When it finishes, the annunciator line re-appears. Now we can view the evaluation results.

7.2.7 Viewing evaluation results as a matrix on the screen

To view the evaluation results, *move* the highlight in menu '1.4 Evaluations' to choice 2, 'View evaluation results', and *press* **Return** to select it. Now you will be presented with a selection list of the kinds of reports, and asked which kind of report you want to see. Let's start by looking at the predicted *net present value* ('NPV'), which considers the time value of money and the sequence of costs and benefits across the entire planning horizon.

Move the highlight to 'N'et present value in the selection list (e.g. by pressing **N**), and *press* **Return** to select it. The screen will be cleared, and a 'Working...' message appears while ALES formats the results for display. Then the message is cleared, and you will see the *evaluation matrix*. Notice in the upper-right-hand corner of the matrix the title and units for this display, in the present case, 'Net Present Value [$/acre]'[36].

ALES displays evaluation results on the screen in the form of a *matrix*, i.e. a two-dimensional array with the *rows* being the map units that were evaluated (here, 'A13', 'CeB', 'Lc', 'OdA', and 'SeB'), and the *columns* being the land utilization types for which an evaluation was computed (here, 'ccc', 'ccc-d', and 'ttpp'). Since map unit 'A13' is *compound*, ALES automatically evaluated its (homogeneous) constituents in order to be able to evaluate the compound map unit, hence there is *one* row in the matrix for *each* of its *constituents*.

The *cells* of the matrix show results; each cell is located at the intersection of a single row (map unit) and column (land utilization type), and shows the result for that map unit and land utilization type. The highlight is initially on the upper left hand corner of the matrix; this cell shows the results for the first row, 'A13', and the first column, 'ccc'. In the present case, the cell contents are predicted *net present values*. So, looking at the highlighted cell, we see that if we implemented land utilization type (column) 'ccc' on map unit (row) 'A13', according to the ALES model and the data we entered, we would expect a net present value of $27.64 per acre.

The matrix provides us with useful comparisons among and within land units and land utilization types. First, we can look in a given *column* (representing a LUT) to see which land area is most remunerative for that LUT. For example, looking in the first column (representing 'ccc'), we see that map unit 'CeB' is far superior to the other map units as a land area on which to implement continuous corn with no land improvements. This answers the question: 'What is the best land for this land use?'.

Next, we can look in a given *row* (representing a map unit) to see which LUT's are most remunerative on that map unit. For example, looking in the first row (representing the compound

---

[36] Each evaluation can have a different currency symbol and unit of areal measure, so the choice of dollars and acres for this example is because the available data was expressed in these units.

map unit 'A13'), we see that LUT 'ttpp' has a far-higher present value than the others. This answers the question: 'What is the best use for this land unit?'.

We can compare entire columns to see the overall relation among land utilization types. For example, comparing the two leftmost columns, for 'ccc' (continuous corn) and 'ccc-d' (continuous corn with land drainage), we can see that land drainage increases the NPV (i.e. is cost-effective according to this economic metric) for map units 'CeB', 'Lc' and 'OdA' (remember, these are the moderately-well, somewhat poorly, and poorly drained soils), but not for the well- drained 'SeB' soils. If a farm was on a typical association of these (i.e. association 'A13'), draining the entire farm would not be cost-effective. Another comparison we can make is between corn (in the leftmost two columns) and pasture (LUT 'ttpp', in the rightmost column). This model indicates that pasture is always much more profitable than corn. Of course, a farm must usually be made up of a balance of enterprises, so this result does not mean that no corn should be grown.

We can compare entire rows to see the overall relative 'value' of map units. For example, it is clear that 'CeB' is superior to the other map units for all three LUTs, although with the exception of 'Lc' all the map units are fairly similar for the pasture land use.

> The above comparisons of Net Present Values are only valid because we have specified *equal-length planning horizons* for all three LUT's in the model.

7.2.8 Using the 'Why?' screens to understand how results were computed

The matrix now on the screen gives results, but it is natural to wonder, what is the basis for these? How were they computed? What data were used? What assumptions were made by the model builder? Which factors were considered, and which were omitted? The model user, i.e. the 'consumer' of the land evaluation, has a right and a duty to be suspicious!

One of the outstanding features of ALES is that it allows you to *review* the logic and data that it used to reach its conclusions. Although computation proceeds in a forward direction, from basic data to final suitability, the logic is most conveniently reviewed using a *backward chain*, beginning with the results and proceeding backwards through intermediate calculations to the land data. In this 'top-down' approach, the most important points are seen first, and only as much complexity as necessary to understand the results need be reviewed.

To examine the reasoning for a particular map unit – land utilization type pair, we must first move the highlight so that the matrix cell containing that pair's result is highlighted. We can move among columns (i.e. left and right) by using the left and right arrow keys. We can move among rows (i.e. up and down) by using the up and down arrow keys, and the 'PgUp', 'PgDn', 'Home', and 'End' keys (see the 'F1' help for a complete description). An easy way to move among rows is the prefix method, i.e. by typing a unique prefix of the map unit name.

Let's see why the Cazenovia map unit ('CeB') has a predicted NPV of $239.45 per acre for continuous corn with land improvement ('ccc-d'). *Move* the highlight to the correct column, i.e. 'ccc-d' by pressing the right arrow '→'once.. *Move* to the 'CeB' row by pressing the down arrow '↓' once. Now that the highlight is on the correct cell, *press* **F2** to initiate the 'Why?' screens.

The rightmost two-thirds of the screen will now be replaced by a 'Why?' box (notice its header). In the box are the full names of the land unit and land utilization type, followed by the *discount rate*, i.e. rate of interest at which cash flows are discounted. Below this is a table showing the cash flows and their present value for each year in the *planning horizon*. The screen is only big enough to show the first 12 years. Notice that the last line in the box says '...<more>...'; this indicates that there are more lines below. To see these lines, *press* **PgDn**, and the screen will scroll until the final lines are all visible, showing all 16 years in the planning horizon. The key number here is the very last one on the lower right: $239.45 per acre is the NPV. This was obtained by summing the column above it, each entry in this column representing the NPV for that year. For example, year 1 has an NPV of $-393.16 (i.e. a net negative cash flow), year 2 has an NPV of $79.14, etc., this column summing to $239.45.

All of the 'Why?' screens are explained by a 'help' screen, which you can see by *pressing* **F1**. Do this now. You will see an explanation of 'present value', namely, that a cash amount (in or out) is discounted as follows:

where the *DiscountRate* is the rate at which money may be borrowed, expressed as a percent (remember, this was shown on the third line of this 'why?' screen, and in the present case is 10%), and *YearsFromPresent* is the number of years from the present when the *CashAmount* will be spent or received. Note that cash *in* is considered to be received at the *end* of a year; this is why the year-1 cash in of $280.00 is only worth $254.54 when discounted; by contrast, cash *out* is considered to be spent at the *beginning* of the year, so that the year-1 cash out of $647.70 is worth the same amount when discounted. Thus the effective years of discounting for cash out always lag those for cash in.

When you are done reading the 'help' screen, *press* **F10** or **ESC** to return to the 'Why?' screen. Notice that the annunciator line at the bottom of the screen shows how to find out *more details* about the calculation: 'F2' is marked 'Why Cash In?' and 'F3' is marked 'Why Cash Out?'. Let's first determine how the figure of $280.00 cash in was determined, so *press* **F2** again. A selection box appears, asking which year's cash flow should be explained. In this case, all the years have the same cash in, so just *press* **Return** to accept the default choice of year 1.

Now a new 'Why?' box appears (the previous one is hidden 'underneath'), explaining the cash flow in: it is based on one harvest this year of one crop, namely grain corn, with a return of $280.00 per crop. If this land utilization type included more crops or harvests in year 1, this box would be more important, showing all the outputs harvested this year. Press **F2** again to see 'Why an output return?'. Since grain corn is the only output, we are immediately shown another 'Why?' box, this one showing why each crop of corn is predicted to return $280.00 per acre.

The computation of one-crop return is straightforward: it is the *optimum yield* (here, 140 bushels of grain per acre), multiplied by the *proportional yield* predicted for this particular map unit (here, 0.8, or 80% of optimum), resulting in a *predicted yield* (here, 112 bushels of grain per acre); this in turn is multiplied by the *selling price* of the crop per unit yield (here, $2.50 per bushel), resulting in the predicted return. The optimum yield was set by the model builder, and can not be edited except in 'model building' mode (see Tutorial 3). The selling price was also set by the model builder, but you can edit it to reflect changing conditions. We'll do that in the next section.

Right now, we will continue the backward chain some more, because one part of the computation is not yet explained, namely, how the proportional yield was obtained.

This is really the heart of an ALES economic evaluation: determining how a particular land unit will perform, relative to the optimum in the evaluation context. To examine this, first *move* to the bottom of the current 'Why?' box, by pressing **PgDn**; you will see the display scroll until the last line is visible. You will notice three groups of lines in the display relating to proportional yield: (1) the *proportional yield decision tree*, (2) *proportional yield factors*, and (3) *limiting yield factors*. We'll discuss these in order.

In the present case, we can see that the *proportional yield decision tree* wasn't defined by the model builder; because in this case there were no interactions between land qualities that couldn't be accounted for by one of the other two methods. These two are appropriate for land qualities whose effect on yields is *independent* of the values of other land qualities. There are two kinds of independent effects that can be accounted for in ALES: *multiplicative* (also known as '*proportional*'), and *limiting*.

Proportional yield factors are used by the model builder to represent *multiplicative* effects of a land quality. The proportional yield predicted by the proportional yield decision tree (if any) is multiplied by any proportional yield factors for the land qualities that were *not* encountered on the decision tree path.

In the present example, there were two proportional yield factors defined by the model builder, namely, for erosion hazard and for moisture availability. Since the severity level of the erosion hazard was determined to be 1 (none) during the computation, and this land quality did not appear in the decision tree path (since there wasn't a decision tree), the yield factor is 1, i.e. no reduction from the optimum. Similarly, the severity level of the moisture availability was determined to be 2 (moderate stress), and the corresponding yield factor is 0.8, i.e. a reduction to 80% of the optimum.

Limiting yield factors are used by the model builder to represent *limiting* effects of a land quality. The proportional yield predicted by the proportional yield decision tree *and* the multiplicative yield factors (if any) is *absolutely limited* by any limiting yield factors for land qualities that were *not* encountered on the decision tree path. In other words, the yield can not be any greater than the most limiting of the limiting yield factors for the set of land qualities in the evaluation, not including those qualities that the model builder already included in the proportional yield decision tree.

In the present example, there was one limiting yield factor defined by the model builder, namely, for temperature regime, and this land quality did not appear in the decision tree path (since there wasn't a decision tree). However, this map unit has no temperature regime limitation, since it was determined to have a full season (class 1). So, this factor did not affect the yield.

Now we know how the proportional yield was determined, but this leads to another question: how were the severity levels of the three land qualities which figured in the proportional yield determined? Notice that 'F2' in the annunciator line is marked 'Why a severity level?'. *Press* **F2** now; a selection box appears, asking which land use requirement, of the three, you want to examine. Let's find out why there was moderate moisture stress. *Move* the highlight to the second line (marked 'm') by typing the unique prefix, namely m, and *press* **Return** to select it.

A new 'Why?' box appears. You can see from the four header lines that it is explaining, for land unit 'CeB', the severity level 2 of land quality 'm' of land utilization type 'ccc-d'. The explanation consists of displaying the path taken in the *severity level decision tree*. This is our first encounter with a *decision tree*, so let's take some time to understand it.

In ALES, the model builder expresses inferences by constructing *decision trees*. At each level in the decision tree a different factor (in the present case, land characteristics) which is necessary to reach a final decision is considered. Each factor divides the tree in *branches*, depending on how many classes there are for the factor. At the lowest level of the tree (the *leaves*) is a *decision*, i.e. the result (in the present case, a severity level of land quality). A severity level decision tree is constructed by the model builder so that ALES may determine the severity level of a *land quality* from one or more values of *land characteristics* .

In the present example, since map unit 'CeB' was deep to bedrock (*first level* of the tree), had no previous erosion (*second level* of the tree), has a silty clay loam subsoil (*third level* of the tree), and has less than 400mm summer precipitation (*fourth level* of the tree), it is predicted to have a moderate moisture stress in the context of this LUT.

Notice that two lines begin with the degree sign, '°', indicating that there are notes that apply to these lines Let's look at the one for the decision tree; this is where the model builder has explained the structure of the tree. *Press* **F9**, and then move to choice 2 in the selection box, and press **Return** to select it. The explanatory note for the severity level decision tree will now appear. When you are done reading the note, *press* **F10** to return to the selection box. If you wish, you can look at the other note. *Press* **F10** to return to the 'Why?' box.

We are almost at the end of the 'Why?' chain, but there is one more level, namely, the actual values of the land characteristics that determined the path through the severity level decision tree. Notice that 'F2' in the annunciator line is marked 'Why a land characteristic value?'. *Press* **F2** now; a selection box appears, asking which land characteristic, of the four, you want to examine. Let's find out why the depth to bedrock was deep. The highlight should already be on the first line (marked 'dbr'); *press* **Return** to select it.

This final explanation is quite simple: the value of the land characteristic was entered from the keyboard (you entered it earlier in this exercise). Suppose it is wrong? A nice feature of ALES is that during the 'Why?' screens, you can correct information at the time you notice it's wrong. Notice that 'F4' in the annunciator line is marked 'Edit land characteristic value'. *Press* **F4** now; a one-item data entry form appears, allowing you to edit the value of the current land characteristic, 'dbr', for the current map unit, 'CeB'. The current value, 'd', is the default answer, which is correct. If you wanted to change the value, you could either type in the new value, or press 'F3' to pop up the list of choices. The value is in fact correct, so *press* **Esc** to cancel the data entry form and return to the 'Why?' screen.

Now we will go back through all the superimposed 'Why?' boxes, removing them one at a time, until we reach a point at which we want to follow another chain. *Press* **F10** now to return to the explanation of the severity level for 'm'. At this point we could go backwards again to another land characteristic. *Press* **F10** again to return to the explanation of the proportional yield. At this point we could go backwards again to another land quality. *Press* **F10** again to return to the explanation of the year's output returns, and *press* **F10** yet again to return to the explanation of the cash flows and net present value.

We have seen the explanation of the cash in; now it is time to follow the 'Why?' chain for the other side of the gross margin equation, namely, the *costs*. Notice that 'F3' in the annunciator line is marked 'Why cash out?'. *Press* **F3** now; a selection list appears, asking for which year we want to see the costs. The highlight is on year 1; press **Return** to select it. A new 'Why?' screen appears, showing the first (highest-level) step in the calculation of costs for this map unit for year 1 in the plan.

This screen shows the map unit, LUT, and year in the plan. The cost per hectare-year of implementing a particular LUT on a given LMU is the sum of three factors:

1. The *S1 cost*, i.e. those costs that are incurred in *every map unit*, regardless of their specific land quality values[37];

2. The *additional cost* of implementing the land utilization type on *this particular map unit*, with its own specific set of land quality values (severity levels)[38]; and

3. The *production-related costs*, i.e. those that depend on the level of production of the various outputs that are produced this year in this LUT[39]. Note that this may differ among map units, because production levels usually differ.

Let's look at these three types of cost in detail. Notice in the annunciator line that 'F2' is marked 'Why S1 cost?', 'F3' is marked 'Why additional cost?', and 'F4' is marked 'Why Production cost?'. *Press* **F2** now. A new 'Why?' screen will appear, explaining the year-1 S1 costs as a sum of two lists of inputs: the annual (recurrent) inputs and the one-time inputs for year 1. Here, the annual inputs are various agricultural chemicals such as herbicides and pesticides, as well as labor, fuel, and other machine charges. These are the costs of planting, growing, and harvesting corn on optimum land. You may notice there is no fertilizer listed here; in this LUT these vary according to a land quality, and so are included in the 'additional' costs, which we'll see a bit later. Scroll the 'Why?' screen down by *pressing* **PgDn**, and you will see the one-time inputs that are specific to year 1, in this LUT, those due to the installation of a drainage system. Again, only those costs of installing the drainage system that do not vary among land units are included here; the others are included in the 'additional' costs.

Costs are the sum of individual inputs, each of which requires a certain number of *units*, and each of which has a certain *purchase price*. For example, looking at this display we see that 23 feet of 8" drainage tile (for main drains) are required per acre, and that each foot of tile costs $2.35 installed, resulting in this one-time expenditure of $54.05 per acre.

---

[37]These were entered with Menu 1.2 'Land Utilization Type' options '4 : Inputs - annual' and '5 : Inputs - by year'

[38]These were entered with Menu 1.2 'Land Utilization Type' option 'Land Use Requirements', sub-options '2: additional Annual inputs' and '3': additional By-year inputs'.

[39]These were entered with Menu 1.2 'Land Utilization Type' option '6 :Outputs', sub-option '5 : Production-dependent inputs'

Press **F10** to go back to the explanation of overall costs. Now that we've seen the S1 costs for the land utilization type, let's see the additional costs for this land utilization type & land unit combination. Notice that 'F3' in the annunciator line is marked 'Why additional cost?'; so *press* **F3** now.

You should now be looking at a list of all nine land use requirements defined for this land utilization type, each with a list of additional annual and one-time (for year 1) inputs. The sum of all these is the additional cost for the land unit as a whole

For example, the first LUR listed is 'c' (temperature regime); there are no additional inputs associated with this LUR, because in this context we can't correct for a limiting climate.

The next LUR listed is 'dr' (suitability for subsurface drainage), and here we see that this particular map unit is rated as severity level 4 ('90-foot spacing') out of 6 possible severity levels of the LUR, and that there is an additional year-1 *one-time* input of 484 feet of 4" drainage tile (for lateral drains), each foot costing $0.85 installed, for a total additional cost of $411.40. Note that the costs for lateral drains are included as an additional charge, because the spacing of the drains, and hence the number of linear feet needed, depends on the land quality 'dr' (which is mainly determined from the subsoil permeability); by contrast, the costs for the main drains were included as an S1 cost, since their spacing does not vary on different kinds of land.

Press **PgDn** to see the bottom of the 'Why?' screen. The last LUR listed is 'nr' (nutrient requirement), and here we see that map unit 'CeB' is rated in severity level 2 ('Honeoye group') out of 6 possible severity levels, and that this LUR has an associated list of *annual* additional inputs, namely, the major fertilizer nutrients (N, P, and K). These are additional inputs because the amount of fertilizer needed depends on the nutrient requirements of the map unit. In the present case, i.e. on map unit 'CeB', these are 100lb N, 40lb P_2O_5, and 40lb K_2O.

Let's see why this map unit was rated in severity level for the 'nutrient requirement' LUR. Notice that in the annunciator line, 'F2' is marked 'Why severity level?', so *press* **F2** now. A selection list will appear, asking which LUR to investigate. Type 'nr' to move the highlight in the selection list to the correct LUR, and press **Return** to select it. A new 'Why?' screen appears, showing the map unit, LUT, LUR, computed severity level (here, 2), and the *decision tree path*. Here the decision is very simple: the nutrient requirement is directly related to the New York State 'soil management subgroup', which is a grouping of New York soils for purposes of fertilizer recommendations. Notice that in the annunciator line, 'F2' is marked 'Why Land Characteristic value?', so *press* **F2** now. Since there is only one LC in the decision tree path, a new 'Why?' screen immediately appears, showing a *land characteristic - to - land characteristic decision tree*, which the model builder constructed to allow inference from the basic map unit properties (here type of parent material, amount of limestone in the parent material, and Soil Taxonomy family particle size class) to the higher-level LC (here the management group).

Press **F10** twice to return to the breakdown of additional costs by land use requirement.

7.2.9 Editing economic parameters and re-computing an evaluation

Although you can't change most aspects of the models, such as decision trees, if you enter ALES with the 'evaluate' command, you are able to edit some economic parameters, namely: input purchase prices, output selling prices, and interest rates. In this section you'll learn how to do this, and then re-compute an evaluation.

Notice that in annunciator line in the present explanation of the additional costs, 'F5' is marked 'Edit prices'. This is the editable economic parameter that affects the calculation that is shown on the current 'Why?' screen.

Let's suppose that the cost of nitrogen fertilizer has increased from $0.20 to $0.25 per pound since the time the model was built. We want to edit the input price, so *press* **F5**. A list of all the inputs in the evaluation is presented. Move the highlight to the one we want to edit, namely 'N', by typing the unique prefix N, and press **Return** to select it. A one-item data entry form now appears, for the price of nitrogen, with the current value, $0.20 per pound, as the default. The cursor is at the end of the answer, so just *type* 5 to specify the new value, 0.25. *Press* **F10** to confirm the form, and *press* **F10** again to return to the last 'Why?' screen.

Notice that the new value has *not* been reflected in either the 'Why?' screen or the evaluation matrix. ***ALES does not recalculate until told to do so***. This allows you to make several changes in the model before recalculating. We want to recalculate now, so return to the evaluation matrix by *pressing* **Esc**. This shortcut will remove all the stacked 'Why?' screens at once.

Now that we're back on the evaluation matrix, notice that 'F3' is marked 'Compute' in the annunciator line. *Press* **F3** now to initiate the recomputation. ALES will ask you if you really want to recompute the evaluation; this is to guard against accidentally pressing 'F3' instead of 'F2'. *Type* 'Y'(es) and *press* **Return** to start the computation.

The annunciator line is replaced by a progress report, exactly as when we originally computed the evaluation results. After the recalculation is complete, the annunciator line reappears, and the matrix is updated with changed values. Any values which have changed during the computation are *highlighted*. In the current case, the two LUTs with corn as their output are changed, but the pasture LUT isn't, since N is only specified as an input for corn. Notice that the NPVs were reduced, e.g. for 'CeB' by $51.63, to $187.82, by the price increase.

Let's change this economic parameter back to its original value, to leave the tutorial in the same condition as we found it for the next user. We can do that right from the evaluation matrix, without going into the 'Why?' screens again. Notice that 'F5' on the annunciator line is marked 'Edit Econ', i.e. edit the economic parameters. *Press* **F5** now, and you will be presented with a menu of three items, headed 'Economic parameters'. We want to edit an input price; this is choice 1. The highlight should already be on this choice by default, so *press* **Return** to select it. Now you will see a selection list of the inputs. *Move* the highlight to the one we changed, namely 'N', by typing the unique prefix N, and *press* **Return** to select it. A one-item data entry form now appears, for the price of nitrogen, with the current value, $0.25 per pound, as the default. Erase the final '5' from the current answer by *pressing* **Backspace**, *press* **F10** to confirm the form, and *press* **F10** again to return to the menu, and *press* **F10** once again to return to economic parameters menu.

Let's get a feel for just how important the discount rate can be in determining the cost-effectiveness of a LUT over time. *Select* item 3 in the menu, and *change* the interest rate from 10 to 2% (this might represent a government-backed low interest loan to encourage land improvement). *Return* to the evaluation matrix, and *press* **F3** to ask ALES to recompute the matrix; *confirm* that this is what you want to do.

When ALES finishes re-computing, you will see that the NPV's of LUT 'ccc-d' have greatly improved, and in fact are now all cost-effective (some were negative at 10% interest). *Press* **F2** to see the cash flows; you will see that the cash flows for the later years in the plan are worth much more at present than they were with the higher discount rate. Since the cash flows for all years past the first are positive, the lower discount rate results in increasing present value. Note that the annunciator line shows that 'F4' may be used to edit the discount rate; so *press* **F4** now, and *change* the discount rate back to the original 10%. *Return* to the evaluation matrix.

Press **F10** to return from the evaluation matrix to the selection box for report type. Notice that there are several report types that you can select. If you wish, you can follow the 'Why?' screens for the gross margin, for physical suitability, for crop yields, or for single land quality values.

When you are all done exploring the 'Why?' screens, *press* **F10** to return to the evaluations menu.

> In this tour of the 'Why?' screens, data editing, and editing of economic parameters, we haven't explored every display or feature of ALES' 'Why?' system. If you wish, you can experiment now. The annunciator line always shows the active function keys, and 'F1' always brings up a 'Help' screen to explain the current 'Why?'. Remember that pressing **F10** enough times will always return you to the evaluation matrix.

7.2.10 Printing evaluation results

The results of an ALES evaluation can be printed in a format that is most useful to land use planners and others who need to know about the suitability of land for land uses. The same sort of conclusions that we drew while viewing the matrix can be presented in this form to planners.

In the 'Evaluations' menu, 1.4, move the highlight to choice 3, 'Print evaluation results', and press **Return** to select it[40]. Now you will be asked to specify where you want the results printed: either on the '**V**'ideo screen, on a '**P**'rinter, to a disk '**F**'ile, or through a '**C**'ommunications port. This may sound a bit redundant, after all, don't you want them to print on the printer? Not necessarily!

√ The option to view the results on the *video* screen allows you to *preview* the results before printing.

√ The option to write to a *disk file* allows you to edit the results with a word processor prior to printing them. For example, you might want to add explanatory notes, or change the formatting.

√ The option to write through a *communications port* (e.g. 'COM1:') allows you to send the output to a printer that was not configured in the MUMPS system, or to another computer system connected through the port.

40The same reports can be requested with option '1 Evaluation Results' of menu '1.5 Reports'.

And of course,

√ The option to *print* allows you to get a *printed report*, on paper.

If you have a printer connected:

> *Move* the highlight to 'P' (it should already be there by default) and press **Return** to select.
>
> > If you have more than one printer configured in your MUMPS system, you will now have to choose one of them. Otherwise ALES will assume you want to use the parallel printer.
>
> You will now be presented with a data entry form, with the *form parameters*, i.e. options for the placement of the report on the page, the width (in characters) and length (in lines) of the printed form, etc. These will be probably be correct as they stand. If not, edit the form as appropriate (remember, 'F2' at any question will give you more information about the required answers). When the form is correct, *press* **F10** to confirm all the answers.

If you don't have a printer, or the printer is not connected:

> *Move* the highlight to 'V' to see the report on the video screen.

Now a selection list will appear, asking which kind of suitability information is to be printed. Since we've been reviewing present values in the 'Why?' screens, let's print or view these. Move the highlight to 'N'et Present Value, either using the down arrow or typing the unique prefix, and press **Return** to select it.

Another selection list appears, asking whether to organize the report by *rows* (land units) or *columns* (land utilization types). In the first case (*by row*), each *map unit* will start on its own page, and each page will show the rating of a single map unit for each land utilization type. This shows what the *best uses are for each map unit*. In the second case (*by column*), each *land utilization type* will start on its own page, and each page will show the rating of all the map units for that LUT. This shows the *best map units for each land use*. The default is by land unit. Let's print the by-row report now, so press **Return** to accept the default choice. The 'Working...' message will appear on the screen, and the report should appear on the printer or screen, depending on your choice of 'P'rinter or 'V'ideo, above.

The report has *one* page for *each* map unit. On each page, the land utilization types are *ranked* according to predicted NPV. So for example on the first page, for map unit 'A13', we see that the LUT's are ranked (1) 'ttpp', (2) 'ccc', and (3) 'ccc-d'. This tells us what the best uses are for this land, using NPV as the economic metric.

Notice at the bottom of the first page is the notation '[VI = $141.46]'. 'VI' is an abbreviation for *versatility index*, i.e. a composite measure of 'goodness' of the map unit as a whole. In the case of economic metrics, it is just the average of the metric over the land utilization types.

After the report has finished printing, the 'Working...' message will disappear from the screen, and you will again be asked for the kind of information to be printed. This is so you can select another report to go to the same output device, without having to set it up again. Notice that the highlight is on the type of report we just printed, namely 'N'et Present Value. Let's print this report again, but this time by columns. Press **Return** to select the NPV report, and in the 'Sort by?' selection list, move the highlight to 'C', for 'by-column', and press **Return** to select it.. Again the 'Working...' message will appear on the screen, and the report should appear on the printer.

This report has one page for each *land utilization type*. For each of these, it shows the map units, ranked in order of their predicted NPV for the LUT. For example, on the first page (for LUT 'ccc') we see that 'CeB' is the most remunerative type of land, with 'SeB', 'A13', 'Lc' and 'OdA' following. The latter two have an identical NPV when used for this LUT. Notice that the sum of the areas with each predicted NPV is printed below the group; this gives the planner a measure of how much land of each type is available.

After the report has finished printing, the 'Working...' message will disappear from the screen, and you will again be asked for the kind of information to be printed. *Press* **F10** to go back to Menu 1.4, and *press* **F10** again to go back to the main menu.

7.2.11 Deleting land units

Now that you're all done with this tutorial, let's return the ALES database for this model to its status before we started. In order to do this we'll have to *delete* the map units we added. This will give you practice deleting entities, which works similarly in all kinds of lists within ALES. To do this, we first need to display the list of map units. From the main menu *select* item '3 Data'; from menu '1.3 Data', *select* option '1 Definitions, from keyboard, enter or edit'. A list of the map units defined in this model should appear.

Notice in the annunciator line that 'F4' is the 'delete' key in this context. Let's try to delete one of the homogeneous map units we defined in this exercise. *Move* the highlight to 'Lc' (Lakemont silty clay loam), and *press* **F4** to delete it. Now an error message should appear: ALES won't let you delete this map unit, since it is a *constituent* of *compound map unit* 'A13'. If we deleted 'Lc', there would be no way to know what the correct proportions of homogeneous constituents were in the compound unit 'A13'. We must either remove this constituent from the compound map unit, or delete the entire compound unit.

We were going to delete 'A13' anyway, since it was one of the units we added earlier in the tutorial. So, *press* **Return** to remove the error message, and then *move* the highlight to 'A13' and press **F4** to delete it. A 'yes-no' box will appear, asking if you really want to delete it. *Press* Y to move the highlight to 'Yes', and press **Return** to confirm. 'A13' will disappear from the list.

Now *move* the highlight back to 'Lc' and *press* **F4** to try again to delete it; now you will be asked to confirm, and when you do so, it will be deleted from the list. Deleting the map unit definition also deletes any data values stored for the map unit, and its evaluation results.

In the same way, *delete* 'OdA' and 'SeB'. Now the list is back to the way it was before we started. We're done with the tutorial, so *press* **F10** enough times to exit ALES back to DOS.

Congratulations, you've finished Tutorial 2! Review the objectives at the beginning of the tutorial to make sure that you have mastered them.

7.3 Tutorial 3 - building a model

The objective of this tutorial is to teach you the mechanics and some of the logic of building ALES models. It is intended for the model builder. This is only an introduction to model building; many of the more sophisticated capabilities of ALES have been saved for Tutorial 4. In this tutorial you will learn to:

- start ALES in *model building mode*
- *create* a new evaluation model
- *back up* a model to a DOS file
- define *reference lists* of land use requirements, inputs, and outputs
- define *land characteristics*
- define a *Land Utilization Type*
- build *decision trees*
- define *data entry templates*
- *refine* and *adjust* models from the evaluation matrix and 'Why?' screens

Before working on this tutorial, you should have completed Tutorials 1 and 2. We will assume you are able to navigate within ALES, and are familiar with all interaction techniques. In this section we won't spell out every keystroke, as in the previous two sections. This tutorial should take about 6 hours.

The emphasis in this tutorial is on the *mechanical* aspects of model building, i.e. *how* to build models using the ALES program. To make the presentation go smoothly, many important *intellectual* aspects of model building, i.e. *why* models are built with a certain structure, have been deferred to Chapter 8. You may wish to read Chapter 8 in parallel with this exercise.

This tutorial is too long to be completed at one sitting. Whenever you want to stop, *press* 'F10' enough times to leave the program. Then when you restart ALES, select your copy of the tutorial and follow the menu structure, making appropriate choices at each choice box, to return to the point where you left off.

> Sometimes we all make *mistakes*; ALES makes it fairly painless to correct them. For any answer, whether in data entry forms, single-answer entry forms, or choice boxes, you have to press 'F10' to confirm your entry. If you catch your mistake *before* you confirm the entry, you can *correct* it by various methods. For example, in a data entry form, you can use the 'backspace' key or 'F4' to erase some or all of a typed-in answer, and then you can just type the correct answer. Another way is to start the interaction all over: instead of pressing 'F10', you can *cancel* any interaction in ALES by pressing the 'Esc' key. Finally, if you have already confirmed an incorrect entry, you can almost always *edit* data in a similar manner to entering new data. In almost every situation where the 'F3' key means 'add a new item', the 'F5' key means 'edit the existing item'.

7.3.1 Starting ALES in 'model building' mode

Using the same procedure given in the first subsection of the previous two tutorials, *start* ALES with the DOS command `ales`. *Move* past the program title screen and the main menu to the list of evaluation models. Notice that the annunciator (bottom line) now includes several new options, including 'F7' to save (make a backup copy) of an evaluation to a disk file, 'F8' to restore a saved evaluation, 'F3' to create a new evaluation model, 'F4' to delete an evaluation model, 'F5' to edit basic information for the evaluation, and 'F6' to copy a model. All this power means you are in *model building* mode: 'with ALES you are *always* in control'.

7.3.2 Creating a new evaluation model

You should now be looking at the list of *existing* evaluation models. For this tutorial we will create a completely *new* model. The model will be a simple physical and economic evaluation for fish production in Central America. In the interest of simplicity, we will only define one land utilization type: 'Tilapia in diversion ponds', to be implemented by small farmers with no mechanization, but with access to adequate credit and inputs[41].

> In a real evaluation we of course would compare alternatives.

In order to keep the tutorial to a workable size, we will simplify it by only including a few of the most important land use requirements and inputs, and by keeping the decision procedures simple. The model will be complex enough to be realistic, but not comprehensive. We are sure you can think of many refinements, especially if you have experience with fish farming. When you are done with the tutorial, you're welcome to experiment with it and enhance it as you see fit.

> Fish farming may seem like an unlikely choice for an ALES model, but actually it is a good example of land evaluation: we have land, and fish farms are land use alternatives to arable and perennial crops.

Press F3 to define a *new* model. You will be asked to enter a *code* by which this evaluation will be identified in the list of evaluations. A good choice would be your initials (to keep it separate from the tutorials that other users may be working on), and some indication that it's a tutorial. So if your name is 'Anthony Rice', you might use the code 'ar-Tut3'. Or, you could use a phrase that fits your personality. Note that codes are short, no more than 8 characters.

Once you have confirmed the code, you will be asked for the basic information that applies to the *entire* evaluation:

(1) a *descriptive name*,

(2) a *currency symbol*, and

(3) the *areal unit* in which land is measured.

---

[41] Thanks to Ms. Christine DeRoller for the original version of this tutorial.

The descriptive name should be a short phrase that explains the evaluation code, e.g. 'Tutorial 3 - R. Johnson's exercise'. The currency symbol defaults to '$' (dollars) and the areal unit to 'ha' (hectares); for this exercise leave the currency symbol as is, but since the size of a typical group of fish ponds is considerably less than a hectare, and cost figures are only available at the smaller scale, change the unit area to 'are' (1ha = 100 ares, so 1 are = 0.01ha = 10m x 10m).

> Or, you could convert existing cost figures to hectares, if you wanted the evaluation results reported on a per-hectare basis.

After you confirm the data entry form, ALES returns to the list of evaluations. Notice that your newly-defined evaluation is in the list, and in fact the highlight is on it. Now let's practice entering an *annotation*. It is very important for the model builder to explain the reasoning behind the design choices to the model user, and annotations are useful for this purpose. You know from the previous tutorial that 'F9' is used to read notes. Pressing 'F9' while holding down the 'Alt' key (usually located to the left of the space bar) is how you enter the *note editor* to enter or edit notes. The way to remember this is that the 'usual' thing to do is *view* the note, whereas the '*alt*ernate' things to do is *edit* it.

√ 'Alt-F9' is used to create and edit annotations.

Press **Alt-F9** now. A large window appears in the leftmost two-thirds of the screen. The title of the window gives the name of the entity for which you are entering a note, in this case the evaluation code. Notice that the cursor is blinking in the upper left hand corner of the window. You are being invited to *type* your note. This is a simple word processor; you can press 'F1' to see a list of the editing keys. In general you just type your note, and the text automatically wraps to the next line as you reach the right margin. You can use the 'Return' key to force a new line.

Type some explanatory text of your choosing now. For example, `ALES Tutorial 3: Fish farming in Central America.` Notice that the text wraps to the next line. If you make a mistake while typing, you can use the 'backspace' key to remove the error, and then continue typing. Experiment with the text editor: use the arrow keys to move back in the text and insert words. If you would like to delete the entire note and start over, press 'F8'. When you are all done editing the note, *press* **F10** to confirm your edits.[42]

Now you should be looking at the list of evaluations again, and the highlight should still be on the new evaluation code. Notice that there is now a degree sign, '', next to the code, indicating that there is now a note for your evaluation. *Press* **F9** to read the note; notice that the note window is only as large as is necessary to show the note. *Press* **F10** when done looking at your note; you should now be looking at the list of evaluation models, with the highlight on your newly-created model.

7.3.3 Backing up models to DOS files

Don't risk wasting all your hard work in model building due to disasters, natural or otherwise, such as (1) disk crashes, (2) someone reformatting the disk, (3) someone erasing ALES or one of

[42]This note editor sometimes gets confused during text editing; if the text becomes garbled, leave the editor with 'F10' and enter it again.

its datasets, (4) someone erasing or altering your model within ALES, (5) ... (other horrors too many to enumerate). *Make a backup copy of your model* to diskette on a regular basis.

Even though your test model is very small right now, let's practice making a backup. Note that the 'F7' key is marked 'Save'. The highlight should already be on the model you've just created.

√ 'F7' is used to save evaluation models to DOS files.

Press **F7** to initiate the save. An entry box appears, asking for the DOS *file specification* to which the backup copy should be written. The default file specification consists of (1) an optional *path*[43] (1) the evaluation code as the file *name*, and (2) a file *extension* of '.ALS', which stands for '*AL*ES *S*ave'.

If the path is missing, the backup is written to the same directory as the ALES program. This provides a backup copy in case the model is deleted or altered within ALES, however, it does not protect you significantly against problems with the hard disk, such as a head crash. A better strategy is to back up your models to a diskette

To specify that the backup should be made to diskette, you have to add a *drive letter* and *semicolon* in front of the file name and extension. You can move the cursor (insertion point) back through the entry box with the '←' key; however, a quick way to move the cursor to the first character of the answer is to press and hold the 'Ctrl' (control) key while pressing the back arrow. Do this now. You will see the cursor blinking under the first character of the file name. Get a formatted diskette to hold your backup, and insert it in a diskette drive. Then type the drive letter, followed by a colon, in front of the file name. For example, if you are backing up to drive 'A', *type* `A:`. Press **Return** to confirm your answer.

If you already have a file of the same name on the diskette (e.g. if you have done a previous backup to the same diskette), you will be told this, and asked what to do ('O'verwrite, 'A'ppend, 'B'ackup, or 'Cancel'). You will usually select 'B' to back up the file already on disk (with the extension '.BAK').

> It is good practice to keep at least two previous backups, so that if you are inadvertently saving a model that has been corrupted or altered for the worse, you have a few days to discover this, and go back to a previous backup. To do this, prepare two backup diskettes and alternate them.

You will see a 'Working...' message, and some gibberish (courtesy of MUMPS) at the upper left of the screen while the model is being saved. When the save is complete, you will be returned to the list of evaluations. *Press* **Return** to select your model and move to the main menu.

As you work in ALES, you should back up your models whenever you have made a significant number of changes. For example, if you have to interrupt this tutorial, it would be a good idea to back up your model before taking a break.

[43] You can set the default path with Menu 2 'System Options' Item 8 'Set default path for saved evaluations'

7.3.4 Defining reference lists of land use requirements, inputs, and outputs

You can see that the main menu is organized in the order in which we might conduct a land evaluation:

(1) define *reference lists* to be used in all LUT's;

(2) define *Land Utilization Types*;

(3) enter the *data*;

(4) *evaluate*;

(5) *print* reports of evaluation results or model contents.

> You can also access the 'consultation' mode (covered in the first tutorial) from this menu, with option 6, and create maps in the IDRISI Geographic Information System, with option 7.

The first order of business is to define the reference lists, so select option 1 now. Now you will see Menu 1.1, 'Reference lists', with four items. A *reference list* is a list of entities that can be used when defining the various land utilization types of the evaluation. You can think of reference lists as data tables that are independent of a particular LUT. As you can see, they are of four types:

(1) *Land Use Requirement codes*

(2) *Outputs* (products)

(3) *Inputs*

(4) *Land characteristics*, i.e. the land database structure

The first reference list is for *land use requirements* (LUR) In this list we define LUT *codes*, a short *descriptive phrase*, a default *number of severity levels*, and default *severity level names*. The rest of the definition of a land use requirement depends on the context of the land utilization type. For example, what constitutes a 'severe' erosion hazard on inter-tilled row crops is quite different than in tea plantations. Even the number of severity levels may be different. The reference list provides a basis to be used in *any* of the LUT's.

Select menu item 1. You will now see a choice box, headed 'Land Use Requirement code'. There are no entities yet, only a dummy item '?' to hold the place in the box. Let's define the LUR's now. Notice in the annunciator line that 'F3' is to be used to add a new item; that is what we want to do, so *press* **F3** now. Now you're being asked to enter a code for the new entity.

The land utilization type that we intend to define requires the construction of the fish-holding structures: i.e. earth ponds fed by diversion ditches from streams or springs. Hence an important land use requirement is the 'potential for construction', i.e., how easy it is to build the ponds. A good code for this might be 'ppc'. In the entry box *type* ppc and confirm it.

A two-item data entry form now appears, asking for (1) a descriptive name and (2) the default number of severity levels. The descriptive name appears next to the LUR code in many contexts. The default number of severity levels is the number of different degrees of limitation that can be distinguished for the LUR; usually this is 4. So, *type* `potential for construction`, *press* **Return** to move to the second question, *type* 4 and *press* **F10** to confirm the form.

A four-item data entry form now appears, asking for the descriptive names for the four default *severity levels*. These names appear along with the severity level. In the present case, we'll enter these when we define the LUT, below, so for now leave them blank. *Press* **F10** to confirm the (empty) form.

The reference list of LUR's should now reappear. Notice that the '?' item in the choice box is replaced by the new code 'ppc'.

Using the same technique as the previous paragraphs, *add* the following codes to the list

(1) LUR code 'wq', with descriptive name 'water quality', 4 levels, no level names;

(2) LUR code 'wa', with descriptive name 'water availability', 4 levels, no level names.

When you are done with the list of three items, return to menu 1.1 'Reference Lists'.

Now *select* option 2 to enter information on *outputs*. In this list, we enter all the outputs (products) that may be produced by *any* LUT. Again you will see an empty list. Only one thing will be produced by our only land utilization type: fresh Tilapia fish. So, *press* **F3** now to define a new code, and when asked for the code, *enter* 'ti' for Tilapia. A three-item data entry form should appear. For each output we must specify (1) a *descriptive name*, (2) the *units* in which the output is measured, and (3) a *selling price* per unit. For this output, *enter* `Tilapia - whole fish` for the descriptive name, `quintals`[44] for the unit of measure, and 400 for the price (i.e. 100kg of fresh fish sells for $400). Confirm this form by *pressing* **F10**; notice that the list now has one item. *Return* to Menu 1.1 'Reference Lists'.

Now *select* option 3 to enter information on *inputs*. These are the things that are needed to implement one or more of the land utilization types. In the present model, we will have inputs associated with construction and with annual maintenance of the ponds, as well as items that are directly related to the crop, such as baby fish, nutrients and other chemicals, and water.

Press **F3** to define a new code, and when asked for the code, *enter* `labor`. You will see a three-item entry form for the new input. You are being asked to specify (1) a *descriptive name*, (2) the *units* in which the input is measured, and (3) a *purchase price* per unit. For this input, enter `manual labor` for the descriptive name, `day` for the unit of measure, and 2 for the price. In other words, we are saying that manual labor is valued at $2/day.[45]

---

[44] 1 quintal = 100 kg

[45] In this model we are not distinguishing the farmer's own labor from hired labor. If we were, we could define two inputs and given them different prices.

Confirm this form; notice that the list now has one item, 'labor'. In the same manner, define the following five inputs:

code	name	units	price
lime	*fine limestone*	*quintal*	*2*
fish	*baby fish*	*each*	*.05*
manure	*farm manure for fertilizer*	*quintal*	*1*
tile	*plastic drainage tile*	*meter*	*9*

You should now have a list of five inputs. *Return* to Menu 1.1 'Reference lists'.

7.3.5 Defining land characteristics

Land characteristics (LC) are the measured or estimated properties of the land which form the data items in the ALES data base. In models, they are used in the *decision trees* that determine the *severity level* of each land quality, and, ultimately, the final suitability ratings of the various LUT's. Characteristics are included in the reference list either because they are useful for determining suitability levels (the *top-down* approach) or because they are available in a resource inventory (the *bottom-up* approach). In this evaluation we'll take the first approach: pick the characteristics we want, and then assume that they can be obtained by field survey, interviews with the land owner, or simple lab test.[46]

> Until we define the severity level decision trees, it won't be clear why we defined this particular set of land characteristics. For now, we will concentrate on the mechanics of entering the chosen LC's into ALES.

From Menu 1.1 'Reference Lists', *select* option 4, 'Land Characteristics'. A choice box, similar to those that we saw for inputs and outputs, will appear, with a single dummy entry, 'add a new item'. As before, *press* **F3** to initiate the definition of the first item in the list.

7.3.5.1 Defining a discrete land characteristic with units

The first land characteristic we'll enter will be *water temperature*[47]. *Type* the code `wt` and confirm it. Now you will see a three-item data entry form, titled 'wt (NEW land characteristic)'. The first question is straightforward: a descriptive name for the characteristic. *Type* `water temperature` and *move* to the next question.

Now comes a crucial question: the *number of classes* for the land characteristic. Land characteristics can have zero classes, i.e. be measured on a *continuous* scale. These are not

[46]This comes from the original application of the model in Guatemala, where Peace Corps volunteers are expected to determine the values of the land characteristics from interviews and simple field tests.

[47]Water temperature is critical in fish farming; each species of fish has a well-defined and usually narrow range of temperatures at which they can grow.

directly usable, however, in decision trees[48]; these require *discrete* LC's, i.e. those measured in two or more *classes*. Although the current LC, water temperature, certainly has an underlying continuous scale (i.e., if we put a thermometer in the water at a given time, we get a single value, e.g. 18.4°C), the scale can be *divided* into a usable number of classes. The number of classes, and the class limits (dividing points) depends on (1) the accuracy desired in the decision procedure, and (2) the accuracy obtainable in the field. In the current situation, we need to distinguish water temperature ranges that are important for Tilapia,. In the present, *four* classes will be sufficient: cold (<16°C), temperate (16-23°C), warm (23-35°C), and hot (>35°C). So to the second question, answer `4`.

WARNING: Once you enter the number of classes for a land characteristic, and confirm it, you will not be able to change this number.

> Unlike almost everything else in ALES, the number of classes of land characteristics, land use requirements, and physical suitability can *not* be edited after entry. The only way to change these is to *delete* the entity (using 'F4' in the appropriate choice box), and *reenter* it. However this is quite difficult once the LC is used in decision trees (see below), since the LC must be removed from each decision tree in which it is used before it can be deleted. So, it is important to *plan* carefully prior to constructing the model.

To the last question (units of measurement), *type* `°C`. Note that to enter the degree sign '°' on most PC keyboards, hold down the 'Alt' key while pressing the numbers '2', then '4', and then '8' on the numeric keypad, then release the 'Alt' key.

> If for some reason you can't enter the degree sign from your keyboard, answer this question with `deg C`.

Now *confirm* all three answers by pressing **F10**. Menu '1.1.4a Specification options for discrete LC (with units)', appears, containing six items.

The first item, name and units, would allow us to edit the descriptive names and the units of measurement we just entered. The fifth and sixth items will not be covered in this introductory tutorial. We will, however, work through items 2, 3, and 4. We have already told ALES that this LC is to have four classes; now, however, we have to give it some information about each class.

Select menu option 2, 'class abbreviations'. A data entry form appears, with four items, one for each of the four classes. You are being asked to supply short (8 character or less) *abbreviations* for each class. These abbreviations are codes which are used in data entry. They should be short and *mnemonic* (i.e., the abbreviation should remind you of what the class signifies). They need not contain any numeric limits, as these limits will always be displayed to the user during data entry after they are entered at menu option 4 (below).

Let's use abbreviations for the descriptive words 'cold', 'temperate', 'warm' and 'hot': *enter* the four abbreviations as follows: 'c', 't', 'w', and 'h', and then *confirm* the form.

---

[48]In tutorial 4 we'll see why and how to use continuous LC's

Now *select* menu option 3, 'class names'. You are being asked to supply longer descriptive names for printed reports, data entry screens etc. *Enter* the descriptive words ''cold', 'lukewarm', 'warm' and 'hot', and then confirm the form.

Now *select* menu option 4, 'class limits'. We are being asked to supply the *lowest* possible value, in the units of measurement (which are °C), for any value of the LC, and *upper* limits for the four classes. The default lower limit is always zero, but it can be reset to any value.

In the present case, we know that flowing water will always be above freezing, so the default value of 0°C is OK. Moving on to the next question, recall that for this LC, that cold water is by definition 16°C or below. So enter `16` as the upper limit for class 1, and *move* to the next question. Similarly, classes 2 and 3 have upper limits of 23°C and 35°C, respectively; *enter* these. Class 4, 'hot' water, was defined as any water above 35°C; certainly it can not be more than 100°C and still be flowing, so *enter* an upper limit of `100` for the final class. *Confirm* the form.

Now *press* **F10** again to leave Menu 1.1.4a. You have now completed the entry of information for this LC. Notice that the choice box now has one land characteristic listed, namely, 'wt', which we just defined.

Except for the number of classes, any other information for land characteristics can be edited at any time. Notice in the annunciator line that 'F5', as well as 'Return', are marked 'Edit'. Let's practice editing an existing LC definition now. The highlight is already on the only item, 'wt', so *press* **Return** or **F5**. Menu 1.1.4a reappears, allowing you to edit anything except the number of classes. *Select* option 3, 'class names'. You will see the four-item data entry form with the four class names. Let's change the name for class 2 from 'lukewarm' to 'temperate'. To do this, *move* to the second question in the form with the down arrow or the 'Return' key. Erase the old answer by *pressing* **F4**, and then *type* the new answer, `temperate`. *Press* **F10** to confirm the form, and then *press* **F10** again to leave menu 1.1.4a and go back to the list of land characteristics.

7.3.5.2 Defining a discrete land characteristic without units

The next land characteristic we'll enter will be *source of water*[49]. *Press* **F3** to initiate the entry of a new LC, and then *enter* as its code `ws` and confirm it. Now you will see the three-item data entry form. For the descriptive name *enter* `source of water` and *move* to the next question, i.e. the *number of classes* for the land characteristic.

Not all land characteristics have underlying continuous scales of measurement; the LC we are defining now is a good example. Water to be used in the fish ponds can come from three sources: rain, river, or springs. Here we have three classes, but no underlying scale or intrinsic ordering: there is no way to order them, other than arbitrarily[50]. So, *answer* the second question with `3`, and *leave the last question blank* (this tells ALES that there are no units of measurement associated with this LC) and then confirm the form.

---

[49]Obviously, a reliable source of water is necessary to keep the fishponds filled with clean water.

[50]We can't say that rain is 'greater than' a spring, for example.

√ A Land Characteristic *without any units of measurement* is called 'discrete, without units'. It is a *nominal* land characteristic.

Again we see a menu, this time titled '1.1.4b Specification options for discrete LC (no units)', with only four items. Since there are no units of measurement, it doesn't make sense to enter any class limits; also there are no commensurate LC's (this will be explained in the fourth tutorial). You need only enter information for menu items 2 and 3. *Select* item 2, and for the three abbreviations *enter* 'rain', 'river', and 'spring'; *confirm* the form.

√ The order of the classes of a nominal LC is arbitrary.

There really isn't anything to add in the way of descriptive names, since the codes are sufficiently descriptive. *Press* **F10** to return to the choice box; notice that 'ws' is now in the list.

7.3.5.3 Defining the rest of the land characteristics

Now *define* the remaining eight land characteristics to be used in the evaluation, according to the following table:

LC code	**descriptive name**	**# of classes**	**units**	**LC lower limit**
class #	*class code*	*class name*		*class limit*
rs	**length of rainy season**	**3**	**months**	**0**
1	short	–		6
2	medium	–		10
3	long	–		12
cfc	**coarse fragment content**		**5**	**% volume 0**
1	1	few		3
2	2	some		14
3	3	many		35
4	4	excessive		55
5	5	extreme		100
cfs	**coarse fragment size**	**4**	**cm (diameter)**	**0**
1	fg	fine gravel		2.5
2	cg	coarse gravel		7.5
3	r	rocks		25
4	b	boulders		1000
sd	**soil depth**	**3**	**cm**	**0**
1	s	shallow		75
2	m	moderately deep		200
3	d	deep		1000
sl	**slope**	**4**	**%**	**0**
1	A	nearly level		3
2	B	sloping		10
3	C	steep		30
4	D	very steep		200

text	**soil texture**	**4**	**–**	**–**
1	c	coarse		–
2	m	medium		–
3	f	fine		–
4	p	peat		–
pH	**pH of the water**	**4**	**pH**	**0**
1	vacid	very acid		4
2	acid	acid		6
3	n	neutral to slightly alkaline		9
4	valk	very alkaline14		
pe	**permeability of subsoil3**		**–**	**–**
1	s	slow		–
2	m	moderate		–
3	r	rapid		–

You should now have a list of nine land characteristics: 'cfc', 'cfs', 'pH', 'pe', 'rs', 'sd', 'sl', 'text', 'ws', and 'wt'. *Press* **F10** to return to Menu 1.1 'Reference Lists', and **F10** again to return to Menu '1 Main Options'.

> If you are familiar with standardized descriptions of map units or soil profiles, you may notice that several of the characteristics do not have the same number of classes, class abbreviations, or class limits, as you may expect. For example, soil texture is measured here in only four classes here, instead of the standard 12 USDA classes. This illustrates one of the key advantages of ALES over a system with a predefined list of characteristics: everything about the LC's can be defined according to local needs. In the present case, we feel that knowing whether the texture of the fine earth is coarse, medium, fine, or peat is sufficient to determine the severity levels of the land qualities that we would like to use for our land utilization type. This simplifies both the model and the data collection process.

7.3.6 Defining a land utilization type

Now that you've completed the preliminaries, it's time to get to the heart of the model: the *land utilization types* (LUT). These are the land use *options* to be considered in the model.

From Menu 1 'Main options', *select* item 2 'Land Utilization Types'. A choice box should appear; since there aren't any LUT's defined yet, it is empty.

Press **F3** to define a new LUT. ALES will then ask for the LUT code; *type* `tdp` (which is a mnemonic for '**T**ilapia in **d**iversion **p**onds') and *confirm* the code by *pressing* **Return**. A four-item data entry form should appear, titled 'tdp (NEW land utilization type)', in which you are to enter basic information about the LUT

7.3.6.1 Information about the LUT as a whole

The *first* item, 'descriptive name', is the text that is displayed along with the LUT code. Here, *enter* 'Tilapia in diversion ponds'.

The *second* item is the length of the 'planning horizon', which is the time over which the model builder can enter by-year inputs and outputs.

> ALES also uses the length of rotation to normalize all cost and return figures to a per-year basis in gross-margin analysis, and to compute the Net Present Value and Internal Rate of Return of a LUT. The planning horizon takes into account both the recurrence time of the rotation or other management cycle, and the length of time over which any capital improvements are effective. This is discussed fully in §8.3.2.

In fish culture, management is based on a one-year cycle; however we have a significant capital investment in pond construction. For the purposes of this tutorial, let's assume that the ponds have an expected life of five years; so *answer* the 'planning horizon' question with '5'.

The *third* item is the 'discount rate'. The meaning of this was covered in Tutorial 2 and is discussed fully in §8.3.1. For this exercise, assume the discount rate is 5% (meaning that the government has subsidized the rate at which farmers can borrow money to build the ponds).

The *final* question, '# of physical classes', is asking for the number of levels of *physical* suitability that the model will determine. Since this will be an economic evaluation, and the economic values will reveal the different levels of suitability, we can save work by only having *two physical suitability classes*, one corresponding to FAO order 's' and the other to 'n'. Finer degrees of suitability will be revealed by the economic analysis. (See §8.2.1 for a discussion of how to decide on the number of physical suitability classes.) So, *answer* this question with '2', and then *confirm* the entire form.

The number of physical suitability can *not* be edited after entry. The only way to change these is to *delete* the entire land utilization type (using 'F4' in the LUT choice box), and *re-enter* it. This would be a lot of work on a large model. So, it pays to design the model before entering it.

WARNING: Once you enter the number of physical suitability classes for a land utilization type, and confirm it, you will *not* be able to change this number.

Menu 1.2, 'Specification options for LUT: 'tdp'', with nine items, will now appear. We will spend a lot of time in this menu.

The *first* item is the descriptive name of the LUT, which we just entered. The *second* item is the length of the planning horizon, which we also just entered. The *third* item includes the discount rate, which we just entered. These can be ignored for now.

7.3.6.2 Annual and by-year inputs

The *fourth* and *fifth* menu items allow the model builder to enter or edit the *inputs* to the land utilization type, *irrespective of the land it is implemented on*. (Land-dependent costs will be addressed under menu item 7, 'Land Use Requirements'.) First let's define the *annual* inputs, menu item 4.

Annual inputs are those which must be supplied to the LUT each year during the plan, i.e. they are *recurrent* inputs.

Select item 4 now. You will see a choice box, titled 'Annual Inputs for LUT "tdp"', with only the dummy item in the list. *Press* **F3** to enter the first item. Now you can see a difference between this choice box and the ones we've seen before. We are not being asked to *type in* a new code; instead, we are being asked to *select* a code out of a *predefined lists*. In this case, the codes we can choose are those that were defined in the *reference list of inputs* (§7.3.4, above). There are five possible annual inputs; we are going to use two of them at this time: labor and manure.

Move the highlight to 'labor' and press **Return** to select it. A one-item data entry form appears, asking how many units (in this case, days per are) per year will be required to implement the fish farm. Labor is required for fertilizing, maintaining the pond banks, harvesting each crop of fish, and replenishing the water after each harvest. None of these is very labor-intensive; a reasonable estimate of the total time required is six days per 10x10m area (are) per year. So, *answer* this question with 6 and *press* **F10** to confirm.

Now we are back on the choice box, with one item, 'labor', in the list. Using the same technique that you used to add the first input, *add* the second input, 'manure' to the list, specifying 2 quintals per are per year. Manure is required to promote algal growth in the ponds (the fish eat the algae).

Press **F10** to *confirm* the list of annual inputs and return to menu 1.2. Now we will enter the *by-year* inputs, i.e. those inputs that are supplied at some *specific time* during the LUT, and are not repeated each year. Like the annual inputs that we just defined, the by-year inputs here are *not* dependent on the specific land, only on the land utilization type.

The by-year inputs are entered with menu item 5. *Select* item 5 now. You will see a choice box, titled 'Years with by-year inputs for LUT "tdp"'. This is a list of the years in which any by-year inputs have been defined. At this point there aren't any yet, so the list consists of only the dummy item.

Press **F3** to add a year to this list; a selection list of the years which can be added (i.e. 1, 2, 3, 4 and 5) appears. The highlight is already on year 1; *press* **Return** to select it. Now you will see a choice box titled 'By-year Inputs for LUT "tdp" [Year 1]', with only the dummy item in the list. This is the list of by-year inputs for the *first* year in the plan. In this LUT, the ponds must be built at the beginning of the LUT, i.e., in the first year; in later years they do not need to be re-built, only maintained. Capital improvements such as this are typical of by-year inputs in the first year of a multi-year LUT.

Add inputs to this list exactly as you did for the annual inputs. The three items, and the amount of by-year input are:

(1) 'fish': 200 units per are (to stock the pond);

(2) 'labor': 50 days per are (to build the ponds); and

(3) 'tile': 4 meters per are (to make a drainage system for the pond).

After you enter these three inputs, *return* to the list of years with by-year inputs. At this point we could add by-year inputs for later years in the plan (e.g. for pond maintenance), but in this model we don't have any, so *return* to menu 1.2.

7.3.6.3 Land use requirements

The *seventh* item of menu 1.2 is used to specify the *land use requirements* for this land utilization type, i.e., the conditions of the land that are necessary for successful implementation of the LUT. This is the most important part of the model.

Select item 7 'Land Use Requirements' now. You will see a choice box, titled 'Land Use Requirements for LUT: 'tdp'', with only the dummy item in the list. *Press* **F3** to enter the first item.

The choice box that appears on the screen is the same kind that we saw when entering inputs in the previous section, i.e. we are not being asked to type in a new code; instead, we are being asked to select a new code out of a predefined lists. In this case, the codes we can choose are those that were defined in the reference list of land use requirements (§ 7.3.4). We will use all three of the available LUR codes in this LUT.

Let's start with the potential for construction. The highlight is already on the code for this LUR, namely, 'ppc', so just *press* **Return** to select it.

7.3.6.3.1 Severity levels

You are now being asked for the number of *severity levels* of the corresponding land quality for the land use requirement 'potential for construction', i.e. the number of classes of this land quality that will be distinguished in the context of the current LUT. The default value of 4 is displayed; recall that we specified this default in the reference list.

For this land quality the default of four severity levels is fine. Level 1 will correspond to *no* problems with construction, level 2 to additional costs for *drainage*, level 3 to additional costs for *earth moving*, and level 4 to extreme conditions that make construction *impractical.*

> In this example level 3 is not necessarily 'worse' than level 2; they just represent different types of additional costs, which are not necessarily increasing as the severity level increases.

The number of severity levels can *not* be edited after entry. The only way to change these is to *delete* the land use requirement from the LUT definition (using 'F4' in the LUR choice box), and *reenter* it.

WARNING: Once you enter the number of severity levels for a land quality of a land utilization type, and confirm it, you will not be able to change this number.

So, *answer* this question with 4 and *confirm* the form. Now you will see menu 1.2.7, 'Specification options for LUT, LUR "tdp, ppc"'.

The *first* menu item is used to give descriptive names to the four severity levels; these will appear along with the severity level in reports, explanations, consultation mode etc., and are a great help to the model user. *Select* menu option 1, and *fill in* the four-item form that appears with: 'no limitation', 'needs drainage', 'earthmoving', and 'impossible'; finally, *confirm* the form.

7.3.6.3.2 Additional inputs

Recall that severity levels 2 and 3 are associated with *additional costs*, i.e., the LUT costs more to implement on land with severity levels 2 and 3 of LUR 'ppc'. We will enter them now. Both costs are incurred in the first year when the ponds are built; there are no annual recurring costs associated with the LUR 'ppc', because the ponds are not built every year, but only in the first year.

The by-year costs are entered with menu choice 3; *select* it now. A choice box appears, titled 'Years with by-year inputs for LUT, LUR "tdp", "ppc"'. This is a list of the years in which any by-year inputs associated with LUR 'ppc' have been defined. There aren't any yet, so the list consists of only the dummy item.

Press **F3** to add a year to this list; a selection list of the years which can be added (i.e. 1, 2, 3, 4 and 5) appears. The highlight is already on year 1; *press* **Return** to select it. Now you will see a choice box titled 'By-year Inputs for LUT, LUR "tdp", "ppc" [Year 1]', with only the dummy item in the list.

This is the list of by-year additional inputs associated with pond construction for the first year in the plan. A choice box appears, asking for these additional by-year inputs. *Press* **F3** to add an input to this list; you will see the five inputs we defined in the reference list of inputs. *Move* the highlight to the last item, 'tile', and *select* it. A form appears, asking for the number of units, in units of measurement per unit area (in this case, linear meters per are), for each severity level of the 'potential for construction' land use requirement.

Recall that severity level *2* is to be associated with additional costs for *drainage construction*, so this is the only one of the four levels with an additional by-year cost for drainage tile. *Move* to line 2 in the form, and *enter* `10`; in other words, we are saying that land that has land quality rating '2' for 'potential for construction' will require 10 linear meters of plastic tile per are (10x10 meter area) in order to make it fit for Tilapia production in diversion ponds. *Confirm* the form, and return to the list of additional by-year inputs, which now has one item, 'tile'.

Press **F3** again to add another input; you will see a list of four inputs to choose from. The list doesn't include 'tile', since we have already entered added by-year costs for this input. *Select* 'labor' from the list. A form appears, asking for the number of units, in units of measurement per unit area (in this case, days per are), for each severity level of the 'potential for construction' land use requirement.

Recall that severity level 3 is to be associated with additional costs for *earth moving*, so this is the only one of the four levels with an additional by-year cost for labor. *Move* to line 3 in the form, and *enter* `5`; in other words, we are saying that land that has land quality rating '3' for 'potential for construction' will require 5 additional days of labor per are, for earthmoving, in order to construct the diversion ponds. *Confirm* the form, and return to the list of additional by-year inputs, which now has two items, 'labor' and 'tile'.

That's all the additional by-year inputs associated with this land use requirement, for any year in the plan (in this case only year 1), so *press* **F10** *twice* to go back to menu 1.2.7.

7.3.6.3.3 Severity level decision trees

Now we are really going to have some fun! It's time to build a *decision tree* that will allow ALES to infer the severity level of the land quality corresponding to LUR 'potential for construction' from some set of land characteristics. During the computation of an evaluation, ALES attempts to place each map unit into one of the four severity levels. In other words, it determines whether the map unit (1) has no limitations for pond construction, (2) will require extra tile drainage, (3) will require extra earth moving, or (4) has so many problems that construction is impractical. To do this, ALES follows *decision trees*, built by the model builder, which relate *land characteristics* to *land qualities*.

> The tree used in this example has purposely been kept small by only considering a few factors.

Select item 4 of menu 1.2.7 now. The main screen we have been looking at for quite a while now, with its overlapping windows, disappears, and is temporarily replaced with the *decision tree editor*. This is a completely different interaction. As you build models, you will spend much of your time building and altering trees with this editor. At first it looks a bit mysterious, but you will soon become used to it. It follows the *structure* of the decision trees, so that editing the trees is as efficient as possible.

> As always, on-screen help is available. If you wish, *press* **F1** now and take a moment to skim the help screen. We will cover most of this information in the exercise, but now you know how to find it on-line. *Return* to the editor now.

As you can see on the screen, at present there is *no* decision tree. The annunciator line shows what keys are active; what we want to do now is *insert* the first level. Notice that 'F4', as well as '↵', is marked 'ins >', which stands for 'insert subtree'.

√ '>' is a shorthand for 'subtree'.
√ 'F4' is used to enter a subtree.

Press **F4** now, and a *selection list* of the discrete land characteristics will appear. You are being asked to pick out of this list the land characteristic that you want to consider *first* when determining the severity level of the land quality 'potential for construction'.

Looking at the list, we see some irrelevant land characteristics, such as the pH of water, but several that are relevant. The idea is to build the simplest model possible, so we pick the land characteristic that will most quickly *narrow* the choices and reach a decision as to whether a given area of land is suitable for construction of diversion ponds.

Assuming that the most important factor is soil depth (we will see why in a moment), *move* the highlight to 'sd' and *select* it.

The selection list will disappear, and the editor now shows the first level of the decision tree. At the top is the first *decision entity*, i.e. 'sd', and below it are four *branches*, one for each land characteristic class (shallow, moderately deep, and deep), and one marked '?'.

> For the remainder of this Tutorial 3 we will ignore this final 'unknown' branch; its use will be covered in Tutorial 4.

To the right of each branch is a single '?', signifying that *no decision* has yet been made with regard to that branch.

√ **'?' is a shorthand for 'no decision' or 'unknown'**

Now we must imagine land with each of the three possible depths. Can we determine its potential for construction just based on soil depth? In the case of *shallow* soils, we can (at least in this model): it is too difficult to excavate into underlying rock, or to transport large amounts of earth from another site to build raised ponds, so we can say that shallow soils are *impossible* to construct ponds on.[51]

So, we want to associate severity level 4 (impossible) with shallow soils, i.e. those with land characteristic class 's' for soil depth. Notice that the highlight is already on branch 1, the one for class 's'. Also notice in the annunciator line that 'F3' is marked 'ins *', which is short for 'insert a final decision'.

√ **'*' is a shorthand for 'final decision'.** √ **'F3' is used to enter a final decision.**

In other words, by pressing a 'F3' we will be able to tell ALES the final decision for this branch. *Press* **F3** now, and you will see a selection box with the four possible choices, i.e. the severity levels. *Move* the highlight to '4' (impossible) and select it.

Notice now that the '?' to the right of the first branch has been replaced with the *decision*, i.e. severity level 4. The asterisk, '*' to the left of the decision is the editor's notation that this is a final decision. Notice also that the highlight has moved down to the second branch.

> You can move among branches in two ways: with the up and down arrow keys, and by typing the branch *number*. If you wish, you can *practice* both methods of moving through the branches, and *return* to branch 2 when you're done.

Now we must make a statement about the severity level, for construction, of *moderately deep* soils. We can't reject them out-of-hand, and neither can we say without qualification that they have no limitations. We must consider other factors.

Notice that 'F4' in the annunciator is marked 'ins >', short for insert subtree. *Press* **F4** now. Again we see the list of land characteristics, but this time it does not include soil depth, 'sd', since that characteristic has already been used on the current path. Suppose that slope is the next most important factor in determining suitability for construction. So, *select* 'sl' out of the list now.

Now things change quite a bit in the display. The first characteristic, soil depth, is *fixed* at branch 2, moderately deep; this is displayed to the right of the LC name. The other branches of soil depth have been *temporarily* removed from the display, and the next LC to be considered, 'slope',

---

[51]You may disagree with this assessment; remember that each model builder makes his or her own decisions based on local conditions and assumptions. It is certainly conceivable that it would be considered economic or otherwise worthwhile to construct ponds on shallow soils despite the difficulties; for the sake of this model we will assume that it isn't.

is displayed as a secondary heading, indented by one space. The four possible values of slope (along with '?') are the new branch headings. The highlight is on branch 1, which corresponds to nearly level slopes.

> Don't worry, the hidden parts of the decision tree are still there, they are just hidden 'behind' the currently-visible parts of the tree. Unfortunately the PC display screen in text mode is far too small to show the entire decision tree, so we can only show the active branch. If you are ever confused about where you are in the tree, you can navigate up and down levels with the arrow keys. Also, you can print the entire tree using Menu 1.5 ('Reports'), option 9 ('Decision Trees').

Let's continue building the tree. Suppose that on moderately deep soils, any slope over 30% is impractical for constructing fish ponds, because there is just too much earthmoving, it is very hard to stabilize the pond banks on such steep slopes, etc.. Let's record that fact now. *Move* the highlight to branch 4, corresponding to 30–200% slopes, and *press* **F3** to enter a decision. From the pop-up list, *select* severity level 4 (impossible). Now the '?' at branch 4 is replaced with an asterisk (signifying a decision), and the decision itself. The cursor has moved to branch 1; notice how it rotates around the branches.

We are now being asked to give a suitability rating for *moderately deep soils* on *nearly level slopes*. A problem with building fish ponds on nearly level sites is that extra drainage must be provided to allow the ponds to be drained and refilled with fresh water every four months or so.

Let's enter this decision now. The highlight is already on the first branch; *press* **F3** and *select* severity level 2 ('needs drainage') out of the pop-up list; this result is displayed and the highlight moves to the second branch, representing sloping lands.

Sloping land is *ideal* for fish pond construction, because simple outlets from each pond can be used to move water to ponds further down the hillside (so, no extra drainage is needed), and earth dug from the pond can be used for the immediately adjacent down slope dike (so, no extra earthmoving is needed). *Press* **F3** again and this time *select* severity level 1 ('no limitation') from the pop-up list; this result is displayed and the highlight moves to the third branch, representing steep lands.

Steep land will have no drainage problem, but will require extra earthmoving. Ponds are smaller and dikes are higher, resulting in more work for the same area of ponds.

Let's enter this decision now. *Press* **F3** again and this time *select* severity level 3 ('earthmoving') from the pop-up list; this result is displayed and the highlight moves to the fourth branch, representing very steep lands; we've already entered a result here.

You have now completed the first paths in this decision tree. The tree should look like this on your screen:

```
sd soil depth 2  m (moderately deep) [75-200 cm]
  sl slope
     1  A (nearly level) [0-3 %] * 2 (needs drainage)
     2 B (sloping) [3-10 %]     * 1 (no limitation)
     3 C (steep) [10-30 %]      * 3 (earthmoving)
     4 D (very steep) [30-200 %]        * 4
(impossible)
     ? [???]    ?
```

Now we've taken care of shallow and moderately deep lands. But what about deep lands? We have to *backtrack* in the tree and enter decisions for them.

> **√ use the 'left arrow' key '←' to move *up* one level**
> **√ use the 'right arrow' key '→' to move *down* one level**

To go back a level, *press* the left arrow. The last LC in the path, slope, disappears, and the branches for the first LC, soil depth, are redisplayed. Notice that to the right of the second branch (the one we just completed), is a '> sl'; '>' indicates a *complete subtree,* i.e. a subtree where all the leaves have been filled with decisions, is attached at that branch, and 'sl' is the land characteristic that begins the subtree.

Recall that as we made decisions for moderately deep soils, the depth never was a factor in our decision making. Thus there is no difference, from the point of view of construction potential, between moderately deep (branch 2) and deep (branch 3) soils. We could re-enter the same subtree as is presently rooted at branch 2, but we can also take advantage of another feature of the editor, *equating* or *joining* branches. Notice in the annunciator line that '=' and 'F7' are both marked 'join'.

> **√ '=' is a shorthand for 'joined' or 'equated'**
> **√ 'F7' or '=' may be used to join branches.**

Move the highlight to branch 3, and then *press* = ; a selection box will pop up, asking to which other branch we want to join branch 3. The default is branch 2, the one immediately above it. In this case, that's what we want, so *confirm* the answer. Now the '?' at branch 3 is replaced with an '=2', signifying a join, and the branch number to which it is joined, i.e. a '2'.

> We have saved some work using the join, since otherwise the subtree would have had to be duplicated. Furthermore, there is a *logical* difference between a join and a separate decision or subtree: in the former case we are stating that from the point of view of the current decision tree, there is *no difference* between the two branches. In the present case, we are saying that moderately deep and deep soils are the same from the point of view of this land use requirement, and so the additional critical limit of 200 cm between the moderately deep and deep soils is irrelevant (of course this limit may prove important for other land qualities).

You have now completed the tree. Only the top level is displayed:

```
sd soil depth
    1 s (shallow) [0-75 cm]                * 4 (impossible)
    2 m (moderately deep) [75-200 cm]> sl
    3 d (deep) [200-1000 cm]              =2
    ? [???]              ?
```

Decision trees can be *annotated*, exactly in the same manner as was done for the evaluation. It is a good idea to make a note that explains your reasoning to anyone who uses the model. Important things to explain are: (1) why you chose the decision factors, (2) assumptions about factors not in the tree, (3) an outline of the decision procedure in words.

√ 'Alt-F9' is used to enter or edit an annotation

Let's enter a note now. *Press* **Alt-F9** to bring up the note editor. Enter a note similar to the following, or, if you wish, you can add some comments of your own.

```
Potential for construction is based on:
   (1) soil depth -- can't excavate into bedrock by hand
   (2) slope --
      -- excessive slopes can't be stabilized
      -- steep slopes require more earthmoving
      -- nearly level slopes require extra drainage
```

When you're done entering the note, *press* **F10** to file it and return to the decision tree editor. We'll see this note again when we compute an evaluation.

Leave the decision tree editor by *pressing* **F10**. The tree disappears, and the overlapping windows reappear. *Press* **F10** to remove menu 1.2.7, and the choice box for the land use requirements is again the active window. Note that 'ppc' has been added to the list of LUR's for this LUT, and the highlight is on it. Also note that the LUR is marked with an asterisk '*', indicating that it has a decision tree, so that ALES can determine the value of the corresponding *land quality*.

7.3.6.3.4 Defining the other land use requirements

Now that you've completely specified the land use requirement 'ppc' (potential for construction) within the context of the current land utilization type 'tdp' (Tilapia in diversion ponds), you must specify the remaining two land use requirements: water availability and water quality. In this section we won't tell you how to complete interactions. By this time you should be familiar enough with ALES to navigate through it on your own. If you get confused, you can refer to the previous three sections, or just use the help screens. Remember that the annunciator line always shows which keys are active.

Water quality

Now *define* a new land use requirement for this LUT: 'wq' (water quality). In this model, water quality is determined by water temperature and pH. The temperature can't be altered, and neither can excessive alkalinity be neutralized. However, excess acidity can be corrected by additional lime added to the water each year (a recurrent expense). This land quality will have *four* severity levels: (1) 'no limitation', (2) 'needs extra lime', (3) 'lots of lime', and (4) 'impossible'. There will be one additional *annual* input, 'lime'. Severity levels 2 and 3 will require 2.5 and 5 extra quintals of lime per are per year, respectively.

Enter the following decision tree for water quality:

```
wt water temperature
  1 c (cold) [0-16 _C] * 4 (impossible)
  2 t (temperate) [16-23 _C]............................> pH (pH of water)
    1 vacid (very acid) [0-4 pH] ....................... * 4 (impossible)
    2 acid (acid) [4-6 pH] ................................> pe (permeability)
      1 s (slow)          ..................................... * 2 (needs extra lime)
      2 m (moderate)  ..................................... =1
      3 r (rapid)         ..................................... * 3 (lots of lime)
    3 n (neutral to slightly alkaline) [6-9 pH]... * 1 (no limitation)
    4 valk (very alkaline) [9-14 pH]................* 4 (impossible)
  3 (warm) [23-35 _C] ....................................=2
  4 (hot) [35-100 _C] * 4 (impossible)
```

Of course as you enter the tree, you will not be able to see it all at once on the screen; only the current path will be visible. After entering the tree, *enter* a note explaining that excessively acid, alkaline, warm, or hot waters are unsuitable for Tilapia, and that acid water can be corrected with lime; more lime is needed if water leaks out of the pond due to rapid subsoil permeability.

Water availability

Now *define* a new land use requirement, 'wa' (water availability). In this model, water availability is determined by water source (rain, river, or springs), months of rainfall, and subsoil permeability. None of these can be altered, so that there will not be any additional costs. Lower water availability will be reflected in lower yields (see below), i.e., there will be periods when there is not enough water to fill the ponds, hence no fish can be raised. This land quality will have four severity levels: (1) 'high', (2) 'moderate', (3) 'low', and (4) 'very low'. There are no additional annual or by-year costs.

Enter the following decision tree for water availability:

```
ws source of water
  1 rain       > rs (length of rainy season)
    1 short [0-6 months]       > pe (permeability)
      1 s (slow)          * 2 (moderate)
      2 m (moderate)          * 3 (low)
      3 r (rapid)             * 4 (very low)
    2 medium [6-10 months] > pe (permeability)
      1 s (slow)          * 1 (high)
      2 m (moderate)          * 2 (moderate)
      3 r (rapid)             * 4 (very low)
    3 low [10-12 months]     > pe (permeability)
      1 s (slow)          * 1 (high)
      2 m (moderate)          * 1 (high)
      3 r (rapid)             * 3 (low)
  2 river      = 1
  3 spring     * 4 (very low)
```

After completing the tree, *enter* a note explaining that springs are not a reliable water source, that both rainwater and river water depend on the length of the rainy season, and that the permeability of the subsoil determines how well the pond retains water.

Now you should have a list of three land use requirements for LUT 'tdp'. *Go back* from the list to the Menu 1.2 'Specification options for LUT 'tdp''.

7.3.6.4 Outputs

Now you will enter information about the outputs, i.e. the products of this land utilization type. This is one side of the *economic* evaluation[52]: the LUT produces outputs which have a cash value. Notice that menu choice 6 is for 'Outputs'; *select* it now.

A list box should now appear, showing the outputs of land utilization type 'tdp'; of course at present there are none, so the list has only the dummy item. As usual, *press* **F3** to add a new output to the list. You will see a selection list of all the outputs that were defined in the reference list of outputs, in this case, only one (Tilapia). This land utilization type will only have the one output. The highlight should already be on that choice, so *confirm* it now.

Now a new menu, 1.2.6, appears, titled 'Specification options for LUT, Output "tdp, ti"'. Notice that the first four menu choices have something to do with *yield*, since yield is the basis of the output side of an economic evaluation.

7.3.6.4.1 Optimum yield, number of crops per rotation

The first thing to do is tell ALES how much yield we expect under 'optimum' conditions, within the context of the LUT and evaluation area. This establishes a baseline for proportional yields

[52]Inputs are the other side.

which will be defined later. *Select* menu option 1 now. A data entry form should appear, asking for two items:

(1) Optimum yield, in quintals per are, and

(2) the years in the plan when the crop is harvested.

The terminology 'optimum' yield is a bit misleading. Really we are talking about the 'expected' yield under optimum conditions. See §8.3.4.2 for a fuller discussion.

Answer question 1 with `.035`, i.e. under optimum conditions we can expect .035 quintals of fresh Tilapia per are per crop; this is equivalent to 35kg per 10 meter–by–10 meter pond. *Confirm* the answer and move to question 2, 'years when harvested'.

The level at which this figure is set is determined by (1) the *ecological context* of the model's *area of applicability*, (2) the *socio-economic* context, (3) the *management level* we assume, and (4) whether we deal with *uncertainties* as optimists, pessimists (risk-averse), or by considering the average. This issue is covered fully in §8.3.4.2.

Recall that we defined a five-year planning horizon in the basic information for the land utilization type. Under optimum conditions, there are three crops of fish per year. However, if there is a water shortage[53], there may be two, one, or even no crops of fish per year. We will take this into account when we enter proportional yield factors. For now, we will enter each year number three times, to indicate the three harvests of fish in each year. The default answer for question 2 is '1, 2, 3, 4, 5', i.e. in the absence of more specific information, ALES assumes that there is one harvest of each output each year. You will have to *erase* this default (it's easiest to use 'F4') before *entering* the correct answer to this question , which is:
`1,1,1,2,2,2,3,3,3,4,4,4,5,5,5` *Confirm* the form and return to menu 1.2.6.

7.3.6.4.2 Determining proportional yield

We have just told ALES that each crop of fish on optimum land will yield .035 quintals per are. Now it is time to *differentiate* the lands, based on the *severity levels* of their land qualities. We must tell ALES how much of the optimum yield we can expect under various combinations of less-than-ideal conditions.

The main benefit of ALES for *economic* land evaluations is that it allows the evaluator to differentiate land areas on the basis of *yields*, levels of *inputs*, or both.

In the present simple example, only one of the three land qualities that we have defined for this land utilization type affects yields: 'wa' (water availability). Certainly the other two qualities can make Tilapia production *impossible*: if we can't construct the ponds, or can't obtain good-quality water, we can't grow fish. However, these two will be handled in the *physical* suitability (below), since either they make production impossible, or they allow full production. This simplifies the proportional yield calculation considerably.

---

[53] We defined a land quality 'wa' for water availability to represent this factor

Recall that we entered additional inputs of lime for the two intermediate severity levels of the land quality 'wq' (water quality), in §7.3.5.3.4. The additional lime completely corrected the limitation due to low pH, so that the growth of the fish will not be lowered in these situations. Therefore, 'wa' does not appear in the proportional yield decision tree. If we had elected not to correct this limitation, we would not have entered additional inputs. This would result in lower costs of production on map units with the two intermediate severity levels of 'wq'; however there would also be a lower yield (due to the slower growth rate of the fish), so that we would have had to include 'wq' in the proportional yield decision tree. This discussion illustrates how a limitation can affect costs, or returns, or both.

Let's enter the *proportional yield decision tree* now. *Choose* menu option 2. You will see the decision tree editor again, with no tree yet entered. *Press* **F4** to choose the first entity. You will notice that the pop-up list of choices now contains *land use requirements*, rather than the land characteristics that were the entities in severity level decision trees. This is because we are at a *higher* level of abstraction: land characteristics determine land qualities, which in turn determine yields.

Select 'wa', water availability, out of the list, as we consider it to be the only quality that affects yield (other than to absolutely prohibit production). The land quality name now heads the top level of the tree, and the four branches are listed below this heading. The highlight is on branch 1, corresponding to 'high' water availability. We can enter a final decision now, since no other factors will considered; so *press* **F3** to enter the decision.

Now you will see an entry box, rather than a pop-up list of choices. Proportional yields are decimal fractions on the interval [0..1], with '0' representing complete lack of production (e.g., crop failure), and '1' representing the optimum. Since we are entering the proportion for the high-quality water, the answer must be '1'; *enter* `1` now and *confirm* the answer.

The decision now appears in the tree, and the highlight has moved to the second branch, which corresponds to moderate water availability. This land quality was defined so that moderate availability means that we can only get two crops, i.e. the proportional yield is approximately 67%[54]. In the same manner as the preceding paragraph, *enter* '.67' for this decision (remember, the numbers are entered as proportions, not percentages).

Finally, enter '.33' for the third branch, and '0' for the fourth. The decision tree should look like this:

```
wa water availability
  1 1 (high)  *1
  2 2 (moderate)   * .67
  3 3 (low)  * .33
  4 4 (very low)   * 0
  ? [???]      ?
```

[54]It would be misleading to be more precise than two decimal places, given the approximate nature of the models and data.

If you like, you can *enter* a note now explaining the reasoning of the previous paragraphs.

When you are done, *press* **F10** to leave the decision tree editor and return to menu 1.2.6. There's nothing else to say about yields for this LUT, so *press* **F10** to go back to the list of outputs.

As we will see in Tutorial 4, there are other ways to define proportional yields. In this simple case, the result would have been the same with all three: the decision tree, limiting yield factors, and multiplicative yield factors.

7.3.6.5 Production-dependent inputs

ALES allows the model builder to specify that certain costs are only incurred when a crop is harvested, and are incurred in proportion to the *amount harvest*, i.e. the yield. These are called *production-dependent inputs*. Notice that menu option 5 is to enter 'Production-dependent inputs'; *select* this option now. You will see a choice box, titled 'Production-dependent inputs for LUT "tdp", Output "ti"', with only the dummy item.

Clearly, labor is needed when harvesting the Tilapia fish to process and package them. Naturally, the more fish, the more work. *Press* **F3** to add an item to the list, and *select* 'labor'. Now you will be asked how many days of labor *per quintal of fish* is required. The key point is that the amount of work depends on how much fish was produced, which you can see by the units of measure.

Enter `.2` (i.e., one person can process 5 quintals in one full day) and confirm the form. There are no more production-dependent inputs, so *press* **F10** to go back to Menu 1.2.6. We don't need to say anything more about output 'Tilapia', so *press* **F10** to go back to the list of outputs. There are no more outputs for this land utilization type, so *press* **F10** again to return to menu 1.2.

7.3.6.6 Determining physical suitability subclasses

We now must tell ALES how to determine the *physical* suitability of each land area from its set of *severity levels* of the land qualities. A physical suitability assessment is always necessary to exclude areas that are unsuitable; such areas must be placed in the *highest-numbered* physical suitability class, since this is the class that corresponds to FAO suitability subclass 'n2', i.e. 'physically unsuitable'. In this evaluation, that is class 2. Recall that we only defined two physical classes, corresponding to FAO orders 'S' and 'N'.

ALES does not carry out an economic evaluation of land that is rated as 'physically unsuited' for a Land Utilization Type.

To specify how ALES should determine the physical suitability of each land unit, you will build a *physical suitability subclass decision tree*. (Another way of determining physical suitability will be discussed in Tutorial 4, see §7.4.8.) Notice that option 8 of menu 1.2 is marked 'Physical Suitability Subclasses'; *select* option 8 now.

Menu, 1.2.8 'Physical suitability options for LUT: 'tdp'', with three items, should now appear. *Select* the first item, 'Enter/edit physical suitability subclass Decision Tree'. You will see the decision tree editor again, with no tree yet entered. *Press* **F4** to choose the first entity. As in the proportional yield tree, the decision entities are the three land qualities. All three will figure in the physical suitability, since all three can result in such a severe limitation as to prohibit the land use.

In this example, the LUR's are equally important for physical suitability, so it doesn't matter which entity we pick first, so let's pick the first one on the list, 'ppc'. *Select* 'ppc' now; the first level of the tree should now appear, with four branches corresponding to the four severity levels.

Let's first consider land with *no limitations* for construction. The highlight should already be on this branch. We now need to consider the other factors, so *press* **F4** to expand the tree again, and then *select* 'wa'. Let's consider land with no limitations either for construction or water availability. *Press* **F4** yet again, and *select* the last land quality, 'wq'. The tree should now look like this:

```
ppc potential for construction 1 (no limitation)
 wa water availability 1 (high)
  wq water quality
   1 1 (no limitation)     ?
   2 2 (needs extra lime)  ?
   3 3 (lots of lime) ?
   4 4 (impossible) ?
   ? [???] ?
```

We are being asked to rate land with (1) *no limitations* for construction, (2) *high* water availability, and (3) *no limitation* due to water quality. What's not to like? Clearly this is suitable for the proposed land use. *Press* **F3** to make a *decision*. An entry box appears, asking for the subclass, which can be either 1 or 2, possibly with some suffixes (we'll see these below). *Enter* `1`, and confirm it; the first branch of the tree is now marked '* 1', meaning that a decision has been reached, and land with this combination of factors would be completely suitable. The highlight has moved to the second branch.

Now we are considering land with only a *slight* water quality limitation, i.e. 'needs extra lime' (branch 2) and no other limitations. Recall that this limitation is going to be corrected with added lime; it will affect costs but not yields or physical suitability, since it does not make the land unsuitable. So the decision is the same as the previous branch. Indicate this by *joining* it to branch 1; press = and *accept* the default branch number of '1'. The same reasoning applies to the third branch; *join* it to branch 1 also. Now for the fourth branch, we have a physically limiting condition: the water quality is 'impossible' to correct. *Press* **F3** to enter a decision.

The suitability *class* number should be '2', corresponding to FAO order 'N' (not suitable)[55], so *type* `2`. Notice that there is some space after the '2' in the entry box. This is so we can enter a *subclass suffix*, which indicates the nature of the limitation in reports. In this case, the land is in physical suitability class 2 *because* of its water quality limitation, so we can indicate this by *typing* the corresponding suffix, `wq`, after the class number. So the final decision is '2wq'; *confirm* this and you will see it in the tree for the fourth branch.

---

[55]Remember that in this LUT there are only two physical suitability classes, 1 = 'S', 2 = 'N'.

Now we must backtrack in the tree and enter the physical suitability for other combinations of severity levels. *Press* the back arrow '←' to go back a level. Now we're considering other severity levels of water availability, still with no limitations to construction. Recall that levels 2 and 3 of water availability will affect yields (by decreasing the number of crops), but will not prevent the land use. So we can *equate* them with branch 1; *do* this now. The tree should look like this:

```
ppc potential for construction  1  (no limitation)
  wa water availability
    1 1 (high)  > wq
    2 2 (moderate)     =1
    3 3 (low)=1
    4 4 (very low)     ?
    ? [???]   ?
```

Let's learn another decision tree technique now: *cutting and pasting*. This is useful whenever one subtree shares much of its information with another, but is not exactly the same (in this case we could join the two subtrees). The basic idea is as follows:

√ A subtree is *cut* out to the paste buffer.
√ It is immediately *pasted* back in the same place, since it is still valid there.
√ We then move the highlight to another branch...
√ ...and *paste* the *same* tree in this new location
√ Finally, we *edit* the pasted subtree, making appropriate changes.

This process is very similar to the cut-and-paste functions of most word processors. The cut subtree remains in a special buffer, called the *paste buffer*, until a different subtree is cut, and can be pasted as many times as desired (even into a different tree); i.e. the paste key functions like a 'rubber stamp'. Only one subtree at a time can be in the paste buffer.

√ **'F5' is used to *cut* a subtree to the *paste buffer*.**
√ **'F6' is used to *paste* a subtree from the *paste buffer*.**

Why are we interested in cutting and pasting now? Because for the fourth branch we want to consider the 'wq' land quality, exactly as we did in the first branch. As you will see, there will only be slight editing of the pasted tree.

Let's see how this works. *Move* the highlight to the *first* branch (i.e. the one marked '> wq'), and *press* **F5** to *cut* it out; notice that the '> wq' is replaced with a '?'. Now that we've got a copy, *press* **F6** to *paste* it back at the same place (branch 1). Now *move* the highlight to branch 4, and *press* **F6** again to paste it there.

Now let's edit the pasted subtree. *Press* the right arrow '→' to *expand* the subtree; notice that it exactly the subtree we built under branch 1, but as you can see from the headings, we are now considering the case where we have no limitation to construction, and very low water availability. The decision for the first branch is class 1, but that can't be correct now, since we have low water availability. To edit this decision, *press* **F3** to enter a decision. The current value of '1' appears in the entry box; *erase* it by backspacing over the '1', and *enter* 2wa, signifying that the land would be in class 2 (unsuitable) *because of* its water availability limitation. *Confirm* your entry.

Branches 2 and 3 are still equal to branch 1. *Move* to branch 4, and *press* **F3** to edit the decision. As it stands, this is '2wq'; we still have the water quality limitation, but *also* a limitation due to water availability. We want to indicate both limitations in the physical suitability subclass name. To do this, *type* /wa after the current entry (there is no need to erase what is already there, we are only adding more to it), and *confirm*. The tree should now look like this:

```
ppc potential for construction  1  (no limitation)
  wa water availability  4 (very low)
    wq water quality
      1 1 (no limitation)        *2wa
      2 2 (needs extra lime)   =1
      3 3 (lots of lime)=1
      4 4 (impossible) * 2wq/wa
      ? [???]  ?
```

It is not necessary to include all the limitations in the subclass name. We could, for example, have decided that the water quality was so much more important a limitation than the water availability that the decision for branch 4 was '* 2wq', omitting the '/wa'. The only function of the subclass name is to indicate the reasons for physical unsuitability to the planner or whoever else looks at the evaluation results.

Now that this subtree has been edited, go back a level by *pressing* the back arrow key '←'. This second level is also complete, so go *back* to the first level. Again, branches 2 and 3 are the same as branch 1, as the intermediate severity levels do not change the physical suitability, only the cost of construction. So, *equate* both branches 2 and 3 to branch 1.

Now we must consider lands that are unsuitable for pond construction, i.e., branch 4 of 'ppc'. Certainly all decisions in a subtree rooted at this point must include '2ppc' in their names. In addition we want to consider water availability and water quality, to indicate multiple sources of unsuitability to anyone using the evaluation results. Since we've already considered these in the subtree that is rooted at branch 1, let's use the cut and paste method again, and edit the decisions to add 'ppc' to the subclass names.

Cut the subtree at branch 1 (marked '> wa'), and *paste* it back at the same place. *Move* to branch 4, and *paste* the subtree again. Now *expand* the subtree at branch 4, and, at the next level down, *expand* the subtree at branch 1. Now we see the four leaves; the first one should be *edited* from '1' to '2ppc', and the fourth one from '2wq' to '2ppc/wq'. *Go back* a level, and then *expand* the subtree at branch 4. *Edit* the first leaf from '2wa' to '2ppc/wa', and the fourth from '2wq/wa' to '2ppc/wq/wa'. This last subclass is as bad as can be: severe limitations for construction, water availability, and water quality!

We're done entering the physical suitability subclass decision tree, so *press* **F10** to leave the decision tree editor and return to Menu 1.2.8; *press* **F10** again to return to Menu 1.2.

7.3.6.7 Determining economic suitability classes

One component of the model remains to be defined: *economic suitability classes*. These are simply divisions of the whole range of predicted net present values or gross margins, to allow ALES to assign land to one of the four FAO suitability classes: S1 = 'highly suitable', S2 = 'suitable', S3 = 'marginally suitable', N1 = 'economically unsuitable'. You can not change the number or names of the FAO classes, but you can determine which land falls in each class, by entering economic class limits, as explained in this section.

> If the FAO economic suitability classes are not required as outputs of the evaluation, this step can be omitted.

Economic class limits can be set for both gross margin and NPV analysis. The first is a better measure of the year-to-year farm income, and the second is a better measure of project feasibility. For this example, we'll look at farm income.

Select menu option 3, 'Economic parameters', and in Menu 1.2.3, *select* option 2, 'Economic class limits - gross margin analysis'. A data entry form appears, asking for the *lower* limits of classes S1, S2, and S3, expressed as $/are. There is no lower limit of class N1; class N2 is reserved for land that is physically unsuitable (in the current example, this is any land that is placed in physical suitability class 2 by means of the physical suitability decision tree).

> The economic class limits are based on the *objectives* of the evaluation. The lower limit of S3 is the level at which the farmer would elect not to carry out the enterprise; typically this is zero (assuming that labor has been included in the costs). The lower limits of S1 and S2 correspond roughly to the NPV one would consider 'excellent' and 'good', respectively. This depends completely on the socio-economic setting. In particular, the per-unit area return that is considered adequate may depend on the typical holding size, to ensure an adequate per-farm income. See §8.2.3 for details.

In the present case, *answer* the three questions with '100', '50'. and '0' [\$ ha^{-1} yr^{-1}]; *confirm* the form and return to Menu 1.2.8. At this point we could, if we wished, specify class limits for the NPV. Instead, *press* **F10** to return to Menu 1.2. We're all done specifying this LUT, so *a* **F10** to go back to the list of LUT's.

In a real evaluation we would certainly define more than one land use alternative (i.e. LUT); in the interests of keeping the tutorial to a manageable length we won't do this. *Press* **F10** to return to menu 1.2; *press* **F10** again to return to Menu 1, 'Main Options'.

Congratulations, you have defined a simple evaluation model!

7.3.7 Defining data entry templates

Before we can compute and view evaluation results (main menu option 4), we must define map units and enter data (main menu option 3). Both of these were completely covered in Tutorial 2. However, there is one aspect of data entry that was not covered in that tutorial, since it is the responsibility of the model builder: defining data entry templates.

Data entry templates are used to specify the land characteristics for which data is to be entered, and their order in the data entry form that will be filled in by the model user. The model builder can define templates that group different sorts of data (e.g. climatic variables in one template, soils variables in another); a related purpose for templates is to specify the order in which data will be read from an external data base[56].

Let's define a template now. *Choose* option 3, 'Data', from the main menu, and then option 7, 'Templates', from Menu 1.3 'Data'. An empty list of template codes should now appear. *Add* a new entity (with 'F3'), and for its code *enter* `All`, because all the land characteristics will be fields in this template. *Confirm* this code, and ALES asks you for the *descriptive name* of the template, *enter* `all land characteristics`. *Confirm* this form, and you will see menu 1.3.7, 'Specification options for Data Entry Template', with two menu options. We've already entered a descriptive name, so *select* menu option 2, 'List of data fields (land characteristics)'.

Now an empty *list box* appear at the upper right of the screen. The list will contain the codes of the land characteristics to be included in the template. This is a bit different from other list boxes we've seen, since the list is not ordered by the en tity code, but rather the *numeric position* in the list. This is because the order of fields is important both for the data entry form to be presented to the model user, and for matching the order of fields in any automatic input.

Press **F3** to insert the first field. A list of all ten land characteristics that are defined in the reference lists should now appear. All of these must be added to the template. Let's group them according to the kind of characteristic: water characteristics first, climate second, and soil third.

Move the highlight to the first water characteristic, 'pH', and *press* **Return** to select it. Notice that it is now the first item in the list, meaning that the first field in the data entry form will be the land characteristic 'pH' of the water.

Press **F3** again to insert the next field. Notice that the characteristic that we've already used for the first field, i.e. 'pH', is no longer in the selection list. *Select* 'ws' (water source) and *confirm* this choice; notice that it is now field 2.

Suppose that we intended that a different characteristic, water temperature, be the first field. We can insert this land characteristic *before* the fields we've already defined. *Move* the highlight to field 1, which at present is 'pH', and *press* **F3** to insert a new field. *Select* the characteristic 'wt' (water temperature) and *confirm* it; now notice that 'wt' is the first field, and 'pH' and 'ws' have been pushed down to fields 2 and 3, respectively.

> It is also possible to change the land characteristic already assigned to a field without pushing down others; 'F5' is used for this.

Now enter the rest of the characteristics at the end of the list. *Move* the highlight to field 4, i.e. the end of the list (you can use the 'End' key). Using 'F3' repeatedly, *insert* the remaining characteristics in this order: 'rs', 'sl', 'sd', 'text', 'cfc', 'cfs', and 'pe'. The list of fields should now look like this:

---

[56] See Chapter 10 for more on reading data from an external database.

— Data Fields for 'All' (all land characteristics) —	
1	wt (water temperature)
2	pH (pH of water)
3	ws (source of water)
4	rs (length of rainy season)
5	sl (slope)
6	sd (soil depth)
7	text (texture of fine earth)
8	cfc (coarse fragment content)
9	cfs (coarse fragment size)
10	pe (permeability of subsoil)
11	*< add to end of list >*

Since we're done with this list, with this template definition, and with defining templates, *press* **F10** *three times* to go back to the 'Data' menu, 1.3.

7.3.8 Debugging an evaluation from the 'Why?' screens and evaluation matrix

Now that the model is complete, and a data entry template has been defined, we can define land units, enter land data, and compute evaluation results exactly as in Tutorial 2. Rather than go through the mechanics of all that, we'll just present a table of hypothetical land units and their data[57]. We've set these up to so that they will have different physical and economic suitabilities. *Define* these units, *enter* the data, and *compute* the evaluation. Then *display* the evaluation matrix for *physical suitability*. (If any of this is unclear to you, review Tutorial 2).

	Land Characteristic code									
LMU code	**wt**	**pH**	**ws**	**rs**	**sl**	**sd**	**text**	**cfc**	**cfs**	**pe**
A	c	n	rain	long	A	d	c	1	b	m
B	t	n	river	medium	B	m	m	2	fg	s
C	w	acid	spring	short	C	s	m	4	b	r
D	w	acid	rain	medium	A	d	f	2	fg	s
E	t	acid	rain	short	C	m	m	1	–	s

In Tutorial 2 we took a brief tour of the 'Why?' screens. However, in model building mode we can do more than examine why and alter economic parameters: we can alter any relevant part of the model, including decision trees, from the 'Why?' screens. In the present section we'll practice editing the model and re-computing the result.

[57] These are *not* real land units from Guatemala.

At this point you should be looking at the evaluation results matrix which shows the physical suitability subclasses of the five test map units for the one land utilization type that we've defined, namely, Tilapia in diversion ponds. Note that two of the units, 'A' and 'C', are physically unsuitable, but for different reasons: 'A' has poor water quality, and 'C' has both excessively low water availability, and the ponds can't be constructed in the first place. Map units 'B', 'D', and 'E' are physically suitable, according to our model.

The model builder should look at the evaluation results matrix for *anomalies* between map units, or *inconsistencies* with what is known from experience. Let's suppose that we know from experience that map unit 'E' is in fact *not* suitable for fish ponds; construction was attempted at a test site, but abandoned because it proved too difficult to dig the ponds; bedrock was exposed on these steep slopes.

Let's see where the model went wrong and adjust it. *Move* the highlight (with the arrow keys or the prefix method) to map unit 'E', and *press* **F2** to initiate the series of 'Why?' screens. The first such screen shows the severity levels of the three land qualities, and their combination, by means of the physical suitability decision tree, into a physical suitability subclass. From this screen you can see that land quality 'ppc' has severity level 3, i.e. earthmoving is required, but construction is possible. Since we know from experience that construction is too difficult, this severity rating must be wrong. Let's find out why.

To go back one more step, to determine why the potential for construction has severity level 3, *press* **F2** again, and select 'ppc' out of the list of land use requirements. Now we can see the decision tree path actually taken for this land unit. Let's assume that the relevant data values are correct, i.e. moderately deep soils and steep slopes. (Of course these could have been wrong, in which case they could be edited as explained in Tutorial 2). The problem is that we incorrectly combined the two land characteristic values in the decision tree: on steep slopes with only moderately deep soils, excavating for ponds will expose the bedrock, which we can't dig into with hand tools. So we must *edit* the decision tree.

Notice that in the annunciator line, 'F3' is marked 'Edit severity level DT'.

> **In the 'Why?' screens, all model factors and/or data values that are relevant to that screen can be *edited*; the annunciator line shows which keys are active, and 'F1' help explains the options.**

Press **F3** now to enter the decision tree editor for this tree. The first level is displayed. Recall that we are going to alter the decision for moderately deep soils on steep slopes. But you can see from the editor that deep soils are equated to moderately deep soils; that will have to be changed.

Our first step, therefore, is to *cut* out the subtree at branch 2 using 'F5', and *paste* it back at both branch 2 and branch 3 (using 'F6'). The pasted subtree at branch 3, for the deep soils, remains correct, but we must edit the subtree at branch 2. *Move* back to branch 2 and *expand* it. Now you can see the incorrect combination at branch 3 (steep slopes). *Move* to this branch and *press* **F3** to edit the decision; in the pop-up box *select* choice '4' ('impossible'); this now replaces the previous choice '3' ('more earthmoving').

We're done editing this tree, so *press* **F10** to return to the last 'Why?' screen. Now let's see how this change affects the evaluation results. *Remove* the stacked 'Why?' screens (with 'F10' or

'Esc') and, at the evaluation matrix, *press* **F3** to force ALES to *recompute* the results. After a few seconds the new results are displayed; note how map unit 'E' is now in physical suitability class '2ppc', i.e. unsuitable because construction is impossible. Any *changes* to the suitability are *highlighted* in the matrix cells. Thus the change to the decision tree had the desired effect, at least for map unit 'E'[58]. *Press* **F10** to remove the matrix and return to the list of report types.

At this point you should *experiment* with several of the report types, such as **Physical suitability** subclass, **Net Present Value**, **Benefit/Cost ratio**, **Internal Rate of Return**, **Gross margin**, **Land** quality values, and **Yield**. Find out what kind of information is presented in each one, and follow the chains of 'Why?' screens, noting on each one how the information was derived, and which parameters or data can be edited at that point. Remember that help is always available, and the annunciator line shows which keys are active.

When you are done experimenting, *press* **F10** enough times to arrive at the list of evaluation models. At this point you should *make a backup* of your model to a DOS file as described above in §7.3.3. Then you can *exit* from ALES.

Congratulations, you have completed Tutorial 3! Although the model you built is fairly simple, it illustrated much of the expressive powers of ALES. In the following tutorial, you'll experiment with some advanced model-building techniques.

7.4 Tutorial 4 - Refining a model

The objective of this tutorial is to teach you how to incorporate some of the more sophisticated capabilities of ALES into your models. In this tutorial you will learn to:

- *copy* evaluation models
- set up *related land utilization types*
- use advanced features of the *decision tree editor*
- define *continuous* land characteristics
- *infer* land characteristic values from other data
- enter *multi-valued* land data
- enter *proportional* and *limiting* yield factors
- use the *maximum limitation method* to define physical suitability
- *delete* evaluation models

Before working on this tutorial, you should have completed Tutorials 1, 2, and 3. This tutorial will be fast-paced, as we expect you to be comfortable with ALES by now. We will emphasize the variety of options that ALES provides for expressing expert knowledge about the relation between land and land use.

---

[58]You should always check to see if *all* the changes are as expected; sometimes fixing one rating causes others to change unexpectedly.

7.4.1 Copying an evaluation model

The model to be used in this tutorial will be a *refinement* the model used in Tutorial 2. We don't want to edit model 'Tutor2', since others may want to use it in its original state for their own tutorial exercise. Instead, we'll make a *copy* of it, and work on the copy. This is a very useful procedure for experimenting with someone else's model or basing one evaluation model on another. *ALES keeps all evaluation models completely separate*, so if you work on the copy of a model, you can't inadvertently delete or modify the original.

> √ **ALES keeps all evaluation models** (reference lists, databases, land utilization types) **completely *separate*.**

Start ALES in model-building mode, and in the list of evaluation models select the model that was used in Tutorial 2, 'Tutor2'.

> If 'Tutor2' does not appear in the list of evaluation models, you must first *load* it into your list of ALES models. The saved model 'tutor2.als' is included in the archive 'examples.zip' on the distribution diskette. Extract it to any convenient place on your hard disk with the 'unzip' program (also found on the diskette) and then load it into ALES by using 'F8' from the list of models.

Notice that 'F6' in the annunciator line is marked 'Copy'. *Press* **F6** now. ALES presents an entry box for the new code for the copied model. *Enter* any short code that you wish; a reasonable choice would be 'Tutor4'.

After you confirm your entry, the 'Working...' message appears on the screen while ALES copies the entire model and its data base. Since there is a lot of information to copy, this will take a while (about 45 seconds on a 12MHz PC/AT). When the message disappears, the copy is complete, and you will be asked to edit the basic information for this new model. Notice that this 3-item data entry form is exactly the same form you had to fill in when defining a 'new' model, but here the answers are already filled in based on the evaluation you copied from.

Clearly, the answer to the first question, 'Description', is wrong. *Replace* it with a description that explains what it is, for example 'Tutorial 4 - to learn advanced ALES techniques'. *Confirm* the form, and you will be returned to the list of evaluations. The newly-copied model appears in the list, and the highlight is on it.

Notice that it is marked with a '°', indicating that there is already a note for this new model. This is because the note was copied along with the rest of the model. Let's see if the note is still accurate. *Press* **Alt-F9** to bring up the note editor.

The first two lines are now incorrect: we need to change the references to tutorial '2' to number '4'. The cursor is now on the first character of the note. Move it right to the first '2' by *pressing* **Ctrl+→** (the 'Ctrl' key and the right arrow key, together) three times. this moves the cursor right by one word). Let's practice another way to type in the note editor: *typeover* mode. *Press* the **Ins** key (usually located on the numeric keypad), and notice that the bottom right of the screen now shows the letter 'T' (which stands for 'typeover'), indicating that any characters we type now will *replace* existing ones, not push them to the right as in the default 'insert' mode. So, *type* 4 now, and notice how the '2' has been replaced by the '4'.

Now move the cursor back to the beginning of the line by *pressing* **Home**. Move down one line by *pressing* the down arrow '↓'. *Move* three characters right with the right arrow '→'; now the cursor should be on the '2' in the number '7.2'. *Change* this to a '4'.

Press **Ins** again to return to 'insert' mode. Notice how the 'T' disappears from the lower right of the screen.

Now you can edit the note some more if you want. You might want to add your name as a co-author. Remember that 'F1' will give you a complete description of the editing commands. When you're done, confirm your changes by *pressing* **F10** and return to the list of evaluations.

7.4.2 Setting up related land utilization types

Now we want to expand the model. ALES makes it easy to define *related land utilization types*, i.e. LUT's that are similar, but which differ enough so that they represent different land use alternatives. The easiest way to build a set of related LUT's is to first build the *base* LUT's, get these models just as you want them, then *copy* them to related LUT's, and edit the newly-copied LUT's to reflect the differences.

> *Examples* of differences between related LUT's are:
>
> (1) different discount rates, e.g., market vs. subsidized;
>
> (2) different optimum yields, e.g., optimistic vs. pessimistic, or risk averse yield levels, or referred to a national vs. local context;
>
> (3) different decision procedures for the same land qualities;
>
> (4) different levels of added inputs, e.g., in one LUT you may elect to correct a limitation rather than take the yield loss);
>
> (5) different sets of Land Use Requirements.
>
> Clearly, there are many permutations of these reasons. With ALES you can compare any number of related LUT's side-by-side in the evaluation matrix. This allows 'what-if?' analysis based on differences in models.

In this tutorial, we want to experiment with different decision procedures for the 'continuous corn' LUT. This will allow us to compare two *different models* for the *same LUT*. This is appropriate if there is a difference of opinion between model builders, or if we would like to incorporate some more factors in the evaluation model, but aren't sure of their effect.

> In this case, the LUT does *not* change, only our way of *modeling* it.

Let's create the related LUT now. Bring up the list of LUT's for 'Tutor4' by *choosing* option 2 in the Main Menu. As you can see there are three LUT's already defined: 'ccc'. 'ccc-d', and 'ttpp'.[59] To copy the 'continuous corn' LUT, make sure the cursor is on LUT 'ccc', and *press* **F6** to initiate the copy.

ALES now asks for the new LUT code; enter `ccc-m` (the 'm' stands for '*m*odified model'). After you confirm your entry, the 'Working...' message appears on the screen while ALES copies the entire LUT, including all the decision trees. This will take about 15 seconds on a PC/AT. When the message disappears, the copy is complete, and Menu 1.2, 'Specification options for LUT: "ccc-m"', appears. Now we can edit any information for the LUT except for the number of physical suitability classes. This menu was covered in detail in §7.3.6 (Tutorial 3).

The first thing to change about this LUT is its descriptive name, so *select* menu option 1, and *edit* the data item, 'descriptive name', to reflect the purpose of this LUT. The new name should be 'corn grain - modified decision procedures'. Confirm the form, and return to menu 1.2.

7.4.3 More on decision trees

One of the purposes of this exercise is to teach you advanced techniques for editing decision trees. We'll alter the decision tree for one of the land use requirements, and then see how these changes affect the suitability ratings. *Select* menu option 7, 'Land Use Requirements', and in the list of LUR's, *select* 'm' (moisture availability). In Menu 1.2.7 ('Specification options for LUT, LUR: "ccc-m, m"') *select* option 4, 'severity level Decision Tree'. The decision tree editor will appear, with the first decision entity, 'dbr' (depth to bedrock) expanded.

7.4.3.1 Cutting and pasting

It is sometimes necessary to *restructure* a decision tree. Imagine that we were writing the tree out as a large, tree-shaped, diagram on a piece of paper. We could *cut* whole subtrees, and then either discard them, or *paste* them back at some other point. We could add a new *top-level* decision entity above all the ones we'd already considered; conversely, we could discard the present top-level entity. All these things can also be accomplished in the ALES decision tree editor. In this section you'll learn these techniques.

The decision tree for moisture availability, as it stands, takes into account four land characteristics: 'depth to bedrock', 'previous erosion', 'USDA texture of the subsoil', and 'May-to-September precipitation'. There are certainly many other factors that can affect moisture availability. For example, the current tree makes no allowance for rooting volume, other than the depth dimension. In particular, if there is a significant volume of coarse fragments in the rooting zone, the effecting rooting volume will be reduced, and moisture stress will be increased. Let's modify this tree now to take coarse fragment content into account.

The volume of coarse fragments could be incorporated at several levels in the tree, but it really makes the most sense to consider it as the *top-level* land characteristic. Then the first two levels, i.e. coarse fragments and depth to bedrock, will together define the potential rooting volume, and the lower levels of the tree, i.e. texture of the fines, previous erosion, and summer precipitation, will determine how much water this volume will hold.

---

[59]In fact, 'ccc' and 'ccc-d' were set up as related LUTs when the original Tutorial 2 was built.

So we want to add a new top-level characteristic. To do this, we will have to:

(1) cut the *entire tree* to the paste buffer;

(2) *add* the new LC at the top level

(3) *paste* the cut tree back as a *subtree* of the new top-level LC.

Notice in the annunciator line that 'F8' is marked 'cutT', which is a shorthand for 'Cut the entire Tree'. This key is only active at the top level of the editor.

√ **'F8' is used from the *top level* of the decision tree editor to *cut* the *entire tree***

Press **F8** now. The 'Working...' message appears for a few seconds, and then the entire tree is removed; the editor now shows an empty tree! Don't worry, the cut tree is now in the paste buffer. Notice that 'F6' appeared on the annunciator line after the cut, indicating that there is a cut subtree that can be pasted.

Now we want to add the new top-level LC. *Press* **F4** to select the LC, and from the list *select* the second item, 'cfv-A' (volume of coarse fragments - surface soil). The empty tree is replaced with a new one-level tree, showing the branches for this LC, none of which have a decision associated with them yet.

The cursor is now on branch 1, 'n' (not enough coarse fragments to be named). This situation is exactly like the one assumed for the original tree (i.e. the one we just cut): no coarse fragments, so we consider the other factors exactly as before. So let's paste the cut tree in here: *press* **F6**, and after a few seconds the '?' for this branch is replaced with a '> dbr', indicating a complete subtree whose first entity is 'depth to bedrock'. Now we are back where we were before the old tree was cut, at least for soils with no significant coarse fragments.

We now must consider the case where there are enough coarse fragments to reduce the rooting volume. *Move* the cursor down to the second branch, 's' (significant enough to be named, 15-50% by volume coarse fragments). We still must consider all the factors that were used in the cut tree, but we will want to edit some decisions. The *structure* of the cut tree is already correct, however. Remember that this tree is still in the paste buffer, so we can paste it in here (as well as at the first branch). *Press* **F6** again, and after a few seconds the '?' for this branch is replaced with a '> dbr'; this is exactly the same subtree that we pasted in for the first branch.

Now we want to edit some decisions for this branch, so *expand* the subtree to the next level. Notice in the first line that the value of 'cfv-A' is *fixed* at 'significant enough to be named'; we are now describing land that has 15-50% coarse fragments. The next factor to be considered, namely, depth to bedrock, is now expanded.

Assume for the sake of simplicity that in deep soils (branch 3 at this level) there is no change in moisture supply even with significant coarse fragments, because corn, being a deep-rooted crop, can extend its roots downward to exploit the same effective volume, compensating for the coarse fragments (this certainly would not be true for shallow-rooted crops). So the subtree at the third branch need not be changed. Also, the decision of 'severe stress' for shallow soils (the first branch) is still correct. So, we only have to edit the second branch, 'moderately deep'. *Move* the cursor to branch 2 and expand the subtree by pressing the right arrow.

Now we are considering previous erosion, on moderately deep soils with significant coarse fragments. Suppose that classes 3 and 4 (moderately severe and severe erosion) are now completely limiting, because not enough water can be stored in the reduced rooting volume. Right now branch 3 has a subtree, marked '> text-B'; we want to remove this and replace it with a final decision. However, if we use 'F5' to cut it, this subtree will replace the present contents of the paste buffer. Suppose we don't want to erase the contents of the paste buffer, but we do want to remove this subtree. In this case we use an 'alternative' to the 'F5' key, namely 'Alt-F5', to *clear* the subtree without replacing the contents of the paste buffer.

√ 'Alt-F5' is used to clear a subtree without replacing the contents of the paste buffer

Let's do this now. *Move* the cursor to branch 3 and *press* **Alt-F5**; you will be asked to confirm that you really want to erase the subtree without saving it in the paste buffer; *change* the answer from the default 'No' to 'Yes', and *confirm*. Now the '> text-B' subtree at branch 3 is cleared, and replaced with a '?'. *Press* **F3** to enter a decision, and select '3' (severe stress); now branch 3 is marked '* 3 (severe stress)'. Note that branch 4 (severely eroded) is already equated with branch 3, so that changing the decision for branch 3 also changes it for branch 4.

It remains to consider the uneroded soils at branch 1. *Move* the cursor to this branch and *expand* to the next level. Now we see a list of USDA textures. Let's suppose that some of the textures with less water-holding capacity now become severely limiting without even considering the summer precipitation. Using the same technique as in the previous paragraph, *clear* (but don't save in the paste buffer) the '> pptMS' subtrees for branches 3 (sandy loam), 6 (silt), 8 (clay loam), and 12 (clay), and *equate* them with the decision for branch 1 (sand), i.e. '* 3 (severe stress)'. At the same time, *equate* the decision for branch 9 (silty clay loam) with branch 4 (loam). When you are all done, this level of the tree should look like this:

cfv-a volume of coarse fragments **2** s (enough to be named) [15-50%]
  dbr depth to bedrock or root-impenetrable layer **2** m (moderately deep) [50-100 cm]
    erode previous erosion **1** n (not eroded)
    text-B USDA texture of subsoil

1 s (sand)	* 3 (severe stress)
2 ls (loamy sand)	=1
3 sl (sandy loam)	=1
4 l (loam)	> pptMS
5 sil (silt loam)	=4
6 si (silt)	=1
7 scl (sandy clay loam)	=6
8 cl (clay loam)	=1
9 sicl (silty clay loam)	=4
10 sc (sandy clay)	=8
10 sic (silty clay)	=8
12 c (clay)	=1
? [???]	?

> Notice in this example how branches can be joined into groups. In this example, the loamy textures 'l', 'sil', and 'sicl' form one group, with the highest water-holding capacity, and the other textures form another group. Only one representative of each group needs to be specified; in this example they are 'l' (loam) and 's' (sand). Because of the structure of the previous tree, more than one 'join' may have to be followed to determine a branch's grouping. For example, for 'sic' (silty clay), first the '=8' is followed to reach 'cl' (clay loam), and at this branch we see '=1', which is followed to reach 's' (sand).

Now we have to alter the final decisions for the remaining subtree at this level, to reflect the increased moisture stress. *Move* the cursor to branch 4, 'l' (loam), and *expand* it. As you can see, as the summer rainfall increases, the moisture stress decreases. But since we're now considering soils with lower water-holding capacity (due to the coarse fragments), the final decisions must be 'downgraded' one class. Branch 1 ('<400') is already in the worst class. *Move* to branch 2 ('400-450') and *replace* the current decision, 'moderate stress', with class 3 ('severe stress'); you can either enter a final decision with 'F3', or else equate this branch to branch 1, which already has the correct decision. *Move* to branch 3 and *replace* the current decision, 'adequate', with class 2 ('moderate stress'). Finally, *move* to branch 4 and *replace* the 'join' with a final decision of class 1 ('adequate'). So the entire group of decisions has been 'shifted' by one class.

> This is a typical use of cutting-and-pasting. The pasted tree now is placed under a more limiting value of a LC, so that some of the final decisions in the pasted tree must be downgraded.

This completes the editing of the subtree for 15-50% coarse fragments. Collapse the bottom four levels, by *pressing* the left arrow '←'four times, to arrive back at the top level of the decision tree, 'cfv-A'. We've completed the decision procedures for branches 1 and 2. Suppose that the very and extremely stony or rocky soils have too little moisture holding capacity without considering any other factors. *Move* to branch 3 ,'v', and *enter* the final decision of class 3 ('severe stress'). Then *move* to branch 4 and *equate* it with branch 3.

7.4.3.2 Alternate criteria and the 'Unknown' branch

Up to now we have ignored the branch marked '?' which appears below the list of numbered branches at every level of every decision tree. This branch is considered by ALES as it follows the tree while computing the suitability of a particular map unit, if the value of the land characteristic at a level is *missing* from the data base for that map unit.

For example, consider the decision tree we've just been editing. At the first level (which should still be on the screen), ALES will require the value of the LC 'cfv-A' (volume of coarse fragments) from the database when it tries to determine the land quality 'moisture availability' for a map unit. Suppose that there is no value of that LC for a particular map unit. As the tree stands now, ALES will not be able to compute the value of this LQ, and, since this LQ is used to determine proportional yield, ALES will not be able to compute economic suitability for this map unit.

Using the '?' branch, the model builder can specify an *alternate* decision entity to be used if there is no value of the main decision entity at this level. In the present example, we can specify an alternate LC to be used if no value is found for 'cfv-A'. Of course, this depends on there being a useful surrogate in the list of land characteristics. In the present evaluation model, there is.

√ ALES follows the '?' branch if there is no data value for the Land Characteristic

Move the cursor to the '?' branch, by using the arrow keys or by pressing '?'. Then *press* **F4** to insert a subtree. The list of available discrete LC's (not including 'cfv-A', which is already on the path) appears. *Select* 'fpsc' (family particle size class) from the list. This is a composite of the fine-earth and coarse fragments, used in the USDA "Soil Taxonomy" soil classification system at the family level of soil classification. The 'fragmental' class is roughly equivalent to 'very' (stony, rocky, etc.) textures, the 'skeletal' classes are roughly equivalent to stony, rocky, etc. (but not 'very') textures, and the other classes have no significant volume of coarse fragments. The correspondence is by no means exact; in addition the family particle size class refers to a particular control section of the soil profile, not the 'subsoil' as such. However, this is still a useful alternative criterion to the volume of coarse fragments, if that data item is not known.

Now we have to complete the decision procedure for this alternate criterion. For the first branch, 'frag' (fragmental), *enter* a final decision of class 3 (severe stress). *Join* the second branch, sandy-skeletal, and branches 4 (clayey-skeletal) and 5 (sandy) to the first branch, since they also would have severe moisture stress.

Loamy-skeletal soils (branch 3) are roughly similar to the situation we encountered before with significant coarse fragments and loamy textures; recall that an additional criterion of summer precipitation was then needed. So *add* a new level at this point, and *select* 'pptMS' as the decision criterion. At this lowest level, for the four classes *enter* the final decisions '3', '3', '2', and '1', and then *collapse* the tree to the previous level.

The subtree we just entered applies to soils with moderate water-holding capacities, either because of reduced rooting volume or because of unfavorable textures. The following family particle size classes can reasonably be *equated* with this subtree: 'col' (branch 6), 'cosi' (branch 8), 'fisi' (branch 9), 'f' (branch 10), and 'vf' (branch 11).

The remaining branch, number 7 (for fine loamy soils) represents the highest water-holding capacities. Again *insert* a subtree here, *selecting* LC 'pptMS' as the decision entity. At this lowest level, for the four classes *enter* the final decisions '3', '2', '1', and '1'.

> Note that these are 'upgraded' one class from the decisions for the soils with moderate water-holding capacity.

Now *collapse* two levels, back to the top level ('cfv-A'). Notice how the '?' branch is now marked with a complete subtree, '> fpsc'. This completes the editing of this tree, so leave the editor by *pressing* **F10** and return to menu 1.2.4.

7.4.4 Comparing related land utilization types

Let's see if the modified decision procedure changes the evaluation results. To do this, we will compute an evaluation for all the map units in the survey area, for both the original and modified LUT's. *Press* **F10** enough times to get back to the Main Menu. *Select* option 4, 'Evaluations', and from Menu 1.4 'Evaluations' *select* option 1, 'Compute an evaluation', Now you will be asked to select the LUT's to compute. Since we only want to compare the effect of the changes to the decision procedure, we need only select 'ccc' and 'ccc-m'; *move* the cursor in turn to each of these and *press* **F3** to include it in the set (notice the '♦' next to the selected items). *Confirm* the LUT set box; you will be asked to select the map units for which the evaluation should be computed. *Press* **F6** to select all the map units, and *confirm* the map unit set box; the computation now begins.

When the computation is complete, *select* option 2, 'View evaluation results', from menu 1.4. Instead of looking first at the overall suitability, let's see if the decision procedure affected the single-factor rating for the land quality of moisture availability, i.e. the land use requirement which we just edited. So, when asked for the report type, *select* 'L'and Quality values, and in the selection box which shows the LUR's which may be displayed, *select* 'm' (moisture availability).

Now you should be looking at an evaluation matrix which gives single-factor ratings for the LQ 'moisture availability'. The map units are the *rows* of the matrix, and the two LUT's we want to compare are the *columns*. You can see that the modification to the decision procedure *did* have an effect. Look, for example, at the row for the 'ErA' map unit: it has a severity level of '2' in column 'ccc' (the original decision procedure) and '3' in column 'ccc-m' (the modified decision procedure). *Display* the first 'Why?' screen for *each* of these cells in turn; you can see the extra LC that we added, namely, 'cfv-A', altering the path taken through the decision tree.

Why are some of the ratings the same for the two decision procedures? For example, *display* the first 'Why?' screen for map unit 'CeB' for *each* column; you will see that the added LC has a value of 'n', i.e. not enough coarse fragments to be named. Recall that we pasted the entire original decision tree below this new top-level branch in the new tree, so, the inferred LQ value remains the same.

Let's see now how the new decision procedure propagates from single LQ's to yields. *Leave* this matrix, and then when asked 'Which report?' again, *select* 'Y' (Yields). You will be asked to select an output; in the two LUT's we are comparing 'cg' (corn grain) is the only output, so *select* it out of the pop-up list.

Now you should be looking at an evaluation matrix which gives the yields of corn grain, in bushels per acre, for each map unit for the two related LUT's. Notice that some of the yields in the 'ccc-m' column are lower. Look, again at the row for the 'ErA' map unit: it has a yield of 81 bushels in column 'ccc' (the original decision procedure) and 60.75 bushels in column 'ccc-m' (the modified decision procedure). *Display* the first 'Why?' screen for each of these cells; you can see that increasing the severity level of the LQ 'moisture availability' from 2 (moderate stress) to 3 (severe stress) reduces the proportional yield from .6 to .45, according to the model builder.

If you wish, you can examine the reports for gross margin, net present value, and economic suitability class, to see how the difference in LQ value propagates to the final suitability. Some map units may drop from marginal economic suitability ('s3') to economically unsuitable ('n1'), due to the lower yield of corn, due to the lower predicted moisture availability.

When you are done with the evaluation matrices, *return* to the Main Menu.

7.4.5 More on land characteristics

So far in these tutorials we have considered only *discrete* land characteristics, i.e., those which are measured in *classes*. However, in some situations you may want to allow the model user to enter values on a *continuous* scale. Or, there may be an existing database with some or all data values reported on a continuous scale, which you want to use as the basis of the ALES database. In this section we'll see how a continuous LC can be used within ALES.

As an example, we'll suppose that single values, on a continuous scale, of 'May-to-September precipitation' are available for the land units to be evaluated. We will see how to define these in ALES, and use them in an evaluation.

7.4.5.1 Setting up a continuous land characteristic

The first step is to define the continuous LC. *Select* option 1, 'Reference Lists' from menu 1 'Main Options', and then *select* option 4, 'Land Characteristic descriptions', from Menu 1.1 'Reference lists'. When the list of land characteristics appears, *press* **F3** to define a new one. For the code, we want to enter something that reminds us of its connection to the existing discrete LC 'pptMS'; so *enter* the code `pptMS-c`, the 'c' standing for 'continuous'.

After you *confirm* the code, the information form for this new LC appears. You can *enter* anything for the first question ('Name'), for example, `May-September precipitation (continuous)`. The second question, 'Number of classes', is critical: to tell ALES that the LC is continuous, *enter* `0` (zero).

√ A Land Characteristic with '0' (zero) classes is continuous

The final question, 'Units of measurement', is also critical. The units must *exactly match* those of the commensurate LC 'ppt-MS' (you'll see why in the next subsection), so to this question *answer* mm.

Confirm the form, and Menu 1.1.4c 'Specification options for Continuous LC', with three items, should now appear. The *first* menu item allows us to edit the basic information we just entered, except of course for the number of classes (since the LC is continuous, that number must remain 0). *Select* the *second* menu item 'Data validation limits'. A data entry form should appear, in which you can enter the *lowest* and *highest* allowable values for the LC. These values are used as validity checks during data entry. *Enter* '0' for the low value, '1000' for the high; these are the extremes in the evaluation area. *Confirm* the form, and go back from menu 1.1.4c to the LC choice box. The newly-defined LC, 'pptMS-c', now appears in the list.

7.4.5.2 Inferring a discrete LC from a commensurate continuous LC

To be able to use values of a continuous LC in ALES decision procedures, they must be *classified* into values (classes) of a so-called *commensurate* discrete LC. Land characteristics are called 'commensurate' in ALES if they have the *same units of measurement*.

√ Two Land Characteristics are *commensurate* if they have the same scale of measurement

Recall that we set up 'pptMS-c' so it would have the same units as the discrete LC 'pptMS'; now we have to tell ALES that these two are linked. We do this by adding an *inference link* to the *discrete* commensurate LC, so that when ALES looks for a value of the discrete LC and can't find one, it infers the value from the *continuous* commensurate LC by classification.

To add the inference link, *move* the highlight to 'pptMS' in the choice box of LCs, and *press* **F5** to edit its definition. Menu 1.1.4a 'Specification options for discrete LC (with units) should now appear.

Notice that menu option 6 is titled 'infer from commensurate LC'. *Select* this option now. A selection list should appear, showing all the LCs (discrete and continuous) in the evaluation model with the *same unit of measurement* as 'pptMS', i.e. 'mm' (note there is only the one other LC in this set), and another item marked '** don't infer this LC', which you select if you want to remove an inference. *Select* 'pptMS-c' out of the list now; you have now set up the inference link. Menu 1.1.4a is again active.

> Notice that option 5 in Menu 1.1.4a is titled 'Lc -> Lc decision tree'. If you should choose this, you would invoke the decision tree editor to build a decision tree that ALES would use to determine values of this LC (here, 'ppt-MS') from any number of other LCs. The tree looks like a severity level decision tree, except that the decision values are LC classes instead of severity levels of a LQ. This is explained in more detail in Chapter 8; we won't go through an example in this tutorial.

Press **F10** to remove Menu 1.1.4a, *press* **F10** again to remove the LC choice box, and *press* **F10** yet again to remove Menu 1.1; you should now be back at the Main Menu.

Now we must provide a way to enter data for the newly-defined LC into ALES. Recall that data can only be entered for LCs that appear in one or more data entry *templates*. *Select* option 3, 'Data', from menu 1, and then *select* option 7, 'Templates', from menu 1.3. A choice box appears with the two templates that are currently defined. We could define a new template for the new LC, but let's add it to the climatological data instead. *Move* the highlight to 'cl' and *press* **F5** to edit this template. *Select* option 2, 'List of data fields', from menu 1.3.1.

The ordered list of data fields now appears. We can put the new LC anywhere in the field list. For practice, let's put it between the existing second and third fields. *Move* the highlight to item 3, and *press* **F3** to insert another LC *above* this item. *Select* our new LC, 'pptMS-c', from the list of LCs. Now this LC should be inserted as the new item 3 in the ordered list, and the previous item 3, 'pptMS', should have been pushed down to the fourth position in the list. *Confirm* the list and then *go back* from menu 1.3.1 and the list of templates to menu 1.3 'Data'.

Let's now define some new map units for our experiment, so we don't affect the existing database. *Select* menu option 1 'Definitions, From keyboard, Enter or edit', *move* the highlight to map unit 'HnB', and *press* **F6** to initiate a copy. *Name* the copy 'test1', and don't bother to change any other information for it. In the same manner, *copy* 'test1' to 'test2', and then *copy* 'test2' to 'test3'. Now we have three test map units; at present they are all exact copies of map unit 'HnB'. *Remove* the choice box and return to menu 1.3 'Data'.

We now have a way to enter the data, and some test map units to enter data for, so let's enter data for these map units. *Select* menu option 4 'Data, From keyboard, Enter or edit', and then *select* 'cl' from the list of templates. *Select* the first test map unit, 'test1', from the list of homogeneous map units.

Now the data entry form for the 'cl' template appears. Notice that it has four items, of which three have data values and the third (the newly-defined continuous LC) is blank. *Move* to this third item, and *enter* a single value on the continuous scale 0 to 1000 (mm), for example, 500.[60][61]

Move to the fourth item, 'pptMS', and erase the current answer by pressing **F4**. This will force ALES to infer 'pptMS', and it will follow the inference link we set up.

As it computes evaluations, ALES always uses a *directly-entered* (i.e. non-inferred) data value as-entered, and does *not* infer it, even if there is an *inference link* set up for the derived characteristic. The idea here is that the model user must have had more specific information about a map unit if a value was entered.

√ **ALES will *not* infer a LC value if there is a *directly-entered* value in the database.**

However, if ALES is asked to *infer* a data value, with Option 9 'Infer Land Characteristic values' of Menu 1.3 'Data', it *will* infer all values, and over-write any directly-entered values. The idea here is that the model user is directly asking ALES to infer values. See §8.4.5 for details.

Confirm the data entry form, and then *enter* data values for LC 'pptMS-c' for the other two test LCs. For 'test2' enter '200', for 'test3' enter '425'. Also, make sure to *erase* the values for LC 'pptMS'. Notice how the highlight automatically moves to the next map unit in the list when you finish entering values for a map unit. Now *go back* from the list of map units, the list of templates, and menu 1.3, to the Main Menu.

To compute an evaluation for the test map units, in the Main Menu *select* option 4, 'Evaluate', and then in menu 1.4 *select* option 1, 'Compute an evaluation'. *Select* all the LUT's and the three test map units. *Confirm* your choices, and ALES will compute the suitability ratings.

When the computation is complete, in Menu 1.4 *select* option 2, 'View evaluation results'. *Select* 'Land Quality values' for the report type, and then *select* LQ 'm' (moisture availability) as the LQ to report. When the evaluation matrix appears, the cursor should be on the upper left cell, namely for map unit 'test1', LUT 'ccc'. *Press* **F2** to see the first 'Why?' screen. *Press* **F2** again to see 'Why?' a particular LC value; a selection list appears with the LCs that were encountered in the LQ decision tree path for this map unit.

Select 'pptMS' out of this list. Now the 'Why?' screen shows that the value of 'pptMS', i.e. the 400-450mm class, was *inferred* from the *commensurate* continuous LC, namely 'pptMS-c' which

---

[60]Try to enter a value outside this range and see what happens.

[61]The pop-up ('F3') method of data entry doesn't work for continuous LCs, because there is no finite list of choices from which to choose.

we just defined, using the data value of 425mm which we just entered. In other words, ALES *classified* the single value on a continuous scale into one of the discrete classes of the commensurate LC. *Press* **F2** again to see 'Why?' for the continuous LC; as you can see from the screen, this value was entered from the keyboard.

7.4.6 Multi-valued land data

At this point we'll edit this data value in order to illustrate another feature of ALES, namely, its ability to process *multi-valued* land data. This is data which has *more than one value*, each of which has an associated *linear probability*. This mechanism provides an elementary way of handling *probabilistic* or *time series* data.

Notice in the annunciator line for this 'Why?' box that 'F4' is marked 'Edit LC'. We can edit the LC value without leaving the evaluation matrix/'Why?' screen section of the program. *Press* **F4** now to bring up a data entry form for this LC value.

In the present example, let's suppose that we have climate records for map unit 'test1', giving the May-to-September precipitation for each of the last 5 years, as follows: 650mm, 325mm, 500mm, 650mm, 375mm. These average to 500mm, the value we entered previously for 'pptMS-c', and which is now displayed as the default answer in the one-item data entry form. Erase this answer by *pressing* **F4**, and *enter* the *multi-valued* answer as follows:

```
325=.2,500=.2,650=.4,375=.3
```

The format of a multi-valued data item is straightforward: a series of pairs of 'data_value = proportion', separated by commas. So the previous entry can be read as: '20% is 325mm, 20% is 500mm, 40% is 650mm, and 30% is 375mm'.

> The proportions are typically derived by dividing the frequency of a data value in the series by the total length of the series. In the present example, we have one of five years with 325mm, so its frequency is 1/5 = 0.2. Two years have 650mm, so its frequency is 2/5 = 0.4.

Notice that we deliberately made a mistake: the proportions don't sum to unity. *Confirm* the form by pressing **F10**; an error alert should appears, informing you of the error. Remove the alert by *pressing* any key, and then *backspace* over the '3' at the end of the answer, and *replace* it with a '2' (i.e., 20% proportion for 375mm). Confirm the answer again, and you will return to the last 'Why?' screen. Let's see how this multi-value propagates during the computation.

Press **Esc** to remove all the 'Why?' screens at once and return to the evaluation matrix. Now *press* **F3** to force ALES to recompute the evaluation. The annunciator line shows the progress of the computation. After the recomputation is complete, the matrix is re-drawn with any *changed* values highlighted. Notice that all the values of LQ 'm' for map unit 'test1' have changed: instead of being class 2, they now say '1=.6,2=.4'. This is the same syntax that we used to enter the multi-value, and means that it is 60% certain that map unit 'test1' has LQ rating 1 for LQ 'm', and 40% certain that the rating is 2.

The meaning of the proportions is not based on a rigorous definition of probability. We can look at these linear proportions two ways: either that, say, 60% of the land area has rating 1, or that it is 60% certain that all the land has rating 1. For soil or other space-series characteristics, the first interpretation is usually preferable. For climate or other time-series characteristics, the second interpretation is usually correct.

Let's see how the multi-valued LQ was derived. *Press* **F2** to bring up the first 'Why?' screen. Notice that *three* paths were taken in the severity level decision tree at the lowest level: one for May-September precipitation '<400' (p=.4), one for '450-500' (p=.2), and one for '>500' (p=.4). ALES has three possible values for the discrete LC 'pptMS', and so carries out the computation for each of these.

Let's see how these classes were derived. *Press* **F2** again to see 'Why?' for an individual LC, and pick 'pptMS' from the list of LCs. Now we see that 'pptMS' was inferred from the commensurate continuous LC 'pptMS-c' (remember, we set up the inference link) by *classification*. In the present case, two continuous values (625 and 650 mm) classified to the same value (>500mm), so their probabilities were summed. Similarly, the continuous values 350 and 375 both classified to <400mm.

Now let's see how this multi-valued LQ propagates to final suitability. *Press* **Esc** to return to the matrix, and then *press* **F10** to remove the matrix. Now *select* the 'G'ross margin report; a new evaluation matrix appears.

Notice that the predicted gross margins for 'test1' are *not* multi-valued. What happened to the two values for LQ 'm'? *Press* **F2** *three* times to arrive at the 'Why?' screen for the predicted return for corn grain. Notice in the display of the proportional yield decision tree that both LQ values (adequate and moderate stress) were taken into account: both paths were taken in the decision tree, to arrive at a predicted yield of 100% assuming adequate moisture (severity level 1, p=.6) and 80% assuming moderate stress (severity level 2, p=.4). ALES *multiplied* the proportions of yield and land in each LQ class to arrive at a predicted proportional yield of 92%.

The calculation is: (100% * 0.6) + (80% * 0.4) = 60% + 32% = 92% yield.

This *method of linear proportions* assumes that:

(1) there is a *proportionate* amount of *land* with each of the LC values, and

(2) that the land use is applied to *all the land.*

Or, looking at this from the point of view of time series data (such as the rainfall data), this method assumes that:

(1) there are a *proportionate* number of *years* with each of the LC values, and

(2) the land use is carried out *every year.*

Under these assumptions, ALES can combine the individual values of yield to obtain a weighted average.

Now we're done exploring multi-valued land data and continuous-to-discrete inferences. *Press* **F10** enough times to return to the Main Menu.

7.4.7 More on proportional yield

In Tutorial 3, we used *proportional yield decision trees* to compute proportional yields. This is the most general method, and can account for any kind of *interaction* between land qualities. However, there are situations where a *simpler* approach to predicting proportional yield may be appropriate. There are two methods, besides the decision tree, to predict proportional yields, namely, *proportional* (or *multiplicative*) yield factors and *limiting* yield factors. In this section we'll see how to enter these, and how they affect the computation. Section 8.5.2 of the next chapter explains in more detail the situations in which these factors may appropriately be used.

Select Main Menu option 2, 'Land Utilization Types'. From the list of LUT's, *select* the LUT we've been modifying, namely, 'ccc-m', out of the list of LUT's, and then *select* menu option 6, 'Outputs', out of menu 1.2 'Specification options for LUT: "ccc-m"'. *Select* the only output, 'cg'. Now you should be looking at menu 1.2.6 'Specification options for LUT, Output: "ccc-m, cg"'. This is the menu we used in Tutorial 3 to enter the proportional yield tree.

Notice that menu items 3 and 4 are to be used for *Proportional* yield factors and *Limiting* yield factors, respectively. We will see examples of both of these.

> *Proportional* (or *multiplicative*) yield factors are strictly multiplicative, i.e., they are factors between 0 (no yield) and 1 (optimum yield), which *multiply* the proportional yield, when the LQ is limiting.
>
> *Limiting* yield factors are absolutely limiting, i.e., they are factors between 0 (no yield) and 1 (optimum yield), which *limit* the proportional yield, when the LQ is limiting. They are used for LQ's which follow the agricultural 'law of the minimum', i.e. that yield can be no higher than the most limiting factor.
>
> In both cases, the factors are only appropriate for LQ's which act *independently* of each other to reduce yield as the level of limitation increases for each one. If there are *interactions* between LQ's, a *proportional yield decision tree* must be used.

Limiting yield factors

Let's first consider *limiting yield factors*. *Select* menu item 4 to bring up a list of all the LUR's defined for this LUT. Notice that LUR 'c' (temperature regime) is marked with an asterisk '*'; this indicates that there are already limiting yield factors associated with it. These were entered by the builder of the model 'Tutor2', which model we copied at the beginning of the present tutorial. *Select* 'c' from the list.

A data entry form appears, asking for the limiting yield factors for each of the four severity levels (i.e. LQ values) for this LUR. The current values are: 100% (full season), 95% (medium season), 80% (short season), and 0% (won't make grain). These factors correspond to the biological fact that the maximum attainable yield for corn depends on the temperature regime[62]. According to this form, a short-season hybrid in the context of central New York state has an 80% yield potential, relative to a full-season hybrid.

> Also implicit in this form is that there is no interaction between temperature regime and any other factor that may limit yield. If there is an interaction, the LQ would have been included in a proportional yield tree by the builder of 'Tutor2'.

Proportional yield factors

Now let's consider *proportional yield factors*. Go back to menu 1.2.6 by *pressing* **F10** *twice*, and then *select* menu item 3. Again, a list of all the LUR's defined for this LUT appears. Notice that LUR 'e' (erosion hazard) is marked with an asterisk '*'; this indicates that there are already proportional yield factors associated with it. These were entered by the builder of the model 'Tutor2', which we copied. *Select* 'e' from the list.

A data entry form appears, asking for the limiting yield factors for each of the four severity levels (i.e. LQ values) for this LUR. Only level 3, 'moderate' erosion hazard, has a proportional yield factor, so the other levels have an implicit factor of 1, i.e. yield not limited by this factor. This form is stating that any land that is rated in class 3 for LQ 'erosion hazard' will have its proportional yield multiplied by .9, regardless of what that yield is. In this case, the *meaning* of severity level 3 for LQ 'erosion hazard' is that the land user must *put aside* 10% of the land for fallow strips to retard water movement. Hence only 90% of the land is available to yield a crop, so that any proportional yield obtained by the decision tree and/or limiting yield factors should be *multiplied* by 0.9.

Press **F10** twice to return to menu 1.2.6. You could of course edit the currently-defined factors, or enter factors for other land use requirements. For now, let's just see how these factors can affect computed yields. *Press* **F10** enough times to return to the Main Menu.

To see the effect of yield factors, we'll alter the data for two of the map units. *Select* Main Menu option 3, 'Data', and then Menu 1.3 option 4, 'Data, From keyboard, Enter or edit'. *Select* data entry template 'cl' and then *select* map unit 'test3'. In the data entry form, *move* to question 2 ('gdd50') and *change* the answer to '1.8-2k' (it's easiest to use the 'pop-up' method). This low value for heat units will result in a shorter season for corn, which will be reflected as a limiting yield factor. *Confirm* the data entry form, and remove the list of map units.

Now *select* data entry template 'ss' and then *select* map unit 'test2'. In the data entry form, *move* to question 3 ('slope') and *change* the answer to 'C'. This value, when combined with the other LCs such as silt loam surface soil, will result in a moderately severe erosion hazard, which will be reflected as a proportional yield factor. *Confirm* the data entry form, and *press* **F10** enough times to return to the Main Menu.

Now let's compute evaluation results for these map units. Select Main Menu option 4, 'Evaluations', and then *select* Menu 1.4 option 1, 'Compute an evaluation'. *Select* LUT 'ccc-m',

---

[62]Because as the number of heat units increases, a longer-season corn hybrid can be grown, and the longer-season hybrids have higher yield potentials

and then *select* the test map units for which we've changed the data, namely 'test2' and 'test3'. *Confirm* your choices, and ALES will proceed to compute the suitability ratings.

When the computation is complete, Menu 1.4 option 2, 'View evaluation results'. Select the 'Y'ields report, and then *select* the output 'cg' (corn grain). The evaluation matrix will appear, showing the predicted yields for the two test map units for corn grain.

The cursor should be on the upper left cell, namely for map unit 'test2', LUT 'ccc-m'. *Press* **F2** to see 'Why?' the predicted yield is 97.2 bushels per acre. The proportional yield is 72%. Notice, however, that the proportional yield tree predicts 80%, based on planting date and moisture availability. However, there is one *proportional yield factor*, namely, 90% because of a moderate erosion hazard on this land. This means that the farmer will have to set aside 10% of the land for fallow strips. Hence the predicted proportion of 80% must be multiplied by 90% (i.e. the amount of land that will actually produce a crop), resulting in a 72% proportional yield.

Now let's see how a limiting yield factor works. *Press* **F10** to go back to the matrix, and *move* the cursor to the 'test3' map unit. *Press* **F2** to see 'Why?' the predicted yield is 97.2 bushels per acre. The proportional yield is 80%. Notice, however, that the proportional yield tree predicts 100%, based on planting date and moisture availability, neither of which are limiting. However, there is one *limiting yield factor*, namely, 80% because of a short season on this land. (You will have to press the down arrow or the 'PgDn' key to see this line in the 'Why?' screen.) This means that the farmer will have to use a short-season corn hybrid with a yield potential of 80% of a full-season hybrid. Hence the predicted proportion of 100% must be *limited* to 80% proportional yield.

All three ways of determining yield, namely proportional yield trees, proportional yield factors, and limiting yield factors, can be used in the same model, in any combination. *Press* **F1** to see the help screen for this 'Why?' screen; this explains how all three methods are combined to determine a predicted proportional yield. §8.5.2 explains this in detail.

When you are done viewing the results, *press* **F10** enough times to return to the Main Menu.

7.4.8 More on physical suitability

In this section we'll explore several advanced features of ALES for evaluating the *physical suitability* of land.

7.4.8.1 The maximum limitation method

In Tutorial 3 (§7.3.6.6) we built a decision tree to determine physical suitability subclasses of land units. There is another way to assign physical suitability subclasses, namely the *maximum limitation method.* This method has the advantage that it is very quick, because you don't need to build a decision tree. However, it depends on the various land qualities having *commensurate* severity levels, i.e., for each severity level (1,2, etc.) the degree of limitation of each land quality must be similar. This is covered more fully in §8.5.1. For now, let's change LUT 'ccc-m' to use this method, and compare it with the results of the decision tree.

Select Main Menu option 2, 'Land Utilization Types'. From the list of LUT's, *select* the LUT we've been modifying, namely, 'ccc-m', out of the list of LUT's, and then *select* menu 1.2 option

8, 'Physical suitability subclasses'. Now you will see menu 1.2.8, with three options; *select* the second one, 'Choose factors for maximum limitation'.

Now you will be presented with a list of the LUR's defined for this LUT; you can select any, none, or all of these to be considered. Select *all* of them by *pressing* **F6** ('All'), and then *pressing* **F10** to confirm your selection.

To see the effects of this change, we'll have to recompute the evaluation with this new decision procedure. *Back up* to the Main Menu, and then *select* option 4, 'Evaluations', and then *select* Menu 1.4 option 1, 'Compute an evaluation'. *Select* the LUT's 'ccc' and 'ccc-m', and then *select* map unit 'A3' (the first association in the list). *Confirm* your choices, and ALES will compute the suitability ratings.

When the computation is complete, *select* Menu 1.4 option 2, 'View evaluation results'. Select the 'P'hysical suitability report. The evaluation matrix will appear, showing the physical suitabilities. You can see a big difference between the two methods. The left-hand column, for 'ccc', only has two homogeneous map units with physical suitability other than 1, since in this LUT the physical suitability subclasses are only being used if the map unit is completely unsuitable. The right-hand column, for 'ccc-m', is different: every map unit has some limitations, resulting in a non-optimum physical suitability subclass (i.e., class number > 1).

Move the cursor to row 'CeB', column 'ccc-m'; notice that the rating is '2m/nr/pl'. Let's see what this means. *Press* **F2** to bring up the 'Why?' screen. You can see that this map unit has land quality rating 2, generally corresponding to slight limitations, for LQ's 'm' (moisture availability), 'nr' (nutrients). and 'pl' (planting conditions); thus the rating '2m/nr/pl' indicates that the map unit has slight limitations for these three factors.

You may have noticed that this map unit also has land quality rating 2 for 'e' (erosion). Why isn't this in the subclass code? *Move* to the bottom of the 'Why?' screen, and you will see that LQ 'e' was already taken into account in the physical suitability decision tree. Although it is rated as severity level 2, the result of the decision tree is 1. This shows that if there is both a decision tree and a list of limiting factors, the decision tree takes precedence. This is because the decision tree is more detailed, and can take interactions into account.

> **√ LQ's encountered during the traversal of the decision tree are *not* taken into account as maximally limiting factors.**

Let's fix this problem, by removing the decision tree so that the factors that were considered in the decision tree ('e', 'f', and 'me') are instead considered by the maximum limitation method.

We could backtrack all the way to the Main Menu in order to edit the LUT, but there is a *shortcut.* Notice in the annunciator (bottom) line of the screen that 'F5' is marked 'Edit LUT'. *Press* **F5** to edit the current LUT; since the highlighted cell of the matrix is in the column for 'ccc-m', this LUT will be edited. *Select* Menu 1.2 option 8 ('Physical suitability subclasses'), and then *select* Menu 1.2.8 option 1, 'enter/edit physical suitability subclass decision tree'. You will see the decision tree editor, with the first level of the physical suitability tree displayed. *Press* **F8** ('cutT'); the tree should disappear from the screen. *Press* **F10** to leave the editor, and then *press* **F10** again enough times to return to the evaluation matrix display. Now recompute the evaluation by *pressing* **F3**. When ALES finishes computing, you should see some changed values. For example 'CeB' now has the rating '2e/m/nr/pl', showing that the erosion hazard LQ, 'e', is now considered as a maximum limitation.

This is a good time to show something else about the maximum limitation method. *Move* the cursor to row 'HsD', column 'ccc-m'; note that the physical suitability subclass is '4e'. *Press* the **F2** key to see the 'Why?' screen. Notice in the list of LQ's and their ratings that LQ 'e' is the only one with a severity level of 4, but that five others ('c', 'lr', 'm', 'me', and 'nr') have a non-optimum rating, in this case 2. However they are *not* shown in the physical suitability subclass name, because the rating of 4 for 'e' is more severe, i.e., it is a *more severe* limitation. When you're done with this screen, press **F10** to return to the evaluation screen, and move the cursor back to row 'CeB'.

There is still a problem with the rating '2e/m/nr/pl': two of the LQ's used in this model are not really related to physical suitability, but were included in the model only to allow added costs or reduced yields to be expressed. These 'economic' LQ's are 'lr' (lime requirement) and 'nr' (nutrient requirement). So, let's remove them from consideration by the maximum limitation method.

From the evaluation matrix, *move* the highlight to any cell in the column for LUT 'ccc-m', and *press* **F5** to edit the LUT. *Select* Menu 1.2 option 8 ('Physical suitability subclasses'). Then *select* Menu 1.2.8 option 2 to edit the list of LQ's to be considered in the maximum limitation method. The list of LQ's will appear, showing (by the '♦' symbols to their left) that all LQ's are currently being considered. *Move* the cursor to 'lr' and *press* **F4** to de-select this LQ; the '♦' symbol next to 'lr' should disappear, indicating that 'lr' is no longer to be considered. *Do* the same for 'nr'. *Press* **F10** three times to return to the evaluation matrix display. Now recompute the evaluation by *pressing* **F3**; you can see the changed values. For example 'CeB' now has the rating '2e/m/pl'; the nutrient requirement is no longer considered as a physical suitability factor.

7.4.8.2 Subclass names and annotations

So far we have only used physical suitability subclass codes like '4e'. These probably are meaningful to the model *builder*, but you may want to give descriptive names to subclasses, to make them more comprehensible to model *users*. Also, you may want to add notes to certain subclasses, for example to give management recommendations. In this section, you will learn how to do this.

Again from the evaluation matrix, *move* the highlight somewhere in the column for LUT 'ccc-m', *press* **F5** to edit the LUT, and *select* Menu 1.2 option 8 ('Physical suitability subclasses'). Then select Menu 1.2.8 option 3, 'Enter/edit names or notes for physical suitability subclasses'. A list will appear, with the dummy item '<add a new item>'; *press* **F3** to add a new subclass. An entry box will appear, asking you for the physical suitability subclass code for which you want to enter a name and/or a note. *Type* `4e`, and *press* **Return** to confirm it. A one-item data entry form will appear, asking for the descriptive name for class '4e'. This is optional; any name you type here will be displayed in 'Why?' screens and reports whenever the '4e' class is displayed. *Type* `non-arable` and press **F10** to confirm it. Now the list of subclasses has one item, namely '4e', with the descriptive name 'non-arable'.

Now let's add a note to this subclass. The cursor is already on this item in the list, so *press* **Alt-F9** to bring up the note editor. *Type* in an appropriate note (you might say something about the land being too steep to farm mechanically) and *press* **F10** to save it. Now you should see the list of subclasses again; note that '4e' has the '°' symbol to its left, indicating that there is a note for this item. *Press* **F10** to remove the list and return to the evaluation matrix.

Let's see under what circumstances the PSC name and note are displayed. *Move* the cursor to row 'HsD' (you should still be in column 'ccc-m'). Notice that the physical suitability subclass for this map unit is '4e', the one we just named and annotated. *Press* **F2** to display the 'Why?' screen. Notice that in the line which displays the subclass, that the descriptive name 'non-arable' is shown after the subclass code. Also notice that this line is marked with the '°' symbol, indicating the presence of a note. *Press* **F9** and *select* the second note from the list; you will see the note you just entered.

When you're done, *press* **F10** enough times to return to the choice box of evaluation models. You might want to *save* your model to diskette (with 'F7').

Congratulations, you've finished Tutorial 4! You should now be able to use a wide variety of techniques to build ALES models. Refer to Chapter 8 for a full discussion of the various model-building concepts.

7.4.9 How to delete evaluation models

In Tutorial 3 you learned how to make a backup copy of an evaluation model (knowledge base and database). Eventually you will want to *delete* evaluation models. For example, once you are done with the tutorials, there is no need to keep them on-line, taking up space on the hard disk[63].

Evaluation models can be *deleted* like any other entity: in the choice box where they were defined (i.e. the one you're looking at now), press 'F4'. If you do this, you will be asked if you really want to delete the evaluation; the default answer is 'No'. If you want to delete it, press 'y' to move the highlight to 'Yes' and then confirm your choice. The model and all its data will be *irrevocably* (unrecoverably) deleted from the knowledge and data bases. Typically you would have made a backup copy on diskette (using the 'F7' method presented in Tutorial 3) prior to deleting a model.

7.4.10 A final word

Even with the four tutorials, we haven't been able to cover every detail of the ALES program. Now it is up to you to *experiment* as much as you want with ALES. Remember that you can always get on-screen *help* by pressing the 'F1' or 'F2' key. Also remember that ALES always asks you for *confirmation* when you are about to perform major surgery on your model or database. Finally, remember that ALES keeps all models (i.e. entities on the opening list of evaluation models) completely *separate*, so that if you create or copy a model for experimentation, you can't inadvertently delete or modify someone else's work.

---

[63]Actually, just deleting a model from the list does not free up any DOS disk space, it only frees space in the MUMPS dataset. You can recover the space for DOS using the dataset compression procedure explained in §11.2.

Experiment, build some models, and we sincerely hope that you will find ALES to be a useful tool in your land evaluation activities.

8 Building ALES models

In this chapter we present *design strategies* for the model builder. By far the most difficult aspect of using ALES is building good models. There are several reasons for this: (1) nature is complex, (2) data sources are limited, and (3) the FAO framework, as implemented within and extended in ALES, imposes a certain structure on the model builder.

Tutorials 3 and 4 of the previous chapter are an introduction to the model building process. However it was not possible in the context of a tutorial to completely describe all the decisions and judgments that go into successful model building. In the present chapter, we present the outline of the model building process, stressing the design decisions that the model builder must make.

> The term *model* as used in ALES refers to a set of *decision procedures*, not to a *process model* (e.g. of plant growth as related to soil conditions) or a *empirical model* (e.g. regression equations relating land use to land characteristics). The term 'model' still is appropriate, however. The ALES model is a representation of the judgment of the land use expert, which in turn is a mental model of reality. Indeed, process or empirical models can be used to help build ALES decision procedures.

8.1 Outline of the model building process

ALES models are built *incrementally* and *interactively*. It is important to have a sound design; however, it is not necessary that all particulars of an ALES model be determined before beginning to build it. In particular, verification should take place on a preliminary model before too much effort is expended on extensive data entry and evaluation. Computer programmers have a well-known folk expression: "Build one (program) to throw it away". While a bit of an exaggeration, it is certainly true that the exercise of building the first model will reveal many pitfalls which can be avoided in a second try. This section describes a typical sequence of actions.

First, the model builder should build a *preliminary* version of a land evaluation model. This involves the following steps:

(1) selecting a few representative land utilization types
(2) expressing these in terms of their most important land use requirements
(3) determining which land characteristics are available to form the basis of evaluation
(4) constructing decision trees to relate land characteristics to land use requirements
(5) collecting economic parameters, such as prices
(6) selecting some representative or well understood map units
(7) collecting and entering land characteristic data for these map units.

Design of the preliminary model requires close cooperation with the many specialists whose knowledge is necessary for successful land evaluation. The model builder will serve as the collator and arbitrator of contributions from agronomists, soils specialists, hydrologists, climatologists, economists, etc. The design must be based on the project objectives and available data sources.

Once a preliminary model is built, and data entered on a representative set of map units, an *evaluation matrix* may be computed. This matrix shows the rating of each map unit for each land utilization type, either as (1) physical suitability subclasses, (2) predicted present value of cash flows in, out, and net, (3) benefit/cost ratio of the discounted cash flows, (4) Internal Rate of Return of the discounted cash flows; (5) predicted cash flows in, out, and net (i.e. gross margin), (6) economic suitability classes based either on discounted cash flows or gross margin,(7) expected yields of crops or other outputs, or (8) ratings for single land qualities.

The predicted ratings should be compared against values computed by other methods, and rankings of map units may be reviewed. If there are actual yield data, they may be compared with expected yields.

Viewing this matrix should cause the evaluator to ask many questions. Some ratings will appear unreasonable, either in absolute terms (e.g. the model predicts total crop failure, but the map unit in question does provide a reliable, low yield), or in relative terms (e.g. two map units are predicted to have very different gross margins for a particular LUT, but farms on the two areas are not noticeably different in their economic status). ALES provides a way to query the model and data to determine the source of any discrepancies and correct them, thus allowing for iterative model building.

While viewing the matrix, the evaluator may repeatedly press a function key (called the '*Why?*' key) to follow a backward chain showing every step of the reasoning by which a particular map unit was assigned a particular suitability. In this way, the model builder will be able to identify those components of the model that are responsible for unexpected or incorrect results.

At each 'Why?' screen, all data, economic parameters, and decision trees that entered into the current step of the computation are available to be edited. Thus the model builder can correct any problems immediately upon discovering them. The evaluation matrix may be recomputed directly from the matrix display; any changed results will be highlighted, so that the model builder can see which results were affected by the changes to the model.

Typically, this step will be repeated many times, until the model builder is satisfied that the model successfully predicts suitabilities, to the desired accuracy.

Once a satisfactory preliminary model is completed, the evaluator can then *extend* it to a wider set of land utilization types. The program allows the copying of entire land utilization type definitions. The copies can be modified to reflect differences in, for example, management method, or economic assumptions, resulting in side-by-side comparisons of related alternatives in the evaluation matrix.

Then the model will typically be turned over to model users, who will enter descriptions and data for all the map units in the survey area, and compute evaluations for use by the output consumers.

8.2 Physical vs. economic evaluations

A major choice which must be made by the model builder is whether to evaluate the land in *physical* terms only, or in both *physical and economic* terms. A physical evaluation results in *relative* suitabilities of each land area for the set of land utilization types in the evaluation, based on the evaluator's judgment. *Within a single land utilization type*, this may be very useful.

However, when *comparing* land utilization types, a physical evaluation has no *common basis*. If we are trying to determine the *best use* for a land area, we need some way to compare suitabilities of different uses. It may well be that land that is rated 'highly suitable' in physical terms for a given land utilization type, but only 'marginally suitable' for another, may preferentially be used for the second LUT by farmers. This is because the farmer is in some sense trying to optimize land use according to some criterion other than crop suitability. Typically, the optimization criterion is largely based on economics, although social and cultural factors may certainly play a large role.

As in many other decision making processes, *money* , or something of value that could in principle be converted to money, is the common denominator. The common saying that one cannot "compare apples and oranges" is not true; they can be compared on the basis of what people are willing to pay for them. While it is true that at times this simplistic view of the common basis of currency can be carried too far, it has a certain validity.

Several objections may be raised to economic evaluations:

(1) "Economic evaluations are unstable over time". This is certainly true. However, ALES makes it extremely easy to recompute an entire evaluation with changed economic parameters, such as prices of inputs and outputs, and interest rates. An economic evaluation implemented in ALES can be completely revised, in response to changed economics, in a matter of minutes.

(2) "Economic parameters are difficult to obtain". This is true in general. However, decision makers, including individual farmers and planners, base much of their decision on economics, and they must obtain best estimates of things like expected prices. So it does not seem unreasonable to ask the evaluator to determine these. In also seems quite desirable to force natural resource specialists to work with agricultural economists.

(3) "People don't base their decisions on economics alone". Again this is very true. However, without adequate economic return, a land use is patently impossible. There are two kinds of distortions to pure economic decisions: cultural and societal. A cultural bias is best handled by omitting the land utilization type in question from consideration. For example, if in a certain area it is considered beneath the dignity of the population to engage in a certain activity, no (??) amount of projected cash return will induce them to do so; therefore the activity should not be considered as an alternative. A societal bias will typically be reflected in economics: the society as a whole (via the government) will subsidize selling prices or input prices, or give tax breaks to encourage an activity. Thus societal imperatives can be incorporated into an economic analysis.

(4) "Economic analysis requires predicting yields, and we're not good at that". True again, yet if we cannot predict yields in some useful form (e.g. average, worst case, 1 year in 5 worst case...), the concept even of physical suitability classes is in question. A difficult problem must be attacked for natural resources information to be useful.

In some situations, a strictly physical evaluation may be sufficient. An example is if land is to be rated for several uses, but the uses are not to be compared with each other. For example, the interpretive tables in a soil survey report for non-farm uses, such as buildings, pipelines, sewage disposal, etc. could be constructed as physical evaluations, with each use represented as a completely independent land utilization type.

8.2.1 Physical evaluations

In a *physical* evaluation, map units are assigned *physical suitability classes*, which indicate a relative suitability, from 1 (best) to some maximum number specified by the model builder. For example, in the FAO framework, there are *four* physical suitability classes: 's1', 's2', 's3/n1', and 'n2'. The boundary between 's3' and 'n1' is defined only in economic terms, hence in a physical evaluation, these two are grouped.

> The default of four classes is not arbitrary. It seems to correspond well to our psychological tendency to value things in everyday life as 'excellent', 'good', 'fair to poor' and 'terrible', or, in land evaluation terms 'highly suitable', 'suitable', 'marginal', and 'not suitable'. With fewer classes we are not satisfied and always seem to want to split a class; with more classes we have trouble deciding between classes.

ALES allows the evaluator more flexibility: the FAO-recommended four classes can be used, but if a lower or higher number seems appropriate, the model builder can define the LUT with that number. For example, if it is only necessary to separate *unsuitable* and *suitable* (i.e. FAO orders 's' and 'n'), a two-class LUT may be defined[64]. This is especially recommended if an economic analysis is to be performed. On the other hand, if the evaluator feels that more classes can be reliably distinguished, more than four classes may be specified.

A physical evaluation is often used for *risk assessment* (e.g. of environmental degradation), even if an economic evaluation is also to be performed to determine optimum land use. It can also be useful to *group* map units according to their limitations and other land qualities that determine *management strategies.*

8.2.2 Economic evaluations

ALES always attempts to compute an economic evaluation following the computation of a physical evaluation. If components of the economic model (e.g. prices, optimum yields, proportional yield information) are missing, ALES will not be able to complete the computation.

Computation of the physical suitability precedes computation of the economic suitability, so that *land that is rated as physically unsuitable will not be considered for the land use*, no matter what the economics. ALES will not perform an economic analysis on land in the *highest-numbered* physical suitability class, which is always assumed to correspond to the FAO class 'n2', i.e. *physically unsuitable* for the use.

If an economic model is constructed, degrees of suitability will be revealed in the economic analysis. It then may suffice to define two physical suitability classes (ALES classes 1 and 2), corresponding to FAO orders 's' and 'n', and build a simple physical model whose only purpose is to place physically unsuitable land in order 'n', i.e. ALES physical suitability class 2.

ALES performs two kinds of economic evaluations: (1) *discounted cash flow analysis* and (2) *gross margin analysis*. The first of these considers the time value of money, and is appropriate for any plan in which cash flows in and out are spread out over a number of years. It is especially

---

[64]This is the method used in Tutorial 3.

necessary if one-time costs, e.g. capital works such as permanent land improvements, are incurred, and the benefits of the one-time cost are then received over a number of following years. Gross margin analysis, on the other hand, is satisfactory for analyzing LUTs with no capital improvements and only recurring costs and outputs.

The measures of economic suitability in discounted cash flow analysis are (1) the *predicted net present value* (NPV), (2) the *benefit/cost ratio* of the present value of cash in vs. cash out, and (3) the *Internal Rate of Return* for each land utilization type – land unit pair. The NPV is measured on a *continuous* scale, as *currency per unit area – plan*, where 'plan' refers to the *planning horizon*, i.e. length of the project. The benefit/cost ratio is dimensionless, and reflects the leverage of investing a unit sum of money into the project.

The measure of economic suitability in gross margin analysis is the *predicted gross margin* for each land utilization type – land unit pair. This is measured on a *continuous* scale, as *currency per unit area – year*, i.e. the cash flows over the length of the project are normalized to a per-year basis.

8.2.3 Economic suitability classes

The NPV and gross margin are expressed on a *continuous* scale, i.e. as currency. For compatibility with the FAO framework, ALES is able to group these continuous values into four *discrete* suitability classes, corresponding to FAO classes 's1', 's2', 's3', and 'n1'. (Class 'n2' is reserved for land that was previously rated as physically unsuitable.) To allow ALES to perform this grouping, the evaluator must assign three *economic suitability class limits*, i.e. values of currency per unit area – plan (for discounted cash flow analysis) or currency per unit area – year (for gross margin analysis) which divide 's1' from 's2', 's2' from 's3', and 's3' from 'n1'. If economic class limits for a kind of economic analysis are not specified by the model builder, ALES will not be able to display or print FAO economic suitability classes for that kind of analysis.

The economic class limits are assigned separately for each land utilization type, and need not be the same. This is because the economic suitabilities are expressed per unit area, and some land uses may be much more extensive than others, and thus require less present value or gross margin, per area, to be equally desirable. An example is small-scale vegetable gardening vs. pasture: the pastoralist is satisfied with a much lower return per unit area than the gardener. The model builder must consider these factors when setting economic class limits.

The limit between 's1' and 'n1' should be set at the gross margin below which the land user will elect not to implement the land utilization type. Typically this limit will be zero (i.e. any positive margin should motivate the farmer), but if the farmer's labor has not been considered as an input to the LUT, the value of the labor may be considered here, since the farmer requires a certain income to be able to live. The higher two limits should be set to differentiate the best land from moderately good land, and moderately good land from marginal land. Social factors can dictate these limits: it may be considered that good land must return a certain amount to allow for a acceptable level of prosperity.

8.3 Defining Land Utilization Types

By far the most significant decision in constructing a model is the selection and definition of land utilization types, i.e. the land use alternatives. This is covered in detail in FAO Soils Bulletins 52 and 55. Suffice it here to say that all aspects of the land utilization type, especially including the socio-economic and technical *context*, must be completely specified. Level of mechanization, access to credit, tenancy system, and many more factors greatly influence the variables that must be considered when constructing ALES models.

The following sections explain design considerations for each aspect of the land utilization type specification.

8.3.1 Discount rate

Each land utilization type has a separate *discount rate,* i.e., the rate at which future cash flows are discounted in order to determine *present value*. This rate is used to discount a cash amount as follows:

where the *DiscountRate* is the rate at which money may be borrowed, expressed as a percent, and *YearsFromPresent* is the number of years from the present when the *CashAmount* will be spent or received. From this we can see that the discount rate can have a very large effect on the present value of a project, especially if a significant portion of either the costs or benefits of the project are incurred or received in the future. For example, a forestry project may incur many of its costs in the first year, but not receive any benefits for 30 years.

The discount rate is equivalent to the *interest rate,* i.e. the rate at which money which is borrowed must be carried, as a percentage per annum, in gross margin analysis.

The discount rate can be set to reflect current or anticipated future market rates, or a subsidized ('social') rate, depending on the purposes of the evaluation. If you want to compare two rates for the same LUT, you can copy the LUT to another, alter the discount rate, compute an evaluation with the two LUTs, and see the comparison side-by-side in the evaluation matrix.

8.3.2 Length of Planning Horizon

Each land utilization type has a separate *length of planning horizon*, i.e., the number of years over which costs are incurred and/or outputs are produced.

In *discounted cash flow analysis*, the planning horizon sets a limit on how far in the future cash flows can be specified. In addition, two net present values (NPV) can only be compared for equal planning horizons.

In *gross margin analysis*, the length of the planning horizon is used to *annualize* expected yields, so that different LUT's can be compared on a common scale.

For *annual* cropping systems with only recurrent costs, the planning horizon is one year. For a *rotational* cropping system with only recurrent costs, the planning horizon is the length of the rotation. This is also appropriate for a *crop-fallow* system. For *perennial* crops, the planning horizon should be set to the expected life of the perennial. For example, a banana plantation may have an expected life of 10 years, after which it will have to be renovated or abandoned.

For any LUT with significant *capital improvements* either at the beginning of the plan or at some specific time within it, the planning horizon should be set long enough to include the years with improvements, and also the years over which the benefits of the improvement will be reaped, i.e. the so-called 'useful lifetime' of the improvement. As a practical matter, depending on the discount rate, 10 (high discount rate) to 30 (low discount rate) years is usually the maximum planning horizon, because cash flows that occur after a time so far in the future have very little effect on the net *present* value of the plan. Once the present value of a cash flow drops below about 10% of the cash amount, it can usually be disregarded, since the present value of earlier cash flows overwhelm it in the calculation of NPV. You can do some calculations with the discounting formula in the previous section, to get some feel for how far in the future it is worthwhile to plan.

8.3.3 Inputs to a Land Utilization Type (annual and one-time)

The *inputs* to a land utilization type are the things that are required to implement it, *regardless of the land on which it is implemented.* If different levels of inputs are required, depending on conditions of the land, they are included as additional costs tied to a severity level of a specific land use requirement.

Inputs are of two kinds: *annual*, i.e. needed each year, and *one-time*, i.e. needed only once, at some specific year during the length of the LUT. The same input can of course be included in both lists; for example, an initial liming may be required to bring land into production, with a supplemental application each year. The same 'one-time' inputs may be specified at more than one specific year.

Examples of inputs are labor, seed or plants, fertilizers, fuel, equipment and tools, and construction materials. Inputs are first defined in the *reference list of inputs* for the entire evaluation. Then they are available for inclusion as inputs of any or all land utilization types in the evaluation. Each LUT can have any number of annual and one-time inputs. Each input in the reference list must be assigned a unit of measure and a monetary value.

When deciding on the unit of measure for an input, remember that it must be expressible in a *per-unit area* basis. Some units of measure are: kilograms, liters, days, or units. Units of concentration are not usable, e.g. parts per million, since they already have the area as part of their denominator.

8.3.4 Outputs

The *outputs* of a land utilization type are the things of value that are produced by it. Commonly, these are the usable parts of crops, e.g. grain, straw, tubers, etc. However, there are other kinds of 'outputs', such as pollutants, soil that is lost to erosion, recreational value, and so forth. As long as a unit of measures can be devised for an output, and a monetary value (positive or negative) can be assigned to each unit of measure, ALES can include the output in its analyses. Output values can also be negative; this is appropriate for pollutants.

When deciding on the *unit of measure*, remember that it must be expressible in a per-unit area basis when yields are discussed. Some units of measure are: kilograms, quintals, pasture days, total digestible nutrients, tons of soil, liters of pollutant. Units of concentration are not usable, e.g. parts per million, since they already have the area as part of their denominator.

Outputs, along with their units of measure, are first defined in the *reference list of outputs* for the entire evaluation. Then they are available for inclusion as outputs of any or all land utilization types in the evaluation. Each LUT can have any number of outputs.

8.3.4.1 Harvest years

For each output of a land utilization type, the *years in the plan when the output is harvested* must be specified. An output may be harvested not at all, once, or several times within any given year. From this information ALES is able to determine the present value of the output over the extent of the plan. Thus ALES can evaluate multiple crops, intercrops, perennial crops etc.

For example, consider a six-year planning horizon, corresponding to the following rotation,: first year: establish alfalfa with an oat nurse crop, harvest the oat grain and straw, and take one cutting of alfalfa; second and third years: take three cuttings of alfalfa; fourth year: take three cuttings of alfalfa and then plow it down with a cover crop of winter rye; fifth year: harvest the rye as green chop, plow it down, and plant corn, harvest corn grain; sixth year: plant and harvest corn grain. In this situation we have the following outputs and years when harvested: oat grain (1), oat straw (1), alfalfa (1, 2, 2, 2, 3, 3, 3, 4, 4, 4), rye green chop (5), corn grain (5, 6).

For perennial crops there are typically fewer harvests than the plan length. For example, a forest stand that is expected to last 30 years (i.e. this is the length of planning horizon) may have a thinning crop at 5, 10, and 15 years, and a timber crop at 30 years, so that we have the following outputs and years when harvested: thinnings for wood chips (5, 10, 15), trees for lumber (30).

Perennials and annuals can be mixed in a LUT (e.g. for agroforestry); the length of the planning horizon should be set by the life of the perennial stand.

8.3.4.2 Optimum yield

For each output of a land utilization type, the *optimum yield* must be specified. This is the yield, per unit area per crop (as defined in the previous section), that would be *expected* in the context of the land utilization type (which includes management, input levels, socio-economic conditions, etc.) assuming *no limitations*, i.e. all land qualities that affect yield with the lowest level of limitation. The optimum yield is not meant to be a biological maximum, but rather a realistically attainable yield in the context of the LUT.

The lowest level of limitation of each land quality, and hence the optimum yield, depends on the context of the evaluation. For example, if an evaluation model has a 'radiation regime' land use requirement, the lowest level of limitation for this is generally set to correspond with the optimum radiation regime expected within the *area of applicability* of the evaluation. This is reflected in the severity level decision tree for this land quality: severity level 1 is obtained with a certain combination of values of land characteristics (e.g. cloud cover, latitude, elevation, ...) which defines 'optimum'. This illustrates the importance of knowing, and stating in the model, the area of applicability of a model.

> Depending on the purpose of the evaluation exercise, the 'area of applicability' may be larger than the area to be evaluated. For example, to compare yields on a national or continental scale, the 'optimum' yield could be set to the expected optimum in the country as a whole, even though only a section of the country is being evaluated.

The level at which the optimum yield is set has a great impact on economic evaluations. It should be set according to the purposes of the evaluation. For example, if the land utilization type is to be implemented by *risk-averse* farmers, it should be set conservatively, e.g., the lowest expected yield over a ten-year period, or an average yield less 20%. In this way the optimum yield will be attained on the best land in the evaluation area except in unusually bad years. On the other hand, if the farmers have sufficient reserves to withstand unfavorable years, an average yield would be an appropriate level for the optimum yield.

In some situations the same crop may have different yields at different harvests within a rotation. An example would be early, midsummer, and late summer cuttings of a forage. It would not be correct to specify three crops per year, each with the same optimum yield. To solve this problem, the same crop should be listed several times in the reference list of outputs, and each of these included in the LUT specification. Then each can get a separate number of crops and optimum yield. In the forage example, we could define three outputs: 'alfalfa-e', 'alfalfa-m', and 'alfalfa-l'; there would be one crop per year of each, having a different optimum yield depending on the time of cutting.

8.3.4.3 Production-related inputs

Some production costs are directly related to the *level of production* of an output. For example, a portion of harvesting costs depends on how much must be harvested. ALES allows the model builder to specify (using Menu 1.1.5.6, Option 5) these *production-related* costs, by means of a list of *inputs*, with each the amount of each input specified *per unit of output*. The inputs are selected from the same reference list of inputs mentioned in §8.3.3.

8.3.5 Defining land use requirements for a LUT

The set of *land use requirements* are the main defining factors of land utilization types. The LURs are the *independent variables* of the land evaluation model. They are of critical importance in ALES because they are used to (1) define *physical suitability*; (2) *increase costs* of production; and (3) *reduce yields* of outputs.

In the FAO framework, Land Utilization Types have one or more *requirements*, which are matched with corresponding *land qualities*. These are just two sides of the same coin: the *requirements* correspond to the *demand* of a land use, whereas the *qualities* correspond to the *supply* by the land of that factor. When you are *building models* of land uses in ALES, you specify the *requirements*. When ALES *computes suitabilities* for each land use it will compute the corresponding *qualities*.

The FAO has published lists of *suggested* land use requirements for rainfed agriculture (FAO 1983), for irrigated agriculture (FAO 1985) and for forestry (FAO 1984), along with suggestions on how they might be evaluated, e.g., which land characteristics are important for these requirements. The land use requirements can be grouped into categories:

(1) ***Agro-ecological***: These affect the growth of the crop or animal. For example, temperature, radiation, water, oxygen, nutrients

(2) ***Management***: These affect how the land user must manage the land. For example, conditions for land preparation and harvest, irrigation frequency.

(3) ***Land Improvements***: These are requirements that must be met in order to bring the LUT into production. For example, land clearing, irrigation and drainage, requirement for leaching of salts.

(4) ***Environmental risks***: These are requirements that the land *not* degrade. For example, erosion hazard, risk of salinization, chemical contamination in runoff or drainage water.

(5) ***Geographical***: These are requirements of location. For example, accessibility to or within the farm, distance from protected areas or to markets.

(6) ***Socio-economic***: These are requirements that the land user must meet. For example, attitudes towards a certain type of land use.

Although these lists are useful starting points, they are not exhaustive nor exclusive. In fact, a more direct way of deciding on a set of LUR's is often quite successful: soliciting answers from the land use experts to the question: "What qualities of the land are important for the successful implementation of this LUT?" A related question is: "What qualities of the land *differentiate* lands that are very suitable, suitable, marginal or not suitable for this land use?"

It is *not* expected that every evaluation will use every requirement, in fact, in order to keep models to a reasonable size, it is important to *limit* the set of land use requirements. A land use requirement should be included in the model if:

(1) it is *important* for the land use;

(2) there exist *critical values* of the corresponding land quality in the study area, i.e., values which cause *differences* in the performance of the set of land units being considered;

> Even factors that are very important to a land use should be omitted from a model's set of land use requirements, if they do not *vary* over the set of land units under consideration. In other words, land use requirements whose corresponding qualities are *constant*, within the *context* of the survey area, should be left out of the model.

(3) it is possible to *predict* values of the land quality, i.e., there is sufficient expert knowledge about the relationship between this land quality and a set of land characteristics;

> In ALES terms, this means that enough is known about the requirement so that it is possible to construct a *decision tree* for it.

(4) it is possible to obtain data values for the *land characteristics* that are used in the prediction.

Here are some *hypothetical* examples of land use requirements that would *not* be included in a LUT, for each of the above reasons:

(1) The LUR 'Erosion hazard' is not relevant for aquaculture LUT's;

(2) There are no critical values for the LUR 'Erosion hazard' in a flat alluvial landscape;

> A good example of important land use requirements that may often be left out of models of limited geographical applicability are those LURs dealing with climatic adaptation of a crop, such as daylength and radiation. However, if it is expected that a model will be used over a wider area, where these factors are not constant, these LURs would be included.

(3) Although soil compaction is a serious problem in a LUT of mechanized grain production in a certain zone, the soil scientists in the area are unable to predict what conditions of the land make it more or less susceptible to compaction, so that the LUR 'risk of compaction' can *not* be included in the set of LUR's;

(4) Suppose the local soil scientists are able to predict compaction based on an expensive and time-consuming field test. If this test has not been performed on the map units in the study area and it is not feasible to do so because of limited time and money, the LUR can not be determined and so must be omitted from the model.

Land use requirements can be defined for one of three purposes, which are of course not mutually exclusive: to differentiate the *physical suitability* of land units, to differentiate the *yield potential* of land units, or to differentiate the *costs* of implementing the land utilization type on the different land units. The next three subsections give examples of each of these.

8.3.5.1 LURs related to physical suitability

This kind of LUR is typically a 'hazard', and influences the land use in a *negative* manner. Examples are: erosion hazard, flood hazard, drought hazard. The idea here is that excessive severity levels of the corresponding land qualities makes the land unfit for the land use. These LURs can also be used to classify land into management groups.

Given an unlimited amount of resources, any limitation could be overcome. This is fairly obvious for limitations of nutrients or water. Even limitations due to daylength could be corrected by putting supplemental lighting in the field! So, strictly speaking, there are no completely limiting land use requirements. In practical terms, however, there certainly are. The model builder determines the concept of impracticality in the context of the land utilization type.

8.3.5.2 LURs which reduce yields

These LURs typically have to do with intrinsic factors of plant growth, such as water, light, temperature, and nutrients. Some limitations to culture can also be included here: e.g. planting conditions or harvesting conditions. The model builder must determine which land qualities can reduce yield.

8.3.5.3 LURs which increase costs

Limitations can result in reduced yields; however, in the context of a land utilization type we may choose to correct (completely or partially) a limitation by increasing inputs. Any yield-reducing land use requirement can be included here as well. If certain severity levels of a land quality increase costs, the model builder expresses this by listing the *additional inputs*, which can be either *annual* (recurring) or *one-time*, at one or more specific years within the plan. The inputs are selected from the same reference list of inputs mentioned in §8.3.3. Each severity level may have a different amount of the input needed to correct the limitation.

8.3.5.4 Deciding on the number of severity levels

A major decision when defining land use requirements within a land utilization type is how many *severity levels* (also called *degrees of limitation* or *land quality classes*) are to be distinguished for the corresponding land quality. ALES allows land qualities to have from two to nine severity levels.

In the original FAO method, the levels of the various LUR's must be *commensurate*, e.g., level 3 'marginal' of one LUR must be in some sense equivalent to level 3 of all other LUR's. This is required in ALES for those LUR's which are to be used in the *maximum limitation* method of computing the physical suitability subclass. There is *no* requirement that the *number* of severity levels be the same; for example, some LUR's may never be absolutely limiting, so that they would not have the highest-numbered (worst) level 'not suitable'.

> An example of this, thinking of a LUT which includes sorghum, might be the LUR 'plant nutrition'. In a certain study area it might be the case that there is never a complete crop failure due to soil fertility problems, in which case this LUR would not need the highest-numbered class. In contrast, for the LUR 'oxygen supply to the roots' there might be some land areas where there is a persistent high water table or ponding, which would kill the sorghum plant. In this case there *would* be a highest-numbered class.

The term 'severity level' as used in ALES is a bit misleading, because the level numbers from 1 to some maximum number do *not necessarily* correspond to increasing levels of physical limitation, or to increasing additional costs, or to decreasing yields. It is certainly common that increasing severity levels do correspond with one of these, but this is not required. In some cases it is not appropriate. For example, a land use requirement of 'nutrient requirement' may have a number of severity levels corresponding to soil fertility groupings, each requiring a different combination of additional inputs; the ordering of the classes may be arbitrary.

Recall that the land use requirements will be used to determine overall physical suitability, proportional yields, and added costs of production. Therefore each severity level must differ from the others with respect to one of these. This is the *top-down* approach to determining the number of severity levels.

For example, for the land quality 'lime requirement', we could define a number of severity levels, each requiring a specific one-time quantity of added limestone. In the top-down approach, the number of levels would correspond to the possible application levels. Since the amount of limestone to be applied is a continuous quantity, there are an infinite number of application levels; however, in common agricultural practice only a discrete number of levels are specified, e.g. in 1-ton increments. So the number of classes would be the total range of possible levels, e.g. maximum 12 tons, minimum 0 tons, divided into 1-ton increments, e.g. 0, 1, 2, _, 12 tons. However, the number of classes in ALES is limited to nine, in order to keep the size of decision procedures manageable, so in this example the range 0-12 tons would have to be divided into 0-1.5, 1.5-3, etc.

Furthermore, it makes no sense to define more severity levels than can be distinguished on the basis of the available land characteristics. Put another way, the *precision* of the land quality *cannot* be higher than the precision of at least one of the land characteristics that will be used in the severity level decision tree. This is the *bottom-up* approach to determining the number of severity levels.

Returning to the example of 'lime requirement', if this is determined from topsoil pH, there is an inherent uncertainty in the measurement, as well as considerable variability within each map unit, both of which are reflected by the fact that pH is typically reported in *ranges* in soil surveys. Suppose that in a certain area pH is reported six classes, e.g. <4, 4-5, 5-6, 6-7, 7-8, >8; it would not be possible to specify more than six severity levels of lime requirement.

8.4 Land characteristics

There are three approaches to defining a set of land characteristics to serve as the basis for an evaluation: top-down, bottom-up, and middle-out.

In the *top-down* approach, those land characteristics are selected which most easily determine the severity levels of the land qualities, which have been previously listed as land use requirements for the LUT. In this approach, decision trees to determine land qualities from land characteristics are usually simple, because of the close correspondence between the qualities and characteristics. There is little or no concern with existing sources of data; it is assumed that values of the land characteristics can be obtained as needed, typically by commissioning a new survey. This approach is appropriate, and indeed preferable, if there are no good data sources, and if a new resource inventory can be carried out.

Usually, however, there is already a source of resource data available. In the *bottom-up* approach, the existing data are used to define the set of land characteristics. Then, the decision trees to determine severity levels of the land qualities from land characteristics must use this set. This will usually result in a less than ideal fit, but it has the advantage of using existing data. Different data sets may be used in the same evaluation model. Section 8.4.4 discusses how to reconcile differences between data sets.

A combination of the two approaches is the *middle-out* approach. This begins by applying the top-down approach in its entirety. Now, however, rather than go to the field to determine values for the set of land characteristics that was defined by the top-down approach, we instead define another set of land characteristics, namely, those that would have been defined in the bottom-up approach. Then we link the two, by using several *land characteristic-to-land characteristic inference methods* to allow ALES to determine the (unknown) values of the top-down set from the (known) values of the bottom-up set. These methods are: (1) land characteristic-to-land characteristic decision trees, and (2) inferring from a commensurate land characteristic.

Land characteristics in ALES can be of three types: (1) *discrete and ordered* (i.e., ordinal), (2) *discrete and unordered* (i.e. nominal), and (3) *continuous*. The following subsections explain each of these in detail.

8.4.1 Discrete, ordered land characteristics (with class limits)

These land characteristics have an underlying continuous scale, for example, _C, % slope, pH, gm/cc, and so forth. However, ALES decision procedures require that the scale be divided into (two to sixteen) *classes* that can be used in decision trees. Each class then has a *range* of values. This is in accordance with the concept that ALES evaluates *map units*, and there is always variability within map units. For example, a single value of % slope is very rarely applicable to a whole map unit, and that is why most map units are described in terms of a range of slopes, e.g. '0-3% slopes'.

The number of classes, and the class limits (dividing points) depends on (1) the precision desired in the decision procedure, and (2) the precision obtainable in the field.

The second criterion is fairly obvious: we cannot get more precision than can be mapped. With existing inventories, this decision has already been made for us; for example, slopes may have been measured in five classes. So we cannot define more classes than what was measured.

The first criterion is more subtle. We are only defining land characteristics in order to arrive at severity levels of land qualities. The number of these severity levels must correspond to perceptible differences in physical suitability, yields, or levels of inputs. Once this decision is made, it is only necessary to define as many classes of each land characteristic as will enable us to place land in one of the severity classes. In general, the number of LC classes is roughly equal to the number of LQ classes.

Often a land characteristic is used in determining the severity level of more than one land quality. For example, slope may be used in decision trees for both erosion hazard and limits to mechanization. In this case, the LC must have enough classes to allow *all* the relevant LQ's to be determined. So, for example, if a critical limit for erosion hazard is 12% slope, and a critical limit for mechanization is 15% slope, both these limits must be used as LC class limits.

Similarly, a LC is very often used to determine the severity level of a land quality in different LUTs. Here again the critical limits may be different, even for the same land quality. Again using the example of erosion hazard, in clean-cultivated crops there may be critical limits of 2%, 8%, and 15%, whereas in pasture there may be critical limits of 12% and 25%; all five of these limits should be LC class limits.

8.4.2 Discrete, unordered land characteristics (without class limits)

These land characteristics are measured in classes, but have no underlying continuous scale. An example, which might be used in evaluations involving supplemental irrigation, is 'source of water for irrigation', with three classes: 'rainfall', 'rivers', and 'springs'. These have no natural ordering, they are just three different sources of water.

The number of classes of unordered land characteristics is determined by the different possibilities; there is usually no problem in determining these. However, if several possibilities always behave the same for the land uses in the evaluation, they can be grouped. For example, a land characteristic of 'parent material' may have a single class for 'till, residuum, and colluvium', rather than have three classes, i.e. 'till', 'residuum', and 'colluvium', if these three parent materials act the same with respect to the proposed land uses.

8.4.3 Continuous land characteristics

These LCs are measured on a continuous scale, for example, pH or % slope. Each homogeneous map unit is described by a single value. The only reason to use this kind of LC is if an existing database only gives single values on a continuous scale.

By itself a continuous LC value can not be used in ALES. The model builder must set up commensurate discrete LC, and then a continuous-to-discrete inference link as explained in §8.4.4.1. Then the value of the commensurate discrete LC can be used in decision procedures.

8.4.4 Inferring a land characteristic from other land characteristics

ALES provides several methods of inferring the value of a land characteristic from one or more other land characteristics. This is required in the middle-out approach to defining characteristics explained above. The following subsections explain these methods.

Note that in all these methods ALES only performs the inference if it does not already have a (non-inferred) value for the target LC in its database. If a value is directly placed in the database, either from the keyboard or read in from a data file, it over-rides any inferred value. If you want to force the inference, you must delete the value of the target LC from the database first.

8.4.4.1 Continuous to discrete (pigeonholing)

Values of continuous LCs are transformed into values of a *commensurate* (i.e., having the same units of measurement) discrete LC automatically, if an *inference link* is set up by the model builder. You specify this link in the definition of the *discrete* LC, by choosing item 5 of menu 1.1.4a (specification options for discrete LCs with units). Tutorial 4 has an example of this procedure.

So the LCs are paired: the discrete one, used in the decision trees, and the continuous one, for which data is entered. These must both have the *same units of measurement*, and the inference link must be set up between them.

8.4.4.2 Discrete to discrete with a decision tree

ALES allows you to set up a decision tree for any discrete LC (either ordinal or nominal), by which the values of the LC are determined from any combination of other discrete LCs. This tree looks exactly like a severity level tree, in that the decision entities are LCs, but the decision values are LC classes of the *target* LC, i.e. the one for which the tree is defined.

This kind of tree is appropriate when a more abstract LC, i.e. one that is closer to a land quality or is difficult to determine during routine survey, is to be used in severity level decision trees. There may be no data values for this LC. However, data values may be available for a set of simpler LCs, from which the value of the abstract LC can be inferred.

For example, consider the LC 'subsoil permeability'. In some cases this is measured or estimated in the course of routine survey, but some surveyors may not have enough experience to do this reliably, so some published surveys may not include values for this LC. However, soil physicists can infer approximate permeabilities, certainly close enough to place a soil into one of a few classes, using simpler LCs that are routinely measured, such as soil texture, structure type and grade, type of clay, flocculation of the clay, dry and wet consistence, etc. The expertise of this soil physicist can be codified in the LC-to-LC decision tree, so that data values of the 'higher-level' LC, permeability, can be inferred by ALES from data values of the 'lower-level' LCs just mentioned.

The advantage of using this sort of tree is that the model builder can use higher-level, more abstract LCs in the severity level decision trees, thereby making them easier to build (fewer factors need be considered) and more accurate (the LCs are closer to LQs), while still being able to use low-level soil survey data that have already been collected.

8.4.4.3 Discrete to discrete by linear interpolation

ALES allows you to set up an inference link between two discrete LCs that share the same unit of measurement (i.e. are commensurate), in exactly the same manner as an inference link is set up between a continuous and a discrete LC. A typical use of this feature would be to go from a land characteristic for which data is available, to a similar land characteristic used in a severity level decision tree. The LC used in the severity level tree is typically selected in a top-down manner, i.e. according to the severity levels that were to be determined. The LC for which data is available is typically found in an existing resource inventory, such as a soil survey.

For example, if a soil survey is available with slopes reported in classes 0-3%, 3-8%, 8-15%, 15-30%, and >30%, but the ALES model was set up to use a 'slope' LC with different classes, e.g. 0-5%, 5-12%, 12-25%, and >25%, the model builder would set up a second slope LC, called, for example, 'slope-ss' (meaning slope, from the *s*oil *s*urvey), and then set up the link in the definition of the LC to be inferred, i.e. the one that is to be used in the severity level decision trees.

An obvious problem with this sort of inference is that the class limits may not match, so that a value of the base LC may be assigned to more than one value of the inferred LC. In the example above, what value of 'slope' should be assigned to a map unit with a 'slope-ss' value of '5-12%'? This range overlaps two classes of 'slope', namely 3-8% and 8-15%. The land unit could in fact all be in the first class (if the slopes are all less than 8%), all in the second (if the slopes are all greater than 8%), or some land could be in each class.

ALES assumes that the land is *uniformly distributed* among the inferred classes, according to the *proportion* of the range of each class. In the current example, ALES would assume that (8-5)/(12-5) or 43% is in the 3-8% class, and that (12-8)/(12-5) or 57% is in the 8-15% class. Similar calculations can be carried out for any two sets of ranges. In the situation where a range of an inferred class is wider than the range of a base class, all the values of that base class will end up in the same inferred class. But in general, the inference procedure will result in a *multi-valued* land characteristic value. In the current example, it would be: 3-8% (p = .43), 8-15% (p = .57). Note that 'p' in this syntax is a *proportion*, not a probability.

The multi-value is propagated as described in the section on multi-values, below.

8.4.4.4 Continuous to continuous by formulas

ALES allows the model builder to enter mathematical *formulas* with which to calculate the value of a *continuous* land characteristic (LC) from a *set* of *other* continuous LC's. Typically the *derived* LC is more complex and less easily measured than the *base* LC's. For example, the predicted soil loss by water erosion, in units of T ha^{-1}, could be computed from the silt content, very fine sand content, proportion of the soil surface covered by stones, and slope, all measured in percent.

> A formula is *not* appropriate for discrete (classified) data. In this case you should use a LC-to-LC *decision tree*. An example would be inferring hydraulic conductivity (fast, medium, slow) from particle size class (sand, sandy loam, ...), structure (granular, blocky, prismatic...), vertic characteristics (none, weak, strong) etc. In this example, all the *base* LC's are discrete; there is no valid way to combine them with a *formula*. Instead, a decision tree can be used to combine their classes.

As with any continuous LC, those that are determined by formulas must be related to a *commensurate discrete* LC to be useful in ALES evaluations. This is explained in §8.4.4.1, above.

Formulas are entered for *continuous* LC's, with option '3 Infer by a formula' of menu '1.1.4c Specification options for continuous LC'. They can have the following components:

1. *Constants*, i.e. numbers (integer or decimal), e.g. '13.4'

2. *Variables*, i.e. the land characteristics whose values should be substituted into the formula. If the LC code contains punctuation that could be confused with mathematical operators, you must write the code between quotes, e.g. LC code 'c-N' could be confused with LC code 'c' minus LC code 'N'; LC code 'abs' could be confused with the function for absolute value.

3. *Arithmetic binary operators*: +, -, *, /, \ (integer division), # (modulus) ** (exponentiation)

4. *Logical binary operators*: = (EQ), '= (NEQ), > (GT), >= (GE), < (LT), <= (LE), & (AND), '& (NAND), ^ (OR), '^ (NOR). The prefix ' means NOT, e.g. '& stands for NAND. The written forms, e.g. AND, are alternate ways to enter the operator, e.g. '&'.

5. *Unary mathematical functions*: ABS, LN, LOG, EXP, SQR (square root), SIN, COS, TAN, CSC, SEC, COT (argument in radians)

6. The *ternary conditional operator* IF: syntax: IF(truth_condition, value_if_true, value_if_false), where 'truth_condition' uses logical operators (AND, OR etc.) and 'value_if_...' is any legal formula

Formulas are entered with standard algebraic notation; operators have the usual precedence.

Examples of valid formulas:

1a. **SQR(dbr)/clay**1.5**

In standard math notation this is: . Note that this is different from:

1b. **SQR(dbr/clay**1.5)** or, equivalently, **SQR(dbr/(clay**1.5))**

Both of which are, in standard math notation:

2. **IF(dbr>1.5,cfsub/2,cfsup*0.35)**

In standard math notation this is:

8.4.5 Computing land characteristic inferences directly

In normal use, ALES *automatically* uses inferences *only when necessary* to determine *land quality* values, i.e., when it computes the *evaluation matrix.* Thus a land characteristic *must be used* either in (1) a *severity level decision tree* or (2) as the inference basis of another LC that is so used. In other words, if ALES doesn't encounter a LC during its traversal of the decision trees, it will *not* infer its value.

In addition, if the model user has *directly entered* a value for a land characteristic, *ALES will not infer it* during the computation of land qualities, even if there is an *inference link* set up for the derived characteristic. The idea here is that the model user must have had more specific information about a map unit if a value was entered.

Both of these problems can be overcome with the use of Option 9 'Infer Land Characteristic values' of Menu 1.3 'Data'. If this is selected by the model user, ALES *will* infer values of the set of derived LC's specified by the user, thus over-writing any directly-entered values. The idea here is that the model user is directly asking ALES to infer values.

To see the values of derived LC's *not* used to determine LQ's, the model builder must set up a *data entry template* which includes these LC's.

8.5 Decision procedures

The heart of an ALES model is the set of *decision procedures* by which land suitability may be assessed. There are several steps to this, as explained in the computation diagram of Appendix 3. The following sections explain the different decision procedures used in ALES models.

8.5.1 Determining physical suitability subclasses

The FAO method differentiates between physical suitability *classes* and *subclasses*. Classes are used to express the overall physical suitability rating on a scale of 1 (completely suitable) to some maximum (physically unsuitable); the FAO framework suggests that four classes be used, corresponding to (1) complete suitable, (2) suitable, (3) marginally suitable, and (4) unsuitable. Within each class, subclasses are used to designate the kind of limitations or other qualities that result in the placement of a map unit in the class. Subclasses can be used as management groupings. For example, subclass '4e/me' might indicate land that is unsuitable because of high risk of erosion ('e') and difficulties for mechanization ('me').

In ALES, the model builder can use either classes or subclasses as the result of a physical evaluation; it is strongly suggested that subclasses be used, as they are much more informative.

ALES has two ways to determine the physical suitability subclass of a land unit for a land utilization type from the severity levels of land qualities. In the most general case, the model builder may construct a *physical suitability subclass decision tree*, which ALES follows during computation of physical suitability, to combine severity levels of LQs into a composite physical suitability. A second method is the *maximum limitation method*, in which the model builder specifies that the maximum value of one or more LQs shall determine the composite physical suitability. As explained below, these two can be used together.

8.5.1.1 Physical suitability subclass decision trees

These trees (Menu 1.1.5.8, Option 1) allow the model builder to assign specific physical suitability classes or subclasses to any combination of severity levels of land qualities. The evaluator can omit some land use requirements from the tree (e.g. because they do no affect physical suitability, only yields or inputs), and can weight some requirements more than others. The result class numbers do *not* have to correspond to the severity level class numbers. For example, if it has been decided to only use two physical suitability classes, (1) suitable, (2) unsuitable, but you have also decided to rate land qualities in four severity levels, then you would want to associate severity level 4 with physical suitability class 2. With the decision tree, you can do this by entering '2' at the appropriate leaves of the tree.

When entering a subclass designation, ALES allows the evaluator to append to the class number as many *limiting* qualities as are on the current path. The suffixes can be in any order. In this manner, the evaluator can (1) omit some qualities, thereby emphasizing the others, and (2) order the suffixes to indicate the most important ones first.

8.5.1.2 The maximum limitation method

In this method, the model builder specifies (Menu 1.1.5.8, Option 2) the set of land use requirements whose corresponding LQs are to be considered limiting. During computation of physical suitability, ALES examines the severity levels of the specified LQs, and assigns the physical suitability class from the *highest-numbered* severity level. It then appends to the suitability class the land use requirement codes for the land qualities that were most limiting, resulting in a subclass.

For example, consider a map unit with the following ratings: LQ 'a': 1/3 (i.e. 1 on a scale of 1 to 3), LQ 'b': 3/5, LQ 'c': 3/3, LQ 'd': 2/2. Suppose in addition that physical suitability is measured in four classes. Then the physical suitability subclass, by the maximum limitation method, would be '3b/c'.

For this method to be meaningful, the degree of limitation of a particular severity level number must be equivalent over all the land qualities, and the number of physical suitability classes must be the same as the maximum number of severity levels. For example, severity level 3 for erosion hazard must be, in some sense, equally 'as bad as' severity level 3 for nutrient availability, and severity level 3 for flood hazard. It is not required that the highest severity level be the same, since some land qualities may never be as limiting as others.

8.5.1.3 Combining the two methods

If the evaluator defines *both* a physical suitability subclass decision tree *and* a set of maximally limiting land use requirements, ALES will first use the tree to determine a provisional physical suitability subclass. It will then examine the severity levels of the maximally limiting LQs, *not including* any that were already taken into consideration by virtue of having been included in the decision tree, and modify the provisional subclass if necessary to append any equally-limiting LQs, or replace the subclass with more limiting LQs.

8.5.2 Determining proportional yield

Predicting the proportional yield, i.e. the percentage of the optimum, for an output of a land utilization type, involves three model components, all of which are based on the actual severity levels of land qualities: a *proportional yield decision tree*, a set of *proportional yield factors*, and a set of *limiting yield factors*. Any combination of these three components can be used. (If none are defined in the model, the proportional yield is assume to be 100%.)

8.5.2.1 Proportional yield decision trees

The most general way to assign proportional yields is to use severity levels of land qualities as decision entities in a decision tree (Menu 1.1.5.6, Option 2). The advantage of this method is that it allows all interactions to be taken into account. The model builder can specify a proportional yield, on the interval 0 to 1, for each combination of severity levels of the land qualities that are in this proportional yield decision tree.

The disadvantage of this method is that , if many factors are to be considered, the tree may become excessively large and unwieldy. If some land qualities affect yield in essentially a strictly multiplicative or limiting manner, the model builder can leave them out of the proportional yield decision tree, and account for them in one of the following two methods.

8.5.2.2 Proportional yield factors

Some land qualities affect yield in a *multiplicative* fashion. More precisely, we may say that if a land unit has a certain severity level assigned to one of these land qualities, the optimum yield must be multiplied by a factor between 0 and 1 to arrive at the predicted yield. If there is more than one of these qualities in an evaluation, all such factors are multiplied together. For these land qualities, there is a shortcut to using them in proportional yield decision trees, i.e., assigning *proportional yield factors*.

The model builder may use proportional yield factors (Menu 1.1.5.6, Option 3) to partially or completely replace proportional yield decision trees. In preliminary stages of model building, it may be useful to define only yield factors, and no tree. The model builder can then examine the predicted yields to see which interactions need to be taken into account by means of a proportional yield decision tree.

In the actual calculation, if a particular land quality has been encountered in the decision tree path, any proportional yield factor that it may have is ignored. Since the tree has precedence over the yield factors, there is no harm in leaving the factors in the model, even for LQs which are included in the tree.

8.5.2.3 Limiting yield factors

Some land qualities affect yield according to the agricultural *law of the minimum*, namely, that the most limiting factor determines yield. More precisely, we may say that if a land unit has a certain severity level assigned to one of these land qualities, the yield can be *no higher than* this minimum. For these land qualities, there is a shortcut to using it in proportional yield decision trees, i.e., assigning *limiting yield factors*.

The model builder may use proportional yield factors (Menu 1.1.5.6, Option 4) to partially or completely replace proportional yield decision trees. In the actual calculation, if a particular land quality has been encountered in the decision tree path, any limiting yield factor that it may have is ignored.

8.5.3 Land use requirement decision trees

The severity levels of a land quality, corresponding to a land use requirement, can only be determined in one way: from a set of discrete land characteristics (either ordinal or nominal), by using a decision tree. Thus the model builder *must* build one severity level decision tree for each land use requirement in the LUT specification. Otherwise, an LUR can not be used in the decision procedures of §8.5.1 and §8.5.2.

8.5.4 Missing data: using alternate data sources in decision trees

ALES provided several means of dealing with *missing data*. First, if a particular data value is not required during the computation, the evaluation can proceed without problems. For example, on land that is very rocky, we may not have chemical analyses. If the rockiness is used to place the land in the 'unsuitable' physical suitability class (i.e. the rockiness is a criterion in the physical suitability class decision tree), ALES will never need to determine the fertility, so will never need the missing chemical analyses.

Sometimes a value of one data item can be inferred from another; this is covered above, under 'Land Characteristics'.

The model builder can specify *alternate criteria* to be used in a decision tree, when a data value is missing. This is the purpose of the 'unknown' branch, marked with a '?', in each decision tree. A subtree can be placed at this branch, which will be followed in the case of missing data for the entity at the current level. This allows the model builder to express the idea of using one type of data 'or' another.

For example, in the Fertility Capability Classification (Sánchez et al. 1982), most fertility limitations, which in ALES correspond to land qualities, can be determined by several methods, which are ranked in order of preference. Section 3.2.3 of this User's Manual gives an example, for the LQ 'high P-fixation by iron'. The preferred diagnostic criterion is the LC 'ratio of free Fe_2O_3 to clay in the topsoil'; however, if this is not available, the LCs 'hue of the topsoil matrix' and 'topsoil structure' may be used instead. This is expressed in the ALES decision tree by rooting a subtree, containing the decision criteria 'hue of the topsoil matrix' and 'topsoil structure', at the 'Unknown' branch for the LC 'high P-fixation by iron'.

8.5.5 Controlling complexity in decision trees

A major problem with decision trees is that their size, measured as the number of leaves, grows exponentially with the number of factors considered. A practical limit is about six factors; beyond this the size of a complete tree is prohibitive.

A detailed mathematical analysis of this process may be found in Rossiter (1988a); a summary is presented in this and subsequent sections. There are two appropriate measures of size for a set of decision trees, namely: (1) the number of leaves, designated by *N*, and (2) the number of trees, designated by *T*, in the set. The size of a complete decision tree is inherently exponential in the number of discriminant entities considered (e.g. LCs in a severity level decision tree).

Consider a complete decision tree of n entities, each with m branches or choices (this m is also called the *branching factor*). Such a tree will have $N = m^n$ leaves, so that as each new decision entity is added to the tree, the number of leaves is multiplied by the branching factor of the new entity. Trees can get very large very quickly. In ALES the decision entities have branching factors (number of classes) of at most 9 (LURs) or 16 (LCs), with 4 being a typical value. So if we consider only six entities, we would already expect a complete tree of $4^6 = 4096$ leaves. Fortunately, there are ways to control this inherent complexity; the following sections explain these methods.

8.5.5.1 Constants vs. variables

As in mathematics, also in land evaluation we can talk about *constants*, i.e. data values that do not vary over the entire area to be evaluated, and *variables*, which do vary. A model that incorporates only a narrow range of land utilization types, and is meant for application in a small area, will not need to consider many variables, because over the range of applicability of the model, they are constants. Conversely, a wide-ranging model will have many variables.

A good example of the effect of the number of variables on model size is erosion hazard (see, for example, Hudson 1981). The Universal Soil Loss Equation considers rainfall erosivity, soil erodibility, crop management, and conservation practices. So it would seem that all these would have to be considered when constructing a severity level decision tree for erosion hazard. However, within the context of a single LUT, the crop management and conservation practices are *constant*, since they are part of the LUT definition. Hence the decision tree need only consider the first two factors, with the others held constant and used in determining the final values.

Continuing with the same example, in a limited geographical area with a single rainfall regime (e.g. intensity of severe storms, total rainfall, average rainfall intensities), the rainfall erosivity could be considered a constant, and hence omitted from the decision tree.

There is a definite trade-off here: if your model is only of limited applicability, you can treat many things as constants, so that your decision trees are small. On the other hand, if you want your model to be of wide applicability, you must consider more variables, resulting in larger decision trees.

8.5.5.2 Make decisions early

The easiest way to control complexity is to first consider the factors that will allow you to reach a decision as early as possible. In this way, the tree need not have the same path length to all decisions. For example, if flooding absolutely prohibits a certain use, it would make sense to put a land quality of 'flood hazard' first in a physical suitability decision tree, with the decision for the classes other than 'no flood hazard' being already the highest class (i.e. not suitable).

8.5.5.3 Include only the most significant factors

In general, the severity levels of land qualities depend on many land characteristics. The model builder's first instinct is to try to include all of these in the decision tree. In most cases, however, a subset of all the relevant LCs can be used, with very little loss of precision in the final decision. A smaller tree is a better tree, for two reasons: (1) it is easier to build and modify, and (2) it is easier for the model user to understand. The smallest tree that can correctly assign severity levels for almost all cases is the best tree to build.

A similar situation occurs when building proportional yield decision trees. Some land qualities have only a small influence on yield, either always or given some combination of severity levels of other land qualities. These can safely be omitted from the yield trees, since their inclusion will not significantly affect the predicted yield. Given that the precision of the input data is not great (e.g. most class ranges are at least 10% of the total range for a land characteristic), it is not realistic to predict yield more precisely. Hence if you are about to add a land quality to a decision tree, and the different severity levels of that LQ would only affect yield by 10%, it is probably a correct decision to leave the land quality out of the tree.

8.5.5.4 Introduce intermediate land characteristics

The ability to create LC-to-LC decision trees (§ 8.4.4.2) allows the model builder to break a large severity level decision tree into several smaller components: a severity level decision tree whose decision entities are abstract or 'high-level' LCs, and a set of LC-to-LC decision trees to infer values of the high-level LCs from other LCs. In fact, 'intermediate' LCs can be set up by the model builder specifically to group sets of related low-level LCs to reduce the size of severity level decision trees.

If the original set of LCs in a decision tree is combined pairwise into intermediate LCs, the total number of leaves needed to express the same inference that required m^n leaves in the original tree is $m^{(n/2)} + (n/2)m^2$. The branching factor m is raised to a lower power, in fact half of the original. However, the number of trees T has increased from 1 to $(1 + n/2)$.

As a simple example of the savings which may be obtained by such a pairing, consider a realistic problem in land evaluation: the determination of the severity level of the 'climate' land quality for tea in Sys (1985, p. 316) from eight land characteristics, each with four classes (possible values). Here $n = 8$, $m = 4$, so that the size of the complete tree would be $4^8 = 65536$, which is prohibitively large. Introducing four intermediate land characteristics, each pairing two of the original eight land characteristics, would result in five trees with a total of $4^4 + 4*4^2 = 320$ leaves, a realistic number.

This divide-and-conquer strategy can be continued by combining the intermediate LCs pairwise into another level of intermediate LCs. Theoretically this results in greater savings, but in the size of problems likely to be encountered in ALES, the benefits are not great.

LCs can be combined in any number, not just in pairs. Grouping more LCs into each intermediate LC results in more total leaves, but fewer trees. For example, in a set of trees with $n = 16$ decision entities, each with $m = 4$ classes, dividing into groups of 4 entities as opposed to pairwise would result in a third as many trees, and about 16 times the number of leaves. This is a substantial difference, but not as dramatic as the difference from introducing intermediate LCs in the first place.

This abstract discussion has up to now avoided the most important point to be considered by the model builder: for this divide-and-conquer strategy to be valid, land characteristics to be combined into intermediate LCs must not interact differently from each other with the remaining entities in reaching the final decision. The combinations to use, as well as the degree of splitting, should be suggested by the nature of the entities. In practical terms, each tree should not have a path length greater than four, and a length of three is more easily manageable. Also for the understanding of the model user, a series of smaller trees breaks down the reasoning into more easily-comprehended steps.

Continuing with the previous example from Sys (1985), a model builder might decide to combine the two characteristics 'length of dry season' and 'length of dormancy' into a derived characteristic named 'lengths', combine the two characteristics 'mean annual temperature' and 'mean minimum temperate' into a derived characteristic named 'mean temperatures', combine the two characteristics 'average minimum temperature of coldest month' and 'average maximum temperature of warmest month' into a derived characteristic named 'average max/min', and combine the two characteristics 'annual rainfall' and 'relative humidity' into a derived characteristic named 'moisture'. Each tree will have $4^2 = 16$ leaves, and four possible results, corresponding to the original *m*. These four derived characteristics can be combined into a single tree, having $4^4 = 256$ leaves. The total number of leaves will be *320*, and there will be five trees. These are certainly manageable and understandable numbers. This example is typical of the size of decision tree sets found in land evaluation problems.

8.6 Multi-valued (probabilistic) land data

In general, each map unit in an ALES database has a *single* data value for each land characteristic used in the evaluation model. Occasionally, however, it may be necessary to describe a map unit with more than one data value. For example, within a single map unit there may be lands of several slope classes. The best way to handle this sort of situation is to define compound map units, and enter single-valued data on their homogeneous constituents. However, it is possible in ALES to enter a *multi-value* in place of a single data value for a land characteristic.

The data entry syntax to enter a multi-value is as follows:

val=prob,val=prob,...,val=prob

where 'val' represents a permissible data value, and 'prob' represents a linear probability, in the interval (0,1). The probabilities in the entire multi-value must sum to 1.0. A multi-value may be entered for any data item in a data entry form. The 'pop-up' feature of selecting among a set of classified data values can't be used when entering a multi-value, as this results in only a single data value.

For example, if a map unit is described as being composed of 60% somewhat poorly and 40% poorly drained soils, in close proximity and inseparable on a map, when asked for the drainage class in a data entry form, we would enter:

```
spd=.6,pd=.4
```

Note that .6+.4 = 1.0, as required. This is an example of a *spatial series* of related data.

Another example is a *time series* of climate data. In this case, the proportions are typically derived by dividing the frequency of a data value in the series by the total length of the series.

When ALES encounters a multi-value in its evaluation, it separately evaluates each case in the multi-value, and combines them at the end. It is not necessarily true that the evaluation results will be multi-valued, since there might be no difference from the point of view of a land use between the values. In the previous example, if a land utilization type is not appropriate on any but well-drained soils, the multi-valued drainage class would reduce to a single result.

The meaning of the proportions is *not* based on a rigorous definition of probability. We can look at these linear proportions two ways: either that, say, 60% of the land area has rating 1, or that it is 60% certain that all the land has rating 1. For soil or other space-series characteristics, the first interpretation is usually preferable. For climate or other time-series characteristics, the second interpretation is usually correct.

8.7 Working with multiple datasets

'Data' Menu 1.2 item 8 - 'Erase the Database', allows the evaluator to *delete all map unit definitions and data*, leaving an empty ALES data base (you will be asked for confirmation before ALES actually deletes the data). The effect is the same as if you deleted each map unit in turn with the **F4** key when viewing the list of map units when selecting a map unit description for editing (Menu 1.2, choice 1).

Why would one want to erase the database? Mainly to allow the testing or use of a model (knowledge base) with more than one dataset. Along with several other options in the 'Data' menu (1.2), multiple datasets can easily be evaluated with the same model.

The basic idea is that you maintain datasets in relational tables (i.e. disk files with one line per map unit), for example, by *writing* first the *definitions*, and then the *data*, to disk with 'Data' menu options 3 and 6, respectively. When using option 6, you can save all the land characteristics in one relational table by defining a *template* that contains fields for every land characteristic. Or, you can split the database into multiple files, each with its own template. For example, you might decide to put soils data in one relational table, and climate data in another. In any case it is important to make a note of which template you used to write the data, as you will have to tell ALES which template to use when you reload the same data. §9.5.1 and _2 give additional information about these disk files.

Once you have saved a dataset (both the map unit definitions and their data) outside of ALES, you can select 'Data' menu (1.2) option 8 to erase the entire dataset from ALES. Now you can create a new dataset for analysis. You can do this from the keyboard within ALES (using Options 1 and 4 for the map unit definitions and data respectively) or you can *read* a prepared dataset in from a relational table (using Options 2 and 5 for the map unit definitions and data respectively). If you read from a file, you will have to specify the template that lists the land characteristics for each field. The prepared dataset may have been created by some other program (e.g. a database manager) or it could be the result of saving an ALES dataset as described above.

9 Interfacing ALES to other systems

ALES provides several ways to *exchanging information* with other computer base systems, including:

√ relational data base programs (e.g. Paradox, dBase and compatibles, RBase)

√ the database component of geographic information systems (e.g. the 'info' part of ARC/INFO)

> The interface to the IDRISI Geographical Information System is built into ALES. Refer to the next chapter and to Tutorial 5 for details.

√ word processors

√ spreadshetts (e.g. Quattro Pro, Lotus 1-2-3)

This chapter first explains what options are provided and in what circumstances they would be used. Then, the mechanics of using them are covered in the section 'File formats'.

There are *three kinds of information* that can be exchanged between ALES and other database systems:

(1) land mapping unit *definitions*
- (1.1) from an ASCII file to ALES
- (1.2) from ALES to an ASCII file

(2) land mapping unit *data*
- (2.1) from an ASCII file to ALES
- (2.2) from ALES to an ASCII file

(3) evaluation *results* (only from ALES to an ASCII file)

Data are exchanged in the form of DOS *text files* (8-bit ASCII). Data is written from the source program (ALES or another program) to a DOS text file, and then this DOS file is read by the destination program (the other program or ALES).

ALES can be thought of as a key part of an *integrated land evaluation system*, with other programs serving both *preprocessing* (i.e., before ALES) and *postprocessing* (i.e., after ALES) functions.

9.1 Preprocessing

A common reason for reading definitions or data *into* ALES is that there is an existing database with the appropriate information, and reading it directly into ALES not only saves typing, but ensures consistency between the two databases. This situation might arise if there is a national soils database which contains land unit definitions or data.

Although ALES V4 provides a way for continuous land characteristic values to be combined according to formulas, you might find it easier to perform simple manipulations, e.g. of laboratory data, in a spreadsheet or database, and then import the calculated data values to ALES.

9.2 Postprocessing, including GIS

A typical reason for writing *land unit definitions or data from* ALES is to make these available to a another database program, for example, a natural resources database. This would be useful if the data was originally prepared within the ALES environment. ALES can be used as a data collection system, either in data entry or consultation mode, which may be easier to use than commercial database systems.

Evaluation results are written *from* ALES in order to further process them with geographic information systems, spreadsheets, linear programs, and other programs that perform analysis that is beyond the scope of ALES.

ALES predicts the performance of each land unit in isolation. However, a land user typically needs a mix of different enterprises (LUTs) in an integrated operation. A dairy farm needs a certain proportion of grain crops, forage, and pasture. Allocation of land in the correct proportion can be done with an *optimization* program, with the input data to the program being the results of the ALES evaluation.

ALES evaluates map units or delineations. In many applications the geographic location, with respect to roads, other map units, villages etc. is important. Although the geographic characteristics of each delineation can be input into the ALES database, it is usually more convenient to conduct the geographical analysis in a *geographical information system* (GIS). The ALES evaluation results, predicted yields, and individual land quality ratings can be used as *data overlays* in the GIS.

9.3 Word processors

The Report menu of ALES allows you to *print* evaluation results, as well as components of the model such as decision trees. Instead of sending these reports to the printer, you can send them to a DOS text file, which can be imported to any word processor for alteration, formatting, and printing.

The procedure to do this is straightforward. After you request a particular report, you will be asked where you want the report sent. Selecting 'V' will 'print' the report on the video screen, allowing you to preview it. Selecting 'P' will, of course, send the report to the printer. Selecting 'C' will send the file out a communications port (e.g., a modem connection or serial transfer cable to another computer). Selecting 'F' will send the report to a DOS text file. If you select 'F', you will be asked to enter a file name for the text file.

You may encounter problems when you read the text file into your favorite word processor. ALES writes the file in a straightforward manner, namely, each line is terminated by a 'Return' character, ASCII code 13. There are no other formatting characters. However, if you have included characters whose codes are not in the '7-bit ASCII' range 33.-126. (decimal) in class names, land characteristic names or values, etc. the word processor may interpret these as its own 'hidden' codes, and the text may appear garbled. Some word processors from the 'dark ages' of the PC only work with 7-bit characters and use overstrike and backspace for extended characters.

Good examples of this kind of 'problem' character are non-English characters, such as accented vowels. For example, 'í' has extended ASCII code 161 (decimal). On an English keyboard, you would have entered it as `Alt 1 6 1`, typing the numbers on the *numeric* keypad, but on a French or Spanish keyboard you could have typed it directly. You will probably not encounter problems with a non-English word processor.

9.4 Transferring data between ALES evaluation models

In ALES there is no shared database between evaluation models; each model has its own separate database as well as knowledge base. You may want to *transfer land unit data and definitions* between ALES evaluation models. The way to do this is:

(1) write the definitions and data from the *source* evaluation to separate temporary DOS files (Menu 1.3, options 3 and 6);

(2) read the definitions from the first temporary file into the *target* evaluation (Menu 1.3, option 2);

(3) set up a data entry template in the target evaluation that matches the template with which the data was written (Menu 1.3, option 7); and

(4) read the data into from the second temporary file into the target evaluation (Menu 1.3, option 5).

9.5 File Formats

The following sections describe the file formats for those who are writing their own programs to exchange data with ALES, or who need to set up data tables in relational databases.

9.5.1 File format: Land unit definitions

This is information on the map units in the project area: their identification codes, descriptive names, and areal extent. For compound map units, the component homogeneous map units and their proportions within the map unit must also be given. This information can be read into ALES from a text file that has been output by another program or prepared by a word processor. Alternately, it can be entered as map unit definitions within ALES and later written out to a text file that can be read by another program or a word processor.

You request that land unit definitions be *written* to a DOS file with Option 3 ('Definitions,.To a disk file, Write') of Menu 1.3 ('Data'); you request that land unit definitions be *read* from a DOS file with Option 2 ('Definitions,.From a disk file, Read') of Menu 1.3 ('Data').

Information is exchanged as *DOS text files*, in which each *record*, or line, of the file is terminated by a *line feed character* (ASCII 10 decimal), and fields within a record are *delimited* by almost any ASCII character, selected by the user. On output from ALES, the user specifies the delimiter just before the file is written; on input, it is specified just before the file is read. The following characters are *not* acceptable delimiters:

" (double quote) (ASCII 34.) -- this is already used to enclose text fields
^V(SYN) (ASCII 22.) -- this is used in the ALES data structures
^J (LF, line feed) (ASCII 10.) -- this is already used to delimit records.
. (period) (ASCII 46.) -- this can appear in numbers.

A file can have any name and extension; the ALES user specifies these when the file is about to be read or written. Files have the following format:

LMU 1
LMU 2
...
LMU n

where each 'LMU i' is one line, or record, with the Land Mapping Unit specification.

Each line has the following format:

{LmuId} {d} {Name} {d} {Type} {d} {Extent}

where {d} is the user-selected *delimiter character*, {LmuId} is the *map unit code*, {Name} is the *descriptive name*, {Type} is either 'h' (homogeneous) or 'c' (compound), and {Extent} is the extent of this map unit, in terms of the unit of measurement used in the evaluation, in the project area. The fields {LmuId}, {Name}, and {Type} are enclosed in double quotes on output from ALES; on input, the quotes are optional.

If {type} is 'c', in other words the map unit is compound, these four fields are followed on the same line by a series of fields, two for each constituent, with the format:

{d} {LmuId} {d} {Percentage}

where {LmuId} is the code of the *homogeneous constituent*, and {Percentage} is the *percentage* of the compound map unit, from 0 through 100, made up by this constituent. The field {LmuId} is enclosed in double quotes on output from ALES; on input, the quotes are optional. The percentages on a line must sum to 100.

On input, a {LmuId} named as a constituent of a compound map unit must already be known to ALES at the time it is encountered in the input file; this will be the case if it is defined as a homogeneous map unit earlier in the same file. On output, all homogeneous LMUs are written before the compound ones, so that a LMU definition file written by ALES can always be read back.

Here is an example file, showing three map units (the field delimiter in this example is the comma, ASCII 44 decimal):

```
"Aa","Alpha silt loam, 3-8% slopes, shallow phase","h",1000
"Bb","Beta fine sandy loam, 8-15% slopes","h",2000
"AB","Alpha-Beta association","c",1500,"AaA",60,"BbB",40
```

In this example, the Alpha-Beta association has two components: Alpha, with 60% of the area, and Beta, with 40%. Notice that it must follow the definitions of its constituents.

9.5.2 File format: Land unit data

This is information on the land characteristics for map units in the project area, in other words, all the information about the map unit except its definition (§9.5.1). This information can be read into ALES from a text file that has been output by another program or prepared by a word processor. Alternately, it can be entered within ALES and later written out to a text file that can be read by another program or a word processor.

You request that land unit data be *written* to a DOS file with Option 6 ('Data,.To a disk file, Write') of Menu 1.3 ('Data'); you request that land unit data be *read* from a DOS file with Option 5 ('Data,.From a disk file, Read') of Menu 1.3 ('Data').

The file format is the same as given in §9.5.1. The restrictions on delimiters given there are also applicable. However, the line format differs: in land data files, each line has the following format:

{LmuId} {d} $\{LcVal_1\}$ {d} $\{LcVal_2\}$ ... {d} $\{LcVal_n\}$

where {d} is the user-selected *delimiter* character, {LmuId} is the *map unit code*, and $\{LcVal_i\}$ is the *value of a land characteristic* that is listed for *field number* 'i' (out of 'n' fields) in the data entry template. Non-numeric fields are enclosed in double quotes on output from ALES; on input, the quotes are optional.

Before data on a land unit can be read into ALES, that land unit must already have been defined, either by having been read in from a file (see previous section) or by having been entered using Option 1 ('Definitions, From keyboard, Enter or edit') of Menu 1.3 ('Data').

The number of fields and their order is specified by the evaluator by means of a *data entry template*, which must be completed prior to input or output of land unit data, using Option 7 ('Templates') of Menu 1.3 ('Data'). The template has an identifying name, and an ordered list of land characteristics, each of which corresponds to one field in the record. Before a land characteristic can be listed as a field in a template, it must in turn have been defined using Option 4 ('Land characteristic descriptions') of Menu 1.1 ('Reference Lists').

The *data fields* in the input or output record are the *class abbreviations* (for discrete land characteristics) or the actual *numeric value* (for continuous land characteristics).Here is an example. Suppose we have defined a template named 'data', with three fields, corresponding to the three land characteristics 'slope', 'texture', and 'reaction'. Note that the order of these fields in the template definition is important: ALES expects to find the data fields in the DOS file in this same order. These land characteristics, as well as the template, must have been previously defined by the evaluator. Suppose that these characteristics were defined as follows:

'slope' : discrete, 5 classes, abbreviations 'A', 'B', 'C', 'D', and 'E'
'texture' : discrete, 12 classes, standard abbreviations, e.g. 'sicl'
'reaction' : continuous, units of measure is pH scale

Here is a sample file, showing data for three map units (the field delimiter in this example is the semicolon, ASCII 59 decimal).

```
"Aa";"A";"sicl";6.5
"Bb","E","sl";7.4
"Cc";"B";"c";5.7
```

In this example, map unit 'Aa' has a slope of class 'A', i.e. nearly level, a texture of 'sicl', i.e. silty clay loam, and a reaction of pH 6.5.

9.5.3 File formats: Evaluation results

The information in the ALES evaluation matrix can be output in a form that can be read into relational database managers.

You request that evaluation results be *written* to a DOS file with Option 4 ('File evaluation results') of Menu 1.4 ('Evaluations'). The map units and LUT's included in the resulting disk files are exactly those that are in the *current evaluation matrix*. So to write just a subset of the entire evaluation, you first have to re-compute the evaluation, selecting just the subset you want to report.

Once you select this option, you then select the kind of file to write by choosing in menu '1.4.4 Output to a disk file'. There are *five* possibilities.

(1)the *suitabilities* of map units for each LUT

(2) the land quality *severity levels* of map units for each LUR within each LUT

(3) the *predicted yields* on map units for each output in each LUT

(4) the *by-year cash flow summary*.for each map unit in each LUT

(5) a *spreadsheet* of any result type.

The following sections explain each format in detail.

9.5.3.1 Suitabilities

Suitabilites are written out as one line per land unit/land use pair. The database *key* is the first two fields: land mapping unit code and land utilization type code. These are followed by 10 *fields* which give different aspects of the evaluation results. Thus each line gives information on the suitability of one land unit for one land use. The record format is:

{LmuId} {d} {LutId} {d} {PSC} {d} {PSSC} {d} {PSSC #} {d} {GM} {d} {ELSCg} {d}
_{ELSCg #} {d} {NPV} {d} {ELSCn} {d} {ELSCn #} {d} {BCR}

where {d} is the *field delimiter*, {LmuId} is the *map unit code*, {LutId} is the *land utilization type code*, {PSC} is the integer *physical suitability class*, {PSSC} is the *physical suitability subclass*, {GM} is the predicted *gross margin*, in currency per hectare-year, {ELSCg} is the *economic suitability class* based on the gross margin, one of the standard FAO classes: 's1', 's2', 's3', 'n1', or 'n2',{NPV} is the predicted *net present value*, in currency per hectare, {ELSCn} is the *economic suitability class* based on the NPV, one of the standard FAO classes: 's1', 's2', 's3', 'n1', or 'n2', and {BCR} is the *benefit/cost ratio* of the present value of cash in vs. out.

The numeric fields {PSSC #}, {ELSCg #} and {ELSCn #} are integer recodings of the text strings given in the {PSSC}, {ELSCg}, and {ELSCn} fields, respectively. These are included because some programs or database systems into which this data may be read can't deal with string information. The integers given in {ELSCx #} are a standard translation:

integer	*FAO class & meaning*	
1	s1	suitable
2	s2	moderately suitable
3	s3	marginally suitable
4	n1	economically unsuitable
5	n2	physically unsuitable

However, the integers given in {PSSC #} are not standard. These are different for each evaluation matrix. The correspondence between {PSSC} and {PSSC #} is shown in an alert box on the screen after ALES finishes writing the output file. Also, this correspondence can be determined by inspecting the output file itself.

Here is some sample output:

```
"AaA","hay",2,"2f/me",2,30.88,"s3",3,27.20,"s2",2,3.2
"AaA","wheat",1,"1",1,10.20,"n1",4,20.00,"s3",3,3.3
```

9.5.3.2 Severity levels

The severity levels output is one line per land unit/land use/land use requirement triple. The database *key* is the first three fields: land mapping unit code, land utilization type code, and land use requirement code. These are followed by a field which gives the integer value of the severity level. The record format is:

{LmuId} {d} {LutId} {d} {LurId} {d} {severity level}

where {d} is the field *delimiter*, {LmuId} is the *map unit code*, {LutId} is the *land utilization type code*, {LurId} is the *land use requirement code*, and {severity level} is the *severity level* of the corresponding land quality, which is an integer from 1 to the maximum severity level defined for the land use requirement of the land utilization type. Here is some sample output:

"AaA","wheat","e",3
"AaA","wheat","f",4
"AaA","wheat/soy","e",1
"BbB","wheat","e",4
"BbB","wheat","f",2
"BbB","wheat/soy","e",4

In this example, map unit 'AaA' has severity level 3 for land use requirement 'e' for land utilization type 'wheat', and so forth.

9.5.3.3 Yields

The yields output is one line per land unit/land use/output triple. The database *key* is the first three fields: land mapping unit code, land utilization type code, and output code. These are followed by two data fields, the first giving the proportional yield, from 0.00 to 1.00, and the second the predicted yield per unit area per crop. So the record format is:
{LmuId} {d} {LutId} {d} {OutId} {d} {pro yield} {d} {pred yield}

where {d} is the field *delimiter*, {LmuId} is the *map unit code*, {LutId} is the *land utilization type code*, {OutId} is the *output code*, {pro yield} is the *proportional yield*, as a fraction from 0.00 to 1.00, and {pred yield} is the *predicted yield*, as a non-negative number. The predicted yield's unit of measurement (numerator) and unit of area (denominator) are not written in this file; you must determine them from the evaluation model. Example are: kg/ha, cwt/acre, number/m^2, etc.

Here is an example output:
"AaA","agh","alfalfa",.8,8
"AaA","agh","hay",1,2000
"AaA","ww","wheat",.9,4.5
"BbB","agh","alfalfa",0,0
BbB","agh","hay",.8,1600
"BbB","ww","wheat",.6,3

In this example, map unit 'AaA' has a proportional yield of .8 for output 'alfalfa', corresponding to a per-unit area, per-crop yield of 8 output units per unit area, for land utilization type 'agh', and so forth.

9.5.3.4 Cash flow summary

The cash flow output is one line per land unit/land use/year in planning horizon triple. The database *key* is the first three fields: land mapping unit code, land utilization type code, and year number (1,2,...). These are followed by five data fields, giving the actual and discounted cash flows. The record format is:
{LmuId} {d} {LutId} {d} {YrId} {d} {cash in} {d} {cash out} {d} {PV in} {d} {PV out} {d} {NPV}

where {d} is the field *delimiter*, {LmuId} is the *map unit code*, {LutId} is the *land utilization type code*, {YrId} is the *year within the planning horizon*, {cash in} is the *actual cash in* to the LUT from sales of outputs, in the local currency, {cash out} is the *actual cash out* from the LUT for purchase of inputs, {PV in} and {PV out} are the discounted values of these at the beginning of the planning horizon, and {NPV} is the *net present value*, i.e. {PV in} – {PV out}.

Here is an example output:

"AaA","agh",1,187.17,156.28,183.5,156.28,27.22
"AaA","agh",2,249.99,335.2,240.27,328.61,-88.34

9.5.3.5 Spreadsheet format of the evaluation matrix

The *spreadsheet format* output resembles the evaluation results matrix:

√ The first (top) row contains the *LUT* codes

√ The first (leftmost) columns of each row after the first contain the *LMU* codes

√ The columns and rows after the first contain the results as they are displayed in the evaluation results matrix.

Any kind of result (physical or economic suitability classes, yields, land qualities, profitability) can be written in this format. Non-numeric results are enclosed in quotation marks. Files written with this option can be read directly by Quattro Pro with the "**T**ools | **I**mport | **C**omma delimited" command.

The record format is:

Row 1: {} {d} {$LutId_1$} {d} {$LutId_2$} {d} ... {$LutId_n$}
Row 2: {$LmuId_1$} {d} {$Result_{2,1,}$} {d} {$Result_{2,2,}$} {d} ... {$Result_{2,n,}$}
...
Row m: {$LmuId_{m-1}$} {d} {$Result_{m,1,}$} {d} {$Result_{m,2,}$} {d} ... {$Result_{m,n,}$}

where {d} is the field *delimiter*, {$LmuId_i$} is the *map unit code* for row *i+1*, {$LutId_j$} is the *land utilization type code* for column *j+1*, and {$Result_{i,j}$} is the evaluation result for map unit *i*, land utilization type *j*. Notice that the first row consists of spreadsheet *column headers*, and that the first column consists of spreadsheet *row headers.*

Here is an example output, for the yield of corn grain from model 'Tutor2':

" ","ccc","ccc-d"
"A3",102.168,114.968
"A8",85.158,99.456
"Ad",59.4,112

If read into a spreadsheet starting in cell A1 with, e.g., the "Tools | Import | Comma delimited" command of Quattro Pro, and after a bit of formatting in the spreadsheet, this data looks something like:

	A	B	C
1		ccc	ccc-d
2	A3	102.2	115.0
3	A8	85.2	99.5
4	Ad	59.4	112.0

Note that this looks exactly like the evaluation results matrix in ALES: the rows are map units and the columns are land uses. Now you can apply the usual spreadsheet manipulations to the data, e.g. calculate averages, sums, differences.

This chapter describes the interface between the ALES land evaluation program and the IDRISI Geographical Information System.

10. 1. What is ALIDRIS?

ALIDRIS is a set of subroutines written in the MUMPS programming language, along with support files in the MUMPS database sytem, which implements an interface from the ALES (Automated Land Evaluation System) land evaluation program, Version 3, to the IDRISI geographical information system (GIS), Versions 3 and 4. ALIDRIS automatically detects which version of IDRISI is installed on your system when it reads the IDRISI environment file 'idrisi.env'.

ALIDRIS only works as an optional module within ALES, it is not a stand-alone program. It comes installed as part of the ALES system and appears on the Menu 1 'Build Models and Evaluate' as item 7.

For more information on IDRISI, please contact:

The IDRISI Project
Graduate School of Geography
Clark University
950 Main Street
Worcester, MA 01610-1477 USA

Phone: USA (508) 793-7526
FAX: USA (508) 793-8842
Bitnet: idrisi@clarku
Internet: idrisi@vax.clarku.edu

Note: Cornell University does *not* endorse the IDRISI software over other similar programs. We wrote the ALIDRIS interface in response to requests by ALES users who also use IDRISI as their GIS. The two programs are very similar in philosophy (i.e., computing power is to be placed in the hands of the analyst at the local level), hardware requirements, and price.

For those with other GIS's, a similar interface can be obtained by using ASCII reports to disk file of evaluation results or database contents as the intermediate step.

The basic idea of ALIDRIS is that the ALES database or evaluation results are used to *reclassify* an existing IDRISI raster image that represents the *map units* in the ALES database. From one IDRISI *base map* which shows the ALES map units, ALIDRIS can make single-factor maps of any *land characteristic* in the ALES database, any *Land Quality* of any Land Utilization Type (LUT), or the final *evaluation results* (physical or economic) for any LUT, without the need for any further work in IDRISI. Then IDRISI can be used to display the derived maps, or apply any GIS operations on them (distances, adjacency operators, cross-tabulation etc.).

ALIDRIS can also be applied to IDRISI values files, commonly used to give legends to vector maps.

10.2. How to use ALIDRIS

The following discussion assumes that the reader is familiar with both ALES and IDRISI, both of which come with extensive documentation. Terms such as raster, documentation file, environment file, land characteristic, etc. are used as they are defined in the respective programs.

ALIDRIS appears as item 7 in the main menu of ALES. When you first select this menu item, you will be asked to specify the location of the IDRISI environment file on your computer. The environment file is used to locate the source and destination maps, exactly as in IDRISI. So before using ALIDRIS, make sure that the IDRISI environment is set to point to the images or value files you want to reclassify.

Once the environment file is located, ALIDRIS determines the IDRISI version number, and displays this at the top of its menu. This menu has four options:

```
              1.7 ALES -> IDRISI Interface [V4]

1. Reclassify an IDRISI raster image with LC values
2. Reclassify an IDRISI raster image with evaluation
results
3. Reclassify an IDRISI values file with LC values
4. Reclassify an IDRISI values file with evaluation
results
```

Options 1 & 2 are discussed in the next section, and 3 & 4 in the following.

10.2.1. ALIDRIS applied to raster images

The existing image must be in integer, ASCII format, with the integer codes representing the ALES map units. This image must have a documentation file with a legend giving the correspondence between the IDRISI integer codes and the ALES alphanumeric map unit codes. (If you decide to use the IDRISI codes as the ALES codes, the legend is not necessary, but the documentation file is.)

Before using ALIDRIS, you must prepare a source raster image, along with a documentation file, showing the area of interest. The legend, which can be added with DOCUMENT (in V4, also with COLOR option 'l'), must give the correspondence between the IDRISI codes and the ALES map unit identifiers. This map may then be used to produce many interpretive maps, both of land characteristics (from the ALES database) and land evaluation results (from the ALES evaluation results matrix).

Here is a typical V3 documentation file, with a legend giving the correspondence between IDRISI and ALES codes:

```
image title : Cayuga County Soil Map Units
data type   : integer
file type   : ascii
rows        : 50
columns     : 80
minimum     :   1.0000000000E+00
maximum     :   2.2000000000E+01
cell x      :   5.0000000000E+01
cell y      :   5.0000000000E+01
legend      : 3
category  1 : Ad
category  2 : AvE
category  3 : CeB
```

The V4 documentation file is more complex, but the information that ALIDRIS uses is the same, namely, the legend categories. Here is an example:

```
file title   : Cayuga County Soil Map Units
data type    : integer
file type    : ascii
columns      : 80
rows         : 50
ref. system  : plane
ref. units   : m
unit dist.   : 1.0000000
min. X       : 0.0000000
max. X       : 4000.0000000
min. Y       : 0.0000000
max. Y       : 2500.0000000
pos'n error  : unknown
resolution   : 50.0000000
min. value   : 1
max. value   : 3
value units  : classes
value error  : unknown
flag value   : none
flag def'n   : none
legend cats  : 4
category  0  :
category  1  : Ad
category  2  : AvE
category  3  : CeB
```

ALIDRIS reclassifies the image with either (1) land characteristic values from the database, or (2) evaluation results. In the latter case, the evaluation must already have been computed for the land utilization types and map units of interest (ALES Menu 1.3 Option 1). ALIDRIS writes a new, recoded map, and a documentation file, and, if the reclassification is to classified values, a legend. Real-valued results, including output yields, predicted monetary values, and the benefit/cost ratio, have no legend.

Reclassification with ALIDRIS produces raster maps that can be directly displayed with COLOR or used as a drape image in ORTHO without the need to manually enter a legend. Thus IDRISI acts as a graphics back end to ALES. However, the full power of IDRISI is also available to, for example, calculate areas or overlay with other coverages. So in this sense ALES acts as a front end, providing new coverages (showing, for example, values of land characteristics, or suitability for a given land use).

10.2.2. ALIDRIS applied to attribute value files

ALIDRIS can also reclassify V3 or V4 attribute values files. The attribute values file to be reclassified must contain the integer polygon identifiers (IDRISI 'feature identifiers') in the first column, and the ALES map unit ID codes (alphanumeric strings) in the second column. Any V3 documentation file for the values file is ignored, since such files are optional in IDIRISI V3. In IDRISI V4, a documentation file (.dvl) is required. It must specify that there are two free-format (format 0) fields, the first one (field 0) integer and the second (field 1) string, and in addition that the values file is ascii.

Here is a typical values file, giving the correspondence between IDRISI feature identifiers (e.g. vector polygons) and ALES codes:

```
1  Ad
2  AvE
3  CeB
```

ALIDRIS reclassifies this file as explained above, and writes a documentation file, with a legend if possible. Attribute files are most commonly used to display vector maps using the PLOT command, taking the graphic treatment from the new attributes file. The legend can be drawn directly from the attributes documentation file. Attribute values files can also be used with the ASSIGN command to reclassify a raster image, although this is a round-about way to create a map based on ALES map units (it's easier to reclassify the raster image directly with ALIDRIS).

3. Installing ALIDRIS

ALIDRIS comes as an optional module already compiled into ALES. It requires no installation. In addition, there are some test files which work with the ALES tutorials.

10.3.1 Files on the distribution disk

```
alidris.zip         ;compressed file containing the following:

     alidris.doc    ;this file, MS Word for Windows format
     alidris.txt    ;this file, plain-text format
     testim4.img    ;test image file for ALIDRIS options 1, 2
                    IDRISI V4)
     testim4.doc    ;documentation file for testim4
     testa4.val     ;test attribute value file for ALIDRIS options 3, 4
                    (IDRISI V4)
     testa4.dvl     ;documentation file for testa4
     testim3.img    ;test image file for ALIDRIS options 1, 2 (IDRISI
                    V3)
     testim3.doc    ;documentation file for testim3
     testa3.val     ;test attribute value file for ALIDRIS options 3, 4
                    (IDRISI V3)
     tutor5.img     ;image for ALES tutorial 5
     tutor5.doc     ;documentation file for tutor5
```

If the files have not already been uncompressed, after installing ALES you can run the command:

```
a:>unzip alidris.zip c:\dtm
```

assuming that you installed ALES to 'c:\dtm'; if not, substitute the drive and path where you actually installed ALES, also assuming that the installation diskette is in drive 'a:'.

The test files apply to ALES model 'Tutor2' (the second tutorial).

If you want to experiment with ALIDRIS using the test files provided on the installation disk, simply un-compress them to a directory on the hard disk as explained in the previous paragraph, and make sure that the IDRISI environment points to that directory (use the IDRISI **ENVIRON** command, if necessary). If necessary, load evaluation model 'Tutor2' (Tutorial 2) into ALES, using <F8> at the list of models. The file 'TUTOR2.ALS', supplied on one of your ALES distribution disks, contains this model. In ALES, select model 'Tutor2'. You can test ALIDRIS Options 1 and 3 right away, as these take land characteristics from the dat base. If you want to test Options 2 and 4, you will first have to compute an evaluation (Menu 1.3).

11. Maintaining an ALES system

This chapter provides general information on how to maintain the integrity of your ALES database and make modifications to your ALES environment. It is intended for the person functioning as *System Manager* for your PC. Typically this is the person who works with the operating system, do backups, add new hardware, and so forth.

11.1. Backup

As system manager, you are ultimately responsible for the integrity of the ALES knowledge bases and databases. The only way to ensure that power failures, media problems, or user error does not destroy all or part of an ALES evaluation is to institute a data integrity policy, also known as a *backup policy.*

The most important responsibility of a system manager is to to design a backup policy and enforce it!

There are two approaches to backup: *user-by-user*, or *system-wide*. These are described in the following sections.

It is important to distinguish between two sorts of problems: those affecting the *entire database*, and those affecting only *one evaluation*. The first can be caused by hardware problems such as a bad disk or controller, by software problems such as an error by the MUMPS software, or by a DOS user who deletes or alters one of the dataset files from DOS. The second sort of problem is caused exclusively by an ALES user inadvertently or mistakenly altering or deleting an evaluation from within ALES.

A problem with the user-by-user approach is that you are relying on the users to act in their own best interests. Although this is clearly desirable from their point of view, experience shows that many computer users don't think about backup until a disaster strikes. In this case, you, as system manager, might want to have acted to 'save them from themselves', and earn their heartfelt thanks for it! This is the motivation behind system-wide backup.

Another advantage to system-wide backup is that it is the first step in the database compression and repair procedure outlined below.

11.1.1. User-by-user backup : models and data

In the *user-by-user* approach, each user is responsible for making backup copies of their own evaluations. Both the model (knowledge base) and data can be backed up to the same DOS file. To do this, the user must be at the first ALES screen in 'model building' mode (i.e. when ALES is invoked from DOS with the 'ales' command). This screen shows the list of evaluations in the system. The user moves the cursor to highlight the evaluation to be saved, then presses the 'F7' key. ALES then asks for the name of the DOS file to which to write the evaluation. The default is to a file named '{eval}.ALS' in the same drive and directory as the ALES program, where {eval} is the evaluation code displayed on the left of the screen display. Typically the user will accept the default file name and extension, but add a drive designation in front of it (like, 'a:'), so that the backup copy will be written to a diskette. This is preferable to backing up an evaluation on the same hard disk that contains the original ALES database; a hard disk failure or reformatting won't erase both the ALES and DOS versions of the same evaluation.

(§7.3.8 presents the same information as in the previous paragraph, as part of the third tutorial example.)

> Note: you can set a *default drive and path* for saved evaluation models, with Menu 2 'System options', option 8, 'Set default path for saved evaluations'.

You will want to restore the saved evaluation if the on-line evaluation is accidentally deleted, or if some part of it is badly damaged by editing actions (e.g. in the decision tree editor). You can also use this kind of backup if for some reason ALES won't run at all. You can re-install ALES from your original distribution diskettes (as described in Chapter 5), and then restore all your evaluations from their individual backups.

To *restore* a saved evaluation (model and data), the user again works from the first ALES screen. The name of the evaluation to be restored must not be present in the list; if necessary, the incorrect version should be deleted, using the 'F4' key. Then the disk containing the saved evaluation is placed on-line, and the evaluation is restored by pressing the 'F8' key and naming the file to be loaded. This must have file type 'ALS'; the file name will be used as the evaluation name.

11.1.2. User-by-user backup : land unit definitions and data only

Once an evaluation model, i.e. an ALES knowledge base, is completed, it is not necessary to keep backing it up, as it doesn't change. However, the ALES data base associated with the model may well change, as more map units are added to the evaluation, and data is entered for them. It is possible to back up the land unit definitions and data only, without the knowledge base, using options 3 and 6 of the 'Data' menu (1.2). Note that this method of backup is also accessible to model users who enter ALES from DOS with the 'evaluate' command, as well as to model builders.

The output resulting from these two options is described in §9.5.1 (definitions) and §9.5.2 (data); here we describe the backup and restore process.

Backing up the list of map unit definitions is simple: select option 3 of the 'Data' menu (1.2), and specify a field delimiter and file name. The default delimiter of ';' (ASCII 59.) is fine; for the file give the drive name (typically a diskette, like 'a:') as well as a file name. When the form is confirmed, the definitions will be written to the selected file, and you will be informed of the number of definitions that were written.

Backing up the map unit data is more complicated. First you will have to define one or *more data entry templates*, using option 7 of menu 1.3 'Data'. If you already have all your LCs in one template you do not need to define another; otherwise, define a new template, e.g. with the name 'backup', and put *all* the LCs defined for the evaluation into the field list. You only need to define this template once, and then it is available for use each time you want to back up the data.

Once you have a template with all the LCs in it, select option 6 of the 'Data' menu (1.2), and when you are asked to select a template, pick the 'backup' template. After you've selected a template, you will be asked to specify a field delimiter and file name. The default delimiter of ';' (ASCII 59.) is fine; for the file give the drive name (typically a diskette, like 'a:') as well as a file name. When the form is confirmed, the data will be written to the selected file according to the backup template, and you will be informed of the number of (homogeneous) map units for which data was written.

To *restore* saved definitions or data, follow the above processes in reverse. Select menu 1.3 'Data' option 2 to restore map unit definitions, and option 5 to restore map unit data. In the second case you will be asked to select a template; pick.the same one you used to back up the data. In both cases you will be asked to specify a delimiter and file name; of course you will have the specify the same ones you used on output.

Before ALES reads definitions or data from the file, it shows you the first few lines of the file, and asks if you want to proceed. This allows you to peek at a file to make sure you have the right one. If you confirm, ALES will try to read the file; any errors will be reported at the bottom of the screen and also written to the log file. After the file is completely read, ALES will tell you how many records were read, and how many were in error. The log file will give details of any problem (you will have to exit ALES to read the log file with any text editor).

If the land unit definitions or data are already in the ALES database, they will be *overwritten* with the values read in from the file.

11.1.3. System-wide backup from MUMPS

In the *system-wide* approach, you use the MUMPS '%dsbackup' routine, described in §7.1.5 of the DataTree manual, to save the user datasets. The procedure is:

(1) From DOS, run the 'util' batch file, located in the same directory as the 'ales' batch file, by typing **util** at the DOS prompt. The MUMPS system program '%utility' appears, with a menu of system maintenance options.

(2) Move the cursor to Option 4, 'Database maintenance utilities', and press Return. A submenu of nine options will appear.

(3) Move the cursor to Option 3, 'back up dataset', and press Return. This will invoke '%dsbackup'.

(4) Select dataset 'EVAL-GBL' as the one to be backed up, by typing the number in front of this name in the list of datasets (this number is probably '4'), and pressing Return. The program will ask for an additional dataset; just press Return without entering anything. A list of parameters will appear, and the active question will be the name of the archive file.

(5) If you have a diskette drive 'a', the default archive file name 'A:DT' is fine. Otherwise, replace this with the name of the DOS file for the backup, without the extension ('%dsbackup' will automatically add the extensions '.01', .'02' etc. as needed). If at all possible, you should back up to a removable medium such as a diskette. When you have a valid name, complete the form by pressing Esc.

(6) You will be prompted to insert the first diskette; do so and press P to proceed.

(7) The dataset will be backed up to the disk you specified. If the dataset is too big to fit on a single diskette, %dsbackup will prompt for more diskettes as necessary.

(8) When the backup is done, you will be returned to the database options menu. Press End to return to DOS.

System-wide backup should be done daily, omitting days when there is no use of the ALES system. You should prepare at least three sets of floppy disks for backup: the 'current', 'father', and 'grandfather' sets. These are rotated, so that, for example, one day's 'grandfather' set becomes the next day's 'current set'. In this way, you can go two backups before discovering a problem, and still recover from it.

If the entire database becomes corrupted, for example by a hard disk problem, accidentally erasing the entire dataset file, or a MUMPS-induced database degradation error, you must *restore the entire database*, using the '%dsrestore' MUMPS routine, described in §7.1.6 of the DataTree manual. You invoke this in the same manner as '%dsbackup' (see above), except that this is Option 4, 'restore dataset', of the 'database maintenance utilities' submenu.

If the problem is that a user has erased or corrupted an evaluation, but the dataset is otherwise intact, the valid evaluations must be saved to floppy using the 'F7' method explained in the 'user-by-user' backup method (above), next '%dsrestore' is used to recover the dataset as it was last saved, and then the valid evaluations are restored with the 'F8' method. If the valid evaluations have not been altered since the most recent backup, it is not necessary to save and restore them with 'F7/F8'.

11.1.4. System-wide backup from DOS

Another approach to system-wide backup is to use the DOS 'backup' command, or a commercial backup product, including your MUMPS dataset files in the backup. This may be desirable if you already have a data integrity policy for your PC; then if you include the ALES datasets, they will be similarly protected.

The only files that need to be backed up are the datasets containing the user evaluations. As distributed, these files are 'c:\dtm\eval\eval-gbl.*'. There is no need to back up the MUMPS program, system utilities, or ALES program source and support files, as these do not change, and can be reloaded from the original distribution floppies should anything happen to them.

If you are doing *incremental* backups of DOS files, the ALES datasets will not be backed up. This is because their date of modification is not altered after they are created, so that backup software can't tell that they've changed.

11.2. Dataset compression

After a period of intensive use, MUMPS databases become *fragmented* in two ways. First, the expansion of the dataset creates DOS fragments on the hard disk. Second, the internal structures used by MUMPS within the dataset become spread out within the disk area available to them. Both of these instances of fragmentation lead to degraded performance and wasted disk space. Periodically, the system manager should compress the MUMPS databases used by ALES. The only database that should need compressing is that containing the user evaluations, i.e., 'EVAL-GBL'.

Compressing a dataset is accomplished by the use of the '%dscompress' MUMPS utility, described in §7.1.8 of the DataTree manual. This is invoked in the same manner as '%dsbackup' (see §10.1.3, above), except that this is Option 5, 'compress dataset', of the submenu. Once the compressed version of the dataset has been made, delete the original version, and then copy the compressed version to the correct location. For example, you could compress dataset 'EVAL-GBL' to a temporary location, e.g. another location in the directory hierarchy (not in C:\DTM\EVAL), by specifying a full path when asked for the file specification by the compression utility. Then these compressed files ('eval-gbl.dat' and 'eval-gbl.ind') could be copied to C:\DTM\EVAL\, over-writing the old (uncompressed) versions.

11.3. Dataset verification

In certain circumstances MUMPS may not shut down correctly, or the database may otherwise be corrupted. You can check for this by running the '%dsverify' utility on all datasets, as explained in the DataTree manual §7.1.7. This utility is invoked in the same manner as '%dsbackup' (see §10.1.3, above), except that this is Option 6, 'verify dataset integrity', of the submenu. If '%dsverify' reports an error (any message other than 'Dataset is OK', you will have to fix that dataset.

Your first approach should be to compress the dataset in question, as explained in the previous section. Then run '%dsverify' again on the fresh copy to see if the error has gone away. If not, run '%dscompress' in 'repair' mode, as explained in DataTree manual §7.1.9.2. Then run '%dsverify' again on the fresh copy to see if the error has gone away.

(Do *not* try to use '%dsrepair' (database maintenance utility submenu Option 9), this is only for those familiar with the internal structure of MUMPS databases, and its incorrect use can irretrievably damage a dataset.)

If the dataset is still corrupt, you will have to reinstall ALES from scratch. You can then re-load the evaluation models and data from user-by-user backup, if any (see §10.1.1). Or, if you have a valid system-wide backup of the 'EVAL-GBL' dataset, you can restore it as explained in §10.1.3, over-writing the 'EVAL-GBL' installed with the distribution.

11.4. Importing and exporting models

Models, including both the knowledge base and data base, can easily be exchanged between ALES systems. The model is *saved* to disk file as described under 'user-by-user backup', §10.1.1; the diskette is mailed or taken to the other ALES system, and then the model is *restored* to this system as described in §10.1.1.

11.5. MUMPS[65]

ALES is an application program running under the DataTree MUMPS database language and run-time environment. In turn, MUMPS is an application program running under your PC's operating system, PC-DOS or a functional equivalent. When you run ALES, you will be unaware of the underlying MUMPS. However, to do system management, you will have to have some understanding of the MUMPS system.

11.5.1. What you need to know about MUMPS

Along with each ALES license you have purchased from Cornell, you have also purchased a single-user license from the DataTree Division of InterSystems Corp. of Boston, Massachusetts, USA to use their MUMPS system. This ALES manual includes as an appendix the relevant portions of the DataTree MUMPS "Reference Manual and Operations Guide" (DataTree 1986) that you will need to do system management: i.e. chapters 2 and 7.

Although you will not be programming in the MUMPS language, you must understand the basic concepts of the MUMPS environment in order to be able to configure and manage ALES. You should read chapter 2 of the DTM-PC "Reference Manual and Operations Guide", supplied with this manual. It is not necessary to understand every detail in this chapter, but you should have some idea of the concepts, since they are referred to throughout the rest of this documentation.

11.5.2. Interacting with the MUMPS interpreter

When you run ALES, MUMPS is started in program execution mode. You can see the syntax for this in the batch file 'ales.bat': the appropriate DOS command is

```
mumps eval L /par=50k,8k,12k
```

The first field ('mumps') is the DOS command name, and the second field ('eval') is the *namespace* in which MUMPS will find its programs and data files. The third field ('L') is the *routine name*, in other words the name of the first module of the program to be run, in this case, 'L'. After these fields is a slash, '/', and then some program options. If there is a third field listed before the slash '/', MUMPS will look for the named program and run it. When the program is completed, MUMPS will halt, and you will again see the DOS prompt. This is the way ALES is set up to run.)The portion of the command line after the forward slash '/' is a list of startup parameters.)

Another batch file, 'util.bat', has been set up to allow you to perform common system maintenance tasks. It consists of a single line:

```
mumps sys %%utility /par=32k,16k,16k
```

---

[65]In 1993, the ANSI language MUMPS officially changed its name to M, and the trade association changed its name from the MUMPS User's Group (MUG) to the M Technology Association (MTA). However, the name MUMPS is still more common than M.

(The two '%' are required to fool the batch processor, which interprets a single '%' as the introduction to a batch file variable name. If you type this command directly to the DOS prompt, you only need one '%'.) To run system maintenance utilities, just type

```
util
```

from the DOS prompt.

To exit from the system maintenance utility program, press the 'END' key

The other way to enter MUMPS is in *direct mode*. You will probably never want to do this, but in case you do, the following paragraphs explain how. In direct mode, you are conversing with an interpretive shell. You type MUMPS programming language commands to the shell, and it immediately responds by executing these commands. The command for entering direct mode in the System namespace is:

```
mumps sys /par=32k,16k,16k
```

Note there is no third field before the slash '/', so MUMPS doesn't start a program. Instead it displays the MUMPS shell prompt, a single left carat ('greater than' sign, i.e. >).

There are many MUMPS commands, but only two that you need to know in order to interact with the MUMPS shell for system maintenance purposes: 'halt' and 'do'; these can be abbreviated to just 'h' and 'd', respectively. All commands are terminated by pressing the **Return** key.

The **halt** command is used to leave the interpretive shell and return to DOS. At the '>' prompt, type:

```
halt
```

and then press **Return**.

To exit from the MUMPS interpreter, type 'halt' and then press 'Return'

The **do** command is used to execute programs. For example, to run the dataset compression utility program from the shell, type `do ^%dscompress` and then press **Return**.

The MUMPS subroutine name (e.g. '%dscompress' in the previous example) must be preceded by the up-arrow or carat '^'. When ALES is finished, the shell will regain control.

11.5.3. Programming in MUMPS

The MUMPS language, which was used to write the ALES software, is a procedural language with a built-in database manager. It was originally developed at the Massachusetts General Hospital in the early 1970s, as a minicomputer language to replace mainframe data processing languages like COBOL. Linguistically, it is a hybrid between LISP and BASIC, with the added feature of sparse, hierarchical, disk-based arrays as the main data structure. The language is fairly widely used in transaction-oriented data processing applications in medicine, libraries, and business. With the emergence of modern 4th-generation database languages, and linguistically-superior procedural languages, MUMPS will probably never be a major programming language, although the MUMPS Development Committee (affiliation with ANSI and ISO) is working a new version of the standard language, and the M Technology Association is actively promoting M (née MUMPS).

If you would like to program in MUMPS, you don't need any additional software, since you have already purchased the required license from DataTree, and the MUMPS programming language is included with the ALES software. However, you will want the complete DataTree user's manual. Write to DataTree Division, InterSystems Corp., One Memorial Drive, Cambridge, MA 02142 USA, or phone them at USA (617) 374-9191, FAX (617) 494-1631, for the price and terms for this manual. A very good *text* on MUMPS is "The Complete MUMPS: An Introduction and Reference Manual for the MUMPS Programming Language", by John M. Lewkowicz of Cornell University, published by Prentice-Hall, Englewood Cliffs, NJ in 1989. You can also obtain tutorials and general information on the language from the M Technology Association, 4321 Hartwick Road, Suite 510, College Park, MD 20740 USA.

We do not recommend attempting to alter the ALES program, so we do not include the program source code with the ALES distribution. You should be able to use the framework provided by the program to build models to suit your needs; in this sense the ALES program provides a high-level "programming language" of its own. If you still feel the urge to alter the ALES program, you will need the source code and accompanying internal program documentation. Write to us and tell us what you have in mind.

11.6. When bad things happen to good people

It is impossible in this manual to cover every problem that you may encounter. If you have previous experience with computer systems, you know the variety of things that can and do go wrong. Perhaps the best advice we can give is to *never compound a problem by making hasty decisions*. Once you are aware of a problem, make sure you *understand its cause* before attempting any corrective action. Then map out a strategy to correct the problem. If you are going to be overwriting any files, it is wise to first make backup copies of them, even if you don't think you'll need them. You always want to be able to recover to at least the point you're at before you start a fix-up process.

You should always be in a situation where, if all else fails, you can *re-install ALES from scratch*, i.e. from the original distribution diskettes, and then re-load your evaluations and data into it.

12 Bibliography

(See also the annotated bibliography of land evaluation in Chapter 3.)

Beatty, M. T., G. W. Petersen, and L. D. Swindale (eds.). 1979. *Planning the uses and management of land.* Agronomy Monograph 21, American Society of Agronomy, Madison, WI, USA.

Beek, K. J. 1978. *Land evaluation for agricultural development.* International Inst. for Land Reclamation and Improvement, Wageningen, publication 23.

Bouma, J. & H. A. J. van Lanen. 1987. *Transfer functions and threshold values: from soil characteristics to land qualities.* pp. 106-110 in: Beek, K.J., P.A. Burrough, and D.E. McCormack (ed.). 1987. "Quantified land evaluation procedures: proceedings of the international workshop on quantified land evaluation procedures, held in Washington, DC 27 April - 2 May 1986". International Institute for Aerospace Survey and Earth Sciences (ITC) publication No. 6, Enschede, The Netherlands.

Burrough, P.A. 1986. *Principles of geographical information systems for land resources assessment.* Monographs on Soil and Resources Survey No. 12, Oxford Science Publishers, New York.

Cornell University. 1989. *1989 Cornell Recommends for field crops.* New York State College of Agriculture and Life Sciences, Cooperative Extension, Ithaca, NY.

Cox, W.J. & S.D. Klausner (eds.). 1987. *Cornell Field Crops and Soils Handbook, 2nd ed.* New York State College of Agriculture and Life Sciences, Cooperative Extension, Ithaca, NY.

DataTree, Inc. 1986. *DTM-PC reference manual and operations guide.* DataTree Inc., Waltham, MA.

Davidson, D. A. 1980. *Soils and land use planning.* Longman's, London.

Davidson, D. A. 1992. *The evaluation of land resources.* Longman Scientific, London.

Davidson, D. A. (ed.). 1986. *Land evaluation.* Van Nostrand Reinhold/AVI

Dent, D. & A. Young. 1981. *Soil survey and land evaluation.* George Allen and Unwin, London, England.

Food and Agriculture Organization of the United Nations. 1976. *A framework for land evaluation.* Soils Bulletin 32, FAO, Rome, Italy.

Food and Agriculture Organization of the United Nations. 1983. *Guidelines: land evaluation for rainfed agriculture.* Soils Bulletin 52, FAO, Rome, Italy.

Food and Agriculture Organization of the United Nations. 1984. *Land evaluation for forestry.* Forestry Paper 48, FAO, Rome, Italy.

Food and Agriculture Organization of the United Nations. 1985. *Guidelines: land evaluation for irrigated agriculture.* Soils Bulletin 55, FAO, Rome, Italy.

Hudson, N. 1981. *Soil conservation, 2nd ed.* Cornell Univ. Press, Ithaca, NY.

Hutton, Frank Z. Jr. 1971. *Soil survey of Cayuga County, New York.* US Government Printing Office, Washington, DC.

van Lanen, H. A. J., Hack-ten Broeke, M. J. D. , Bouma, J., & de Groot, W.J.M. 1992. *A mixed qualitative/quantitative physical land evaluation methodology.* **Geoderma 55**: 37.

van Lanen, H. A. J. and H. Wopereis. 1992. *Computer-captured expert knowledge to evaluate possibilities for injection of slurry from animal manure in the Netherlands.* **Geoderma 54**: 107.

León Pérez, J. C. 1992. *Aplicación del Sistema Automatizado para la Evaluación de Tierras (ALES), en un sector de la cuenca del río Sinú (Córdoba, Colombia).* **Revista C.I.A.F. 13**(1): 19-42.

Lewkowicz, J. 1988. *The Complete MUMPS.* Prentice-Hall, Englewood Cliffs, NJ USA

MUMPS Users Group. 1985. *ANS MUMPS programmers' reference manual.* MUMPS Users Group, College Park, MD.

Purnell, M. 1987. *Quantitative land evaluation in some FAO programs.* pp. 7-15 in: Beek, K.J., P.A. Burrough, and D.E. McCormack (ed.). 1987. "Quantified land evaluation procedures: proceedings of the international workshop on quantified land evaluation procedures, held in Washington, DC 27 April - 2 May 1986". International Institute for Aerospace Survey and Earth Sciences (ITC) publication No. 6, Enschede, The Netherlands.

Riquier, J.. 1974. *A summary of parametric methods of soil and land evaluation..* p. 47-53. In: Food and Agriculture Organization of the United Nations, Soils Bulletin 22, FAO, Rome, Italy.

Rossiter, D.G. 1988a. *The Automated Land Evaluation System: a microcomputer program to assist in land evaluation.* 1988. Ph.D. disseration, Cornell University. University Microfilms, Ann Arbor, MI.

Rossiter, D.G. 1988b. *ALES: a microcomputer program to assist in land evaluation.* Proc. International Workshop on Land Qualities in Space and Time, Wageningen, the Netherlands, 22-26 August 1988, Pudoc, Wageningen.

Rossiter, DG. 1990. *ALES: A framework for land evaluation using a microcomputer.* **Soil Use & Management** 6(1): 7-20.

Rossiter, D. G., M. Tolomeo, & A. R. Van Wambeke, 1988. *Automated Land Evaluation System Version 1.0 User's Manual.* Department of Agronomy, Cornell University, Ithaca, NY.

Rossiter, D. G. and A. R. Van Wambeke. 1991. *Automated Land Evaluation System (ALES) Version 3 User's Manual*. SCAS Teaching Series 2. Ithaca, NY, Cornell University, Department of Soil, Crop & Atmospheric Sciences.

Sanchez, P.A. , W. Couto, & S.W. Buol. 1982. *The fertility capability soil classification system: interpretation, applicability, and modification*. **Geoderma 27**(4): 283-309.

Siderius, W. (ed.). 1984. *Proceedings of the workshop on land evaluation for extensive grazing* (LEEG). International Inst. for Land Reclamation and Improvement Publ. 36, Wageningen, The Netherlands.

Siderius, W. (ed.). 1986. *Land evaluation for land-use planning and conservation in sloping areas*.. International Inst. for Land Reclamation and Improvement Publ. 40, Wageningen, The Netherlands.

Snyder, D.P., & W.F. Lazarus. 1987. *Field crop enterprise budgets: 1987 projections and grower worksheets*. New York State. Mimeo A.E. Res 87-8. Department of Agricultural Economics, Cornell University, Ithaca, NY.

Snyder, D.P. 1988. *Field crop enterprise budget update: 1988 cost and return projections and grower worksheets*. New York State. Mimeo A.E. Res 88-6. Department of Agricultural Economics, Cornell University, Ithaca, NY.

Sys, C. 1985. *Land evaluation* (Parts I, II, III, and Appendices). Publications Agricoles 7, Administration Générale de la Coopération au Développement, Brussels, Belgium.

Wood, S.R., & F.J. Dent. 1983. *LECS: a land evaluation computer system*. Ministry of Agriculture, Government of Indonesia, publications AGOF/INS/78/006/Manual 5 (Methodology) and 6 (User's Manual), Bogor, Indonesia.

Appendix 1 - ALES program structure

This appendix shows the structure of the ALES program, from the model builder's point of view (DOS command 'ales.bat'), by means of a nested key. The various types of interactions are:

Menu: choose one of the numbered action items
Choose: add to, delete from, edit in, or copy in a list; new items typed in
List: add to, delete from, edit in a list; new items selected from another list
Ordered list: a list where the items are numbered
Select: pick exactly one item from a list
Set: pick any number of items from a list
Form: type in answers to one or more questions all pertaining to the same thing
Enter: type in an answer to one question
Decision tree editor

Opening screen: Copyright, licensee name, and serial number

Main **Menu** - 'ALES [Build models]'

- 1 Build models & evaluate
  Choose: evaluation model [may also save to (**F7**), restore from (**F8**) disk file]
  Menu 1 - 'Main Options [Build Models]'
  - 1 Reference Lists: **Menu** 1.1 'Reference Lists'
    - 1 Land Use Requirement reference list
      - **Choose**: land use requirement code
        - **Form**: descriptive name
    - 2 Output reference list
      - **Choose**: output code
        - **Form**: descriptive name, units of measurement, price per unit
    - 3 Input reference list
      - **Choose**: input code
        - **Form**: descriptive name, units of measurement, cost per unit
    - 4 Land Characteristic descriptions
      - **Choose**: land characteristic code
        - [If a new LC]
          - **Enter**: land characteristic code
          - **Form**: descriptive name, number of classes, units of measurement
        - [Case: LC has classes and units of measurement]
          - **Menu** 1.1.4a - Specification options for Discrete LC (with units)
            - 1 Name, units
              - **Form**: descriptive name, units of measurement
            - 2 class Abbreviations
              - **Form**: abbreviations for each class
            - 3 class Names
              - **Form**: descriptive names for each class
            - 4 class Limits
              - **Form**: lower limit of class 1, upper limit of each class

5 Lc -> Lc decision tree

Decision Tree Editor

6 infer from commensurate LC

Select: infer from which commensurate LC?

[Case: LC has classes but no units of measurement]

Menu 1.1.4b - Specification options for Discrete LC (no units)

1 Name

Form: descriptive name

2 class Abbreviations

Form: abbreviations for each class

3 class Names

Form: descriptive names for each class

4 Lc -> Lc decision tree

Decision Tree Editor

[Case: LC has no classes]

Menu 1.1.4c - Specification options for Continuous LC

1 Name, units

Form: descriptive name, units of measurement

2 Data validation limits

Form: lowest, highest values

3 Infer by a formula

Enter: formula

---

2 Land Utilization Types

Choose: land utilization type code [may also save to (**F7**), restore from (**F8**) disk file]

Menu 1.2 : 'Specification options for LUT'

1 Name

Form: descriptive name

2 Length

[Case - planning horizon is >1 year]

Menu 1.2.2 - Length of planning horizon for LUT

1. Add a year (lengthen)

Select: add a year where?

2. Delete a year (shorten)

Enter: delete which year?

[Case - planning horizon is exactly 1 year]

Select: add a year where?

3. Economic parameters

Menu 1.1.5.3 - Economic parameters for LUT

1 Discount Rate

Form: discount rate

2 Economic class limits - Gross Margin analysis

Form: lower limits for suitability classes s1, s2, and s3

3 Economic class limits - Discounted Cash Flow analysis

Form: lower limits for suitability classes s1, s2, and s3

4 Inputs (annual)

List: which input?

Form: number of units per unit area per year

5 Inputs (by year)

Choose: which year?
List: which input?
Form: number of units per unit area this year
6 Outputs, including Proportional Yields
List: output
Menu 1.2.6 'Specification options for LUT, Output'
1 Optimum yield, years when harvested
Form: Optimum yield, years when harvested
2 proportional yield decision tree
Decision Tree Editor
3 Proportional Yield Factors
Select: proportional yield factors for which LUR?
Form: proportional yield factors for each level
4 LimitingYield Factors
Select: limiting yield factors for which LUR?
Form: limiting yield factors for each level
5 Production-dependent inputs
List: which input?
Form: Number of units of the selected input per unit of production of this output
7 Land Use Requirements
List: land use requirements
Menu 1.2.7 'Specification options for LUT, LUR'
1 severity level names
Form: name for each level
2 additional Annual inputs
List: annual inputs
Form: number of units of the input for each severity level
3 additional by-year inputs
List: which year?
List: by-year inputs
Form: number of units of the input for each severity level
4 Severity level decision tree
Decision Tree Editor
5 Proportional Yield Factors
Select: proportional yield factors for which output?
Form: proportional yield factors for each level
6 LimitingYield Factors
Select: limiting yield factors for which output?
Form: limiting yield factors for each level
7 Maximum limitation
Enter: include this LUR in the list of maximally-limiting factors?
8 Physical Suitability Subclasses
Menu 1.2.8 'Physical Suitability options for LUT'
1 Enter/edit physical suitability subclass decision tree
Decision Tree Editor
2 Choose factors for maximum limitation
Set: Consider which LURs in the maximum limitation?
3 Enter/edit names or notes for physical suitability subclasses

Choose: physical suitability subclass code

Form: descriptive name for PSSC

9 land units Not Rated

List: map units not to be rated

---

3 Data: **Menu** 1.3

1 Definitions, from keyboard, enter or edit

Choose: land mapping unit code

Form: name, hectares, homogeneous/compound

[If map unit is compound]

List: constituent homogeneous map units

Form: proportion of each constituent

2 Definitions, from a disk file, Read

Form: field delimiter, input file, log file

3 Definitions, to a disk file, Write

Form: field delimiter, output file

4 Data, from keyboard, enter or edit

Select: data entry template

Select: land mapping unit

Form: data for land mapping unit

5 Data, from a disk file, Read

Form: field delimiter, input file, log file

6 Data, to a disk file, Write

Form: field delimiter, output file

7 Templates

Choose data entry template

Menu 1.3.7 Specification options for data entry template

1 Descriptive name

Form: descriptive name

2 List of data fields (land characteristics)

Ordered list: data fields

8 Erase the data base

9 Infer land characteristic values

Set: infer values of which continuous LCs?

---

4 Evaluations: **Menu** 1.4

1 Compute an evaluation

Set: Which land utilization types?

Set: Which land mapping units?

2 View evaluation results

Select: What kind of report?

[Case: one Land Quality]

Select: Land Quality to report

[Case: Yield]

Select: output to report

Display evaluation matrix

F2: Why?

F3: Recompute

F4: Edit data for map unit in current row (Option 4 of Menu 1.2)

F5: Edit specification of land utilization type in current column (Menu 1.1.5)

3 Print evaluation results
  Select: Which kind of output device?
    Select: What kind of report?
      [Case: summary report]
        (report is printed with no further questions)
      [Case: not a summary report]
        [Case: one Land Quality]
          Select: Land Quality to report
        [Case: Yield]
          Select: output to report
        Select: Sort by Rows (LMU) or Columns (LUT)?

4 File evaluation results
  Menu 1.4.4 : Output to a disk file
    1 Suitability Matrix
      Form: output file name, field delimiter
    2 Severity Levels
      Form: output file name, field delimiter
    3 Output Yields
      Form: output file name, field delimiter
    4 Cash Flow Summary by year
      Form: output file name, field delimiter
    5 Spreadsheet
      Select: What kind of report?
      Form: output file name, field delimiter

5 Save current evaluation results
  Enter: Description for saved matrix

6 Delete evaluation results
  Set: Delete which saved evaluation matrices?

---

5 Reports: **Menu** 1.5
  1 Evaluation results
    (same as 1.3.3, see above)
  2 Land Utilization Types
    Set: Print which land utilization types?
  3 Land Characteristics
    Set: Print which land characteristics?
  4 Land Unit Definitions
    Set: Print definitions of which land units?
  5 Land Unit Data
    Set: Print data for which land units?
  6 data entry Templates
    Set: Print which templates?
  7 Inputs and Outputs
    Set: Print which inputs and outputs?
  8 Land Use Requirement codes
    Set: Print which LUR codes?
  9 Decision Trees

Set: Print which decision trees?
A Notes
Set: Print which notes?

---

6 Consult
Select: evaluate for which LUT?
Select: start with data from a map unit in the database?
Yes
Select: map unit
Consultation
F2 Why?
F3 Reconsult
F10 Done with consultation
Select: save the data from this consultation?
Yes
Enter: map unit code
Enter: descriptive name

---

7 ALIDRIS
Enter: path to IDRISI program [first time only]
Menu 1.7 'ALES --> IDRISI interface [V4]'
1 Reclassify an IDRISI raster image with LC values
Enter: name of the source image
Select: land characteristic to map
Enter: name for the new IDRISI map
2 Reclassify an IDRISI raster image with evaluation results
Enter: name of the source image
Select: Land Utilization Type to map
Select: Type of evaluation result to map
Enter: name for the new IDRISI map
3 Reclassify an IDRISI values file with LC values
Enter: name of source values file
Select: land characteristic to map
Enter: name for the new IDRISI values file
4 Reclassify an IDRISI values file with evaluation results
Enter: name of source values file
Select: Land Utilization Type to map
Select: Type of evaluation result to map
Enter: name for the new IDRISI values file

---

2 System Options:
Menu 2 - System Options

---

1 Change colors for main screen
Select: color scheme

---

2 Change colors for banner, function keys
Select: color scheme

---

3 Change colors for help screens
Select: color scheme

---

4 Change colors for consultation mode
Select: color scheme

5 Change colors for annotations
Select: color scheme

6 Change display language
Select: language

7 Change numeric format
Select: American or European?

8 Change default path for saved evaluations
Enter: path

9 Change default path for database import & export
Enter: path

A Change default path for log files
Enter: path

B Change default path for reports to a disk file
Enter: path

C Change model group
Enter: model group code

3 About ALES
Menu 3- About ALES

1 Author & Copyright
2 How to contact the ALES project
3 License
4 Quick help

Appendix 2: ALES knowledge base schema

--1 Evaluations
-- --2 Land Use Requirements
-- -- -- descriptive name
-- --2 Inputs
-- -- -- units of measurement, purchase cost
-- --2 Outputs
-- -- -- units of measurement, selling price
-- --2 Land Characteristics
-- -- --*Case:* discrete land characteristic
-- -- -- -- number of classes, units of measurement, +class limits
-- -- -- -- LC-to-LC decision tree
-- -- -- -- commensurate LC from which to infer
-- -- --*Case:* continuous land characteristic
-- -- -- -- units of measurement, upper and lower limits
-- -- -- -- formulas (from a group of continuous LCs to this LC)
-- --2 Land Utilization Types
-- -- -- length of planning horizon, discount rate
-- -- -- number of physical suitability classes
-- -- -- physical suitability subclass decision tree
-- -- -- physical suitability subclass names
-- -- -- economic suitability class limits
-- -- --3 recurrent annual Inputs
-- -- -- -- annual amount of input
-- -- --3 Year in planning horizon
-- -- -- --4 by-year Inputs
-- -- -- -- -- amount of input this year
-- -- --3 Land Use Requirements
-- -- -- -- number of severity levels
-- -- -- -- severity level decision tree
-- -- -- --4 Severity Levels
-- -- -- -- ---- severity level names
-- -- -- -- --5 Inputs
-- -- -- -- ---- additional annual amount
-- -- -- -- ---- additional one-time amount
-- -- --3 Outputs
-- -- -- -- harvest years
-- -- -- -- optimum yield
-- -- -- -- proportional yield decision tree
-- -- -- -- limiting yield factors:
-- -- -- -- --4 Land Use Requirements
-- -- -- -- ---- 5 Severity Levels
-- -- -- -- ---- -- limiting yield factor
-- -- -- --- multiplicative yield factors:
-- -- -- -- --4 Land Use Requirements
-- -- -- -- ---- 5 Severity Levels
-- -- -- -- ---- -- limiting yield factor
-- -- -- -- ---- -- multiplicative yield factor
-- -- -- -- --4 production-related Inputs
-- -- -- -- ---- input level per unit of output

Underlined entities are their primary definition; Numered entries may be repeated.

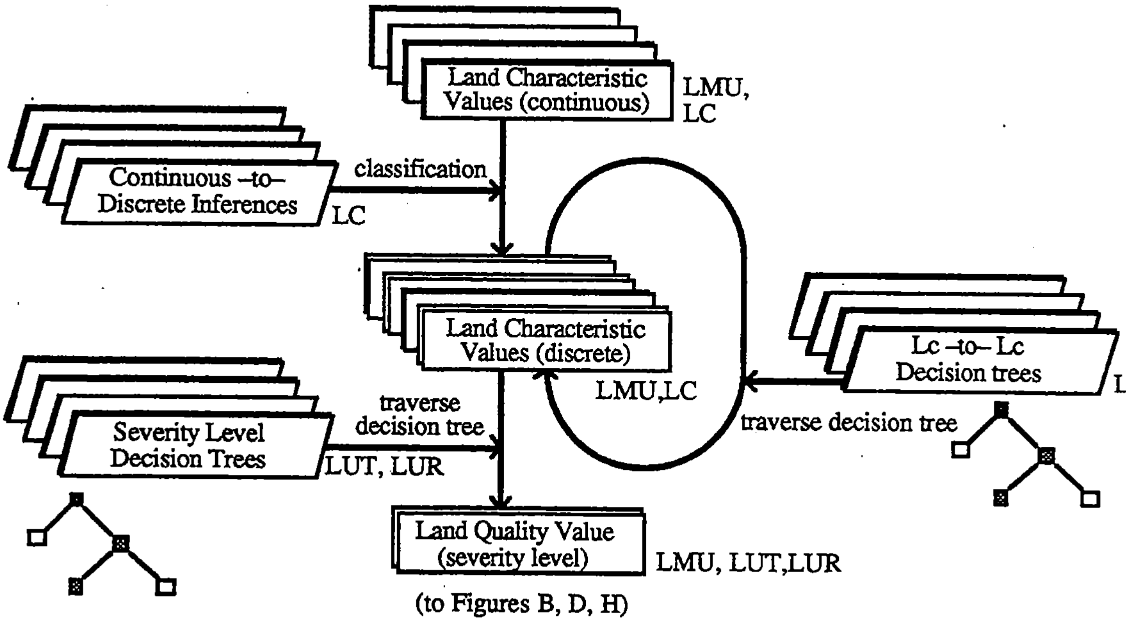


Figure A : Computation of a land quality value

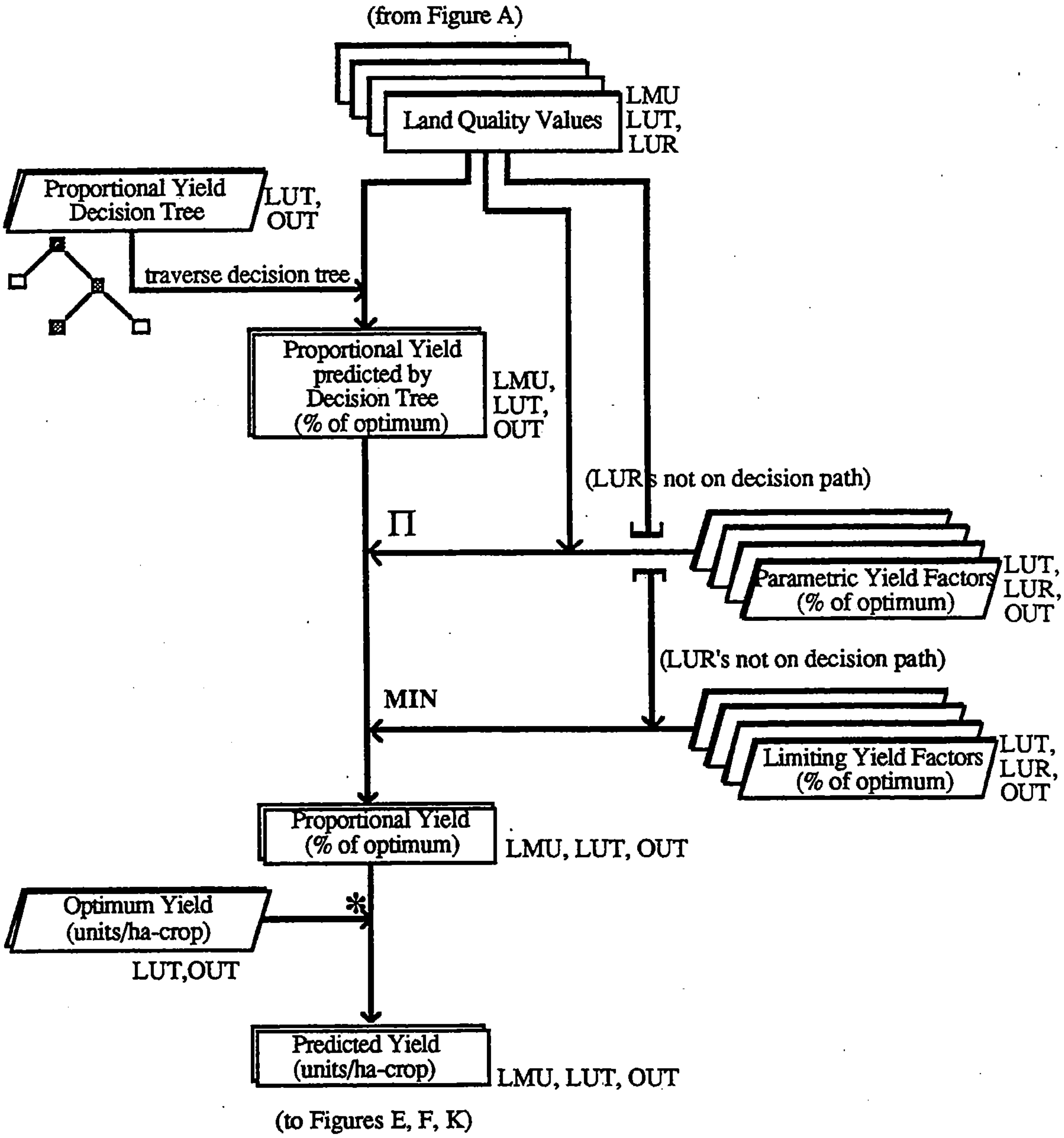


Figure B : Computation of a predicted yield

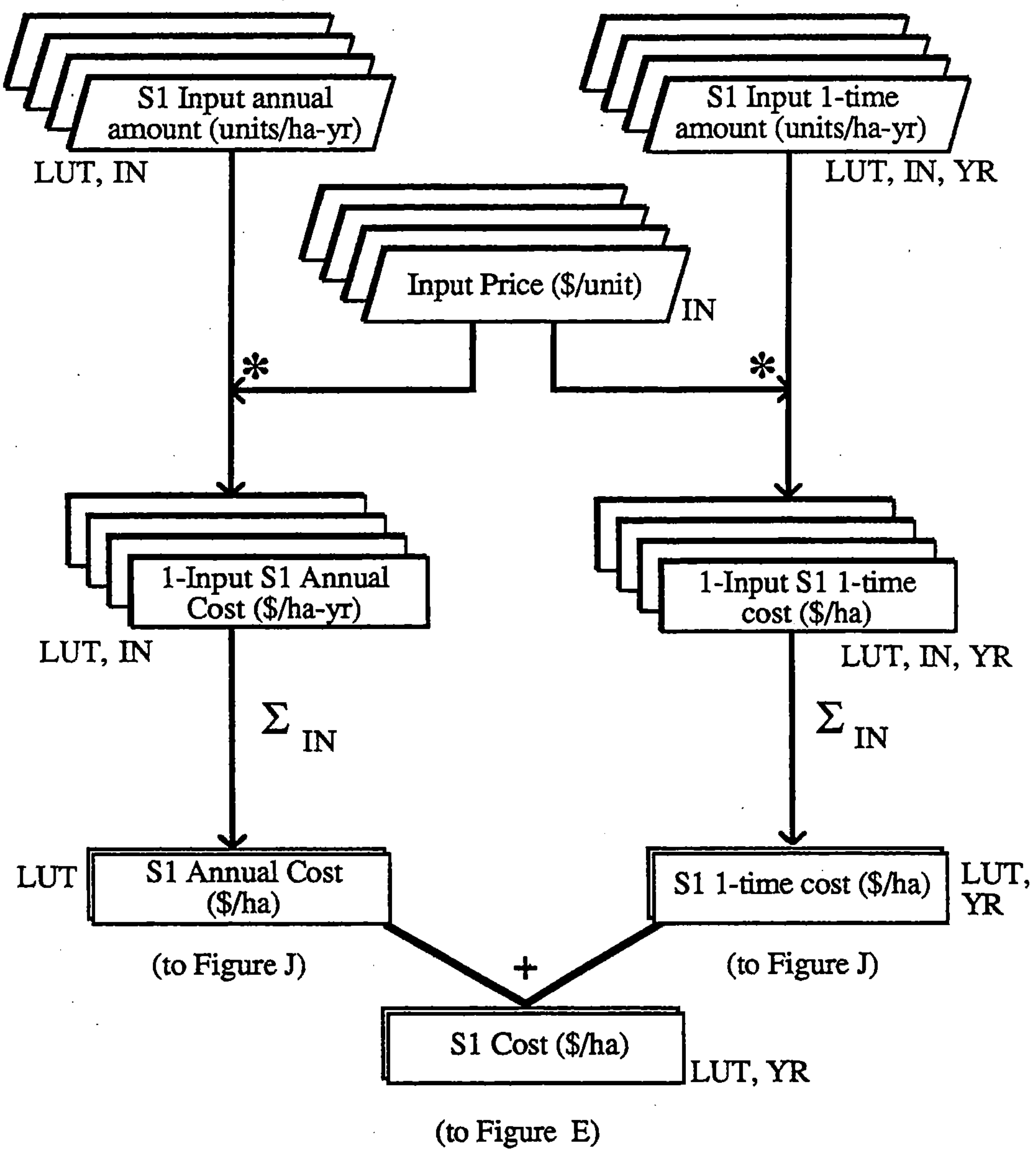


Figure C : Computation of yearly costs that do not vary with severity levels of land qualities

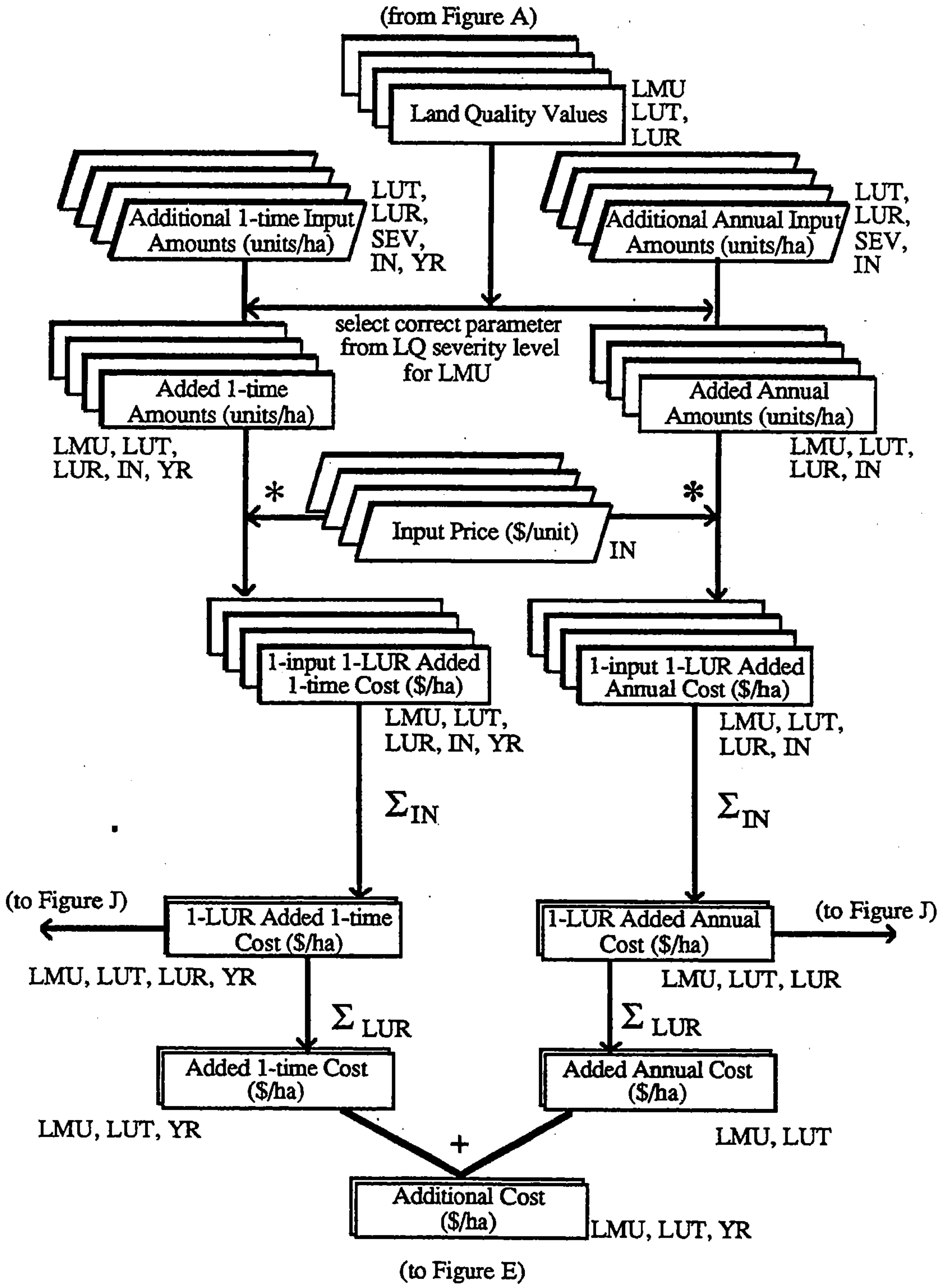


Fig. D. Computation of yearly costs that vary with severity levels of land qualities

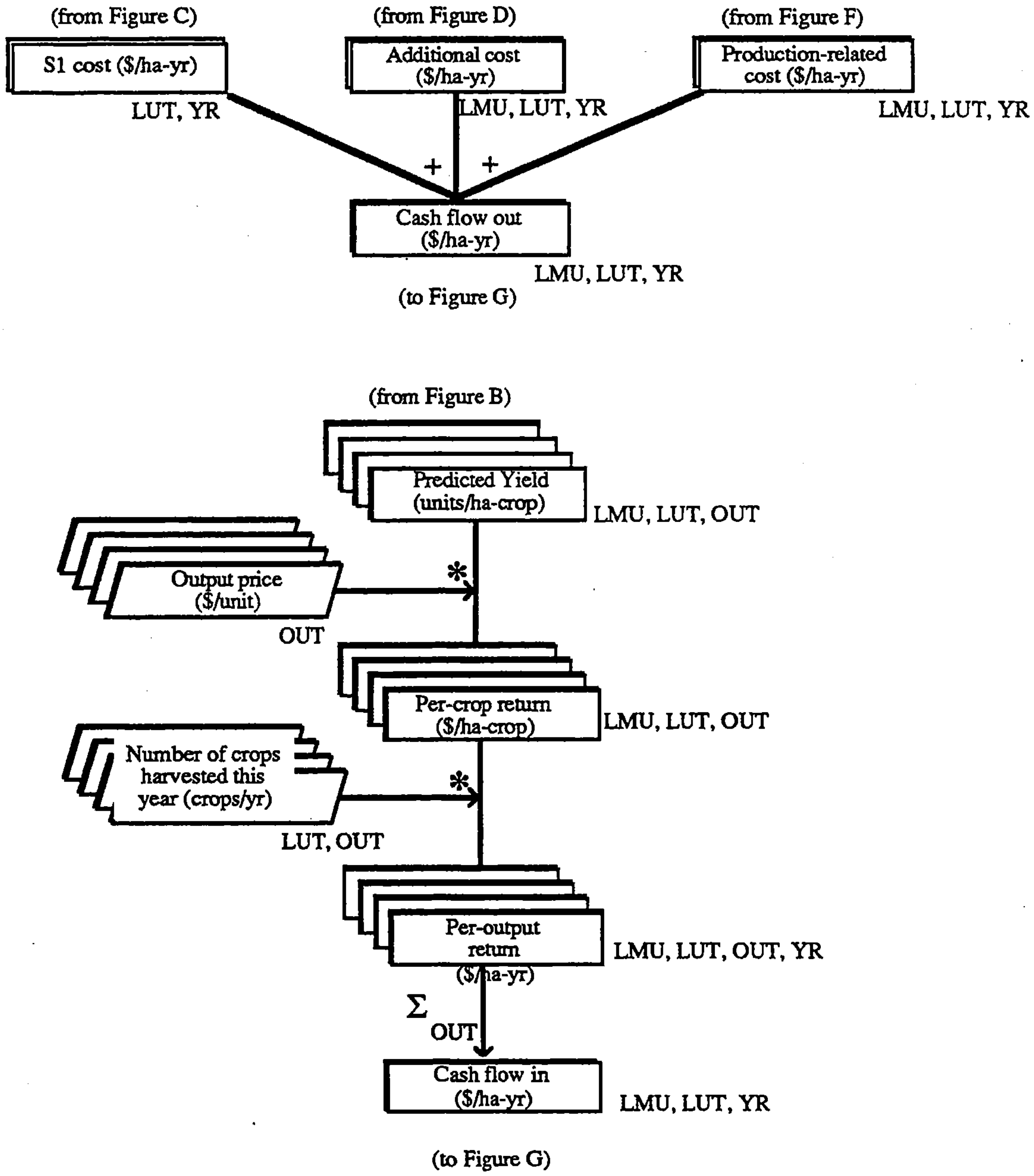


Figure E : Computation yearly cash flows in and out

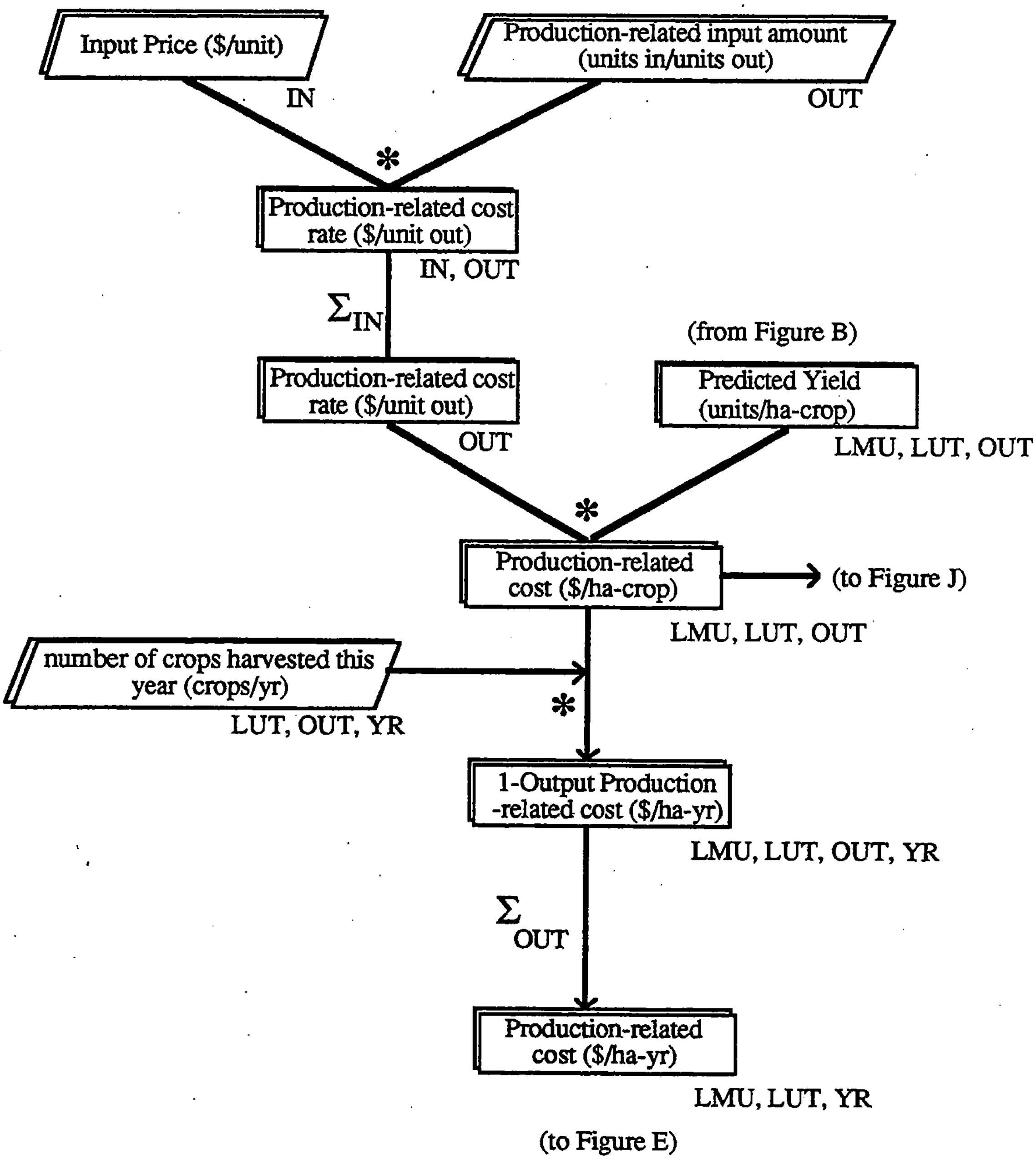


Figure F : Computation of yearly production-related costs

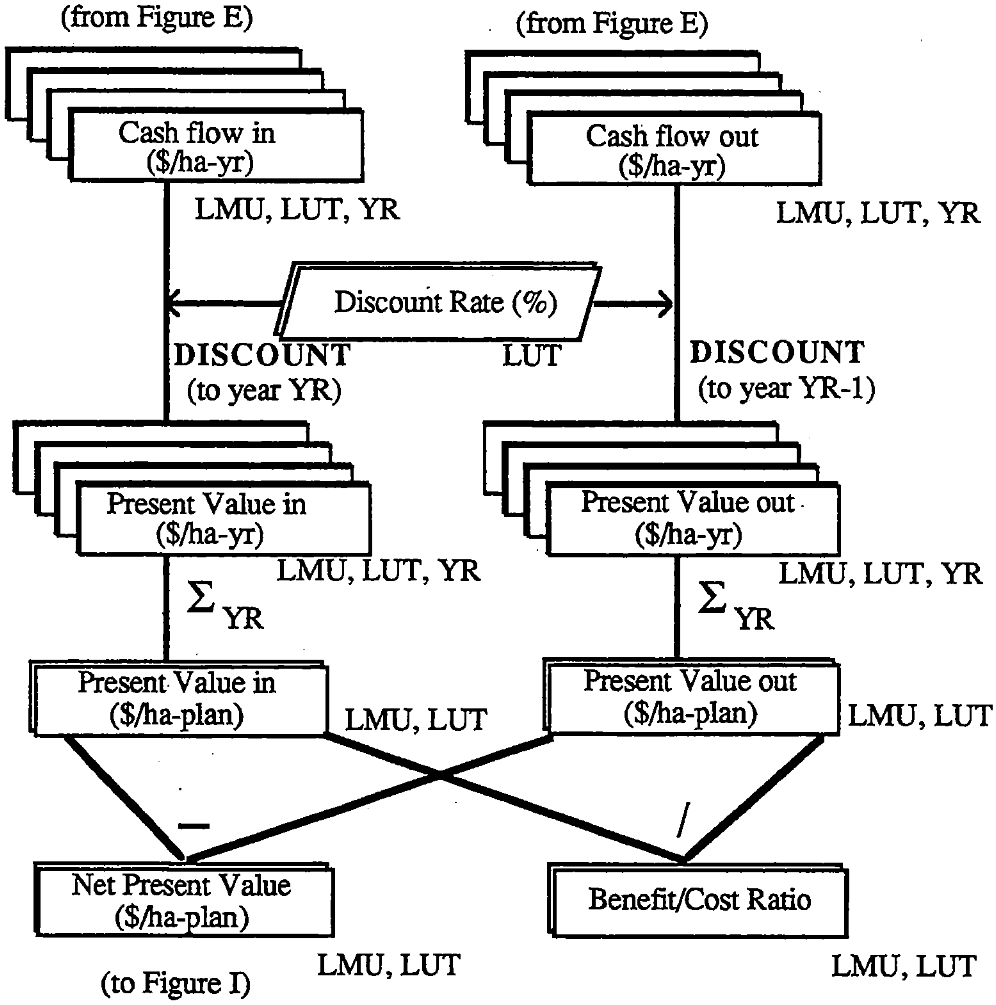


Figure G : Computation of net present value and benefit/cost ratio

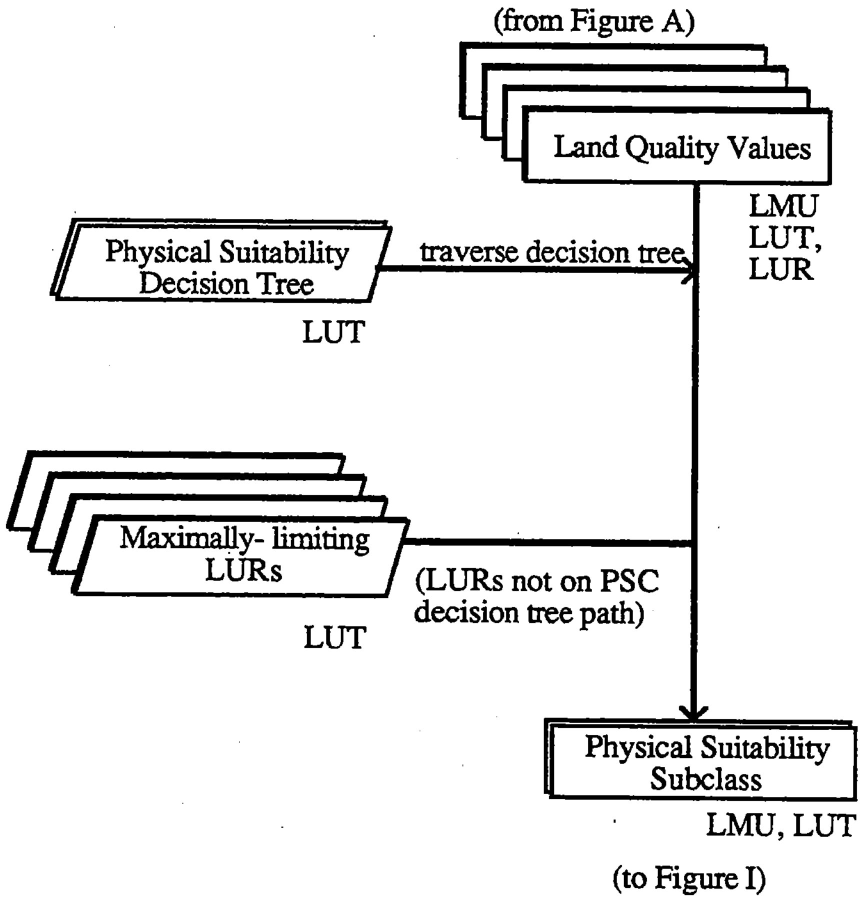


Fig. H. Computation of physical suitability classes

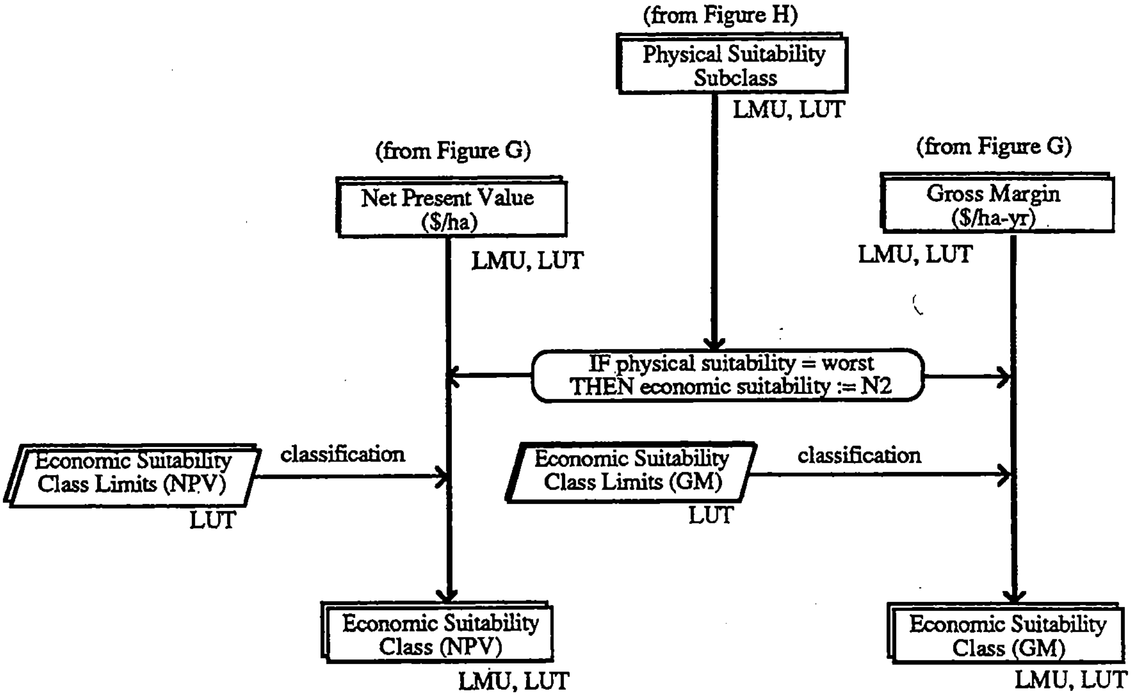


Fig. I. Computation of economic suitability classes

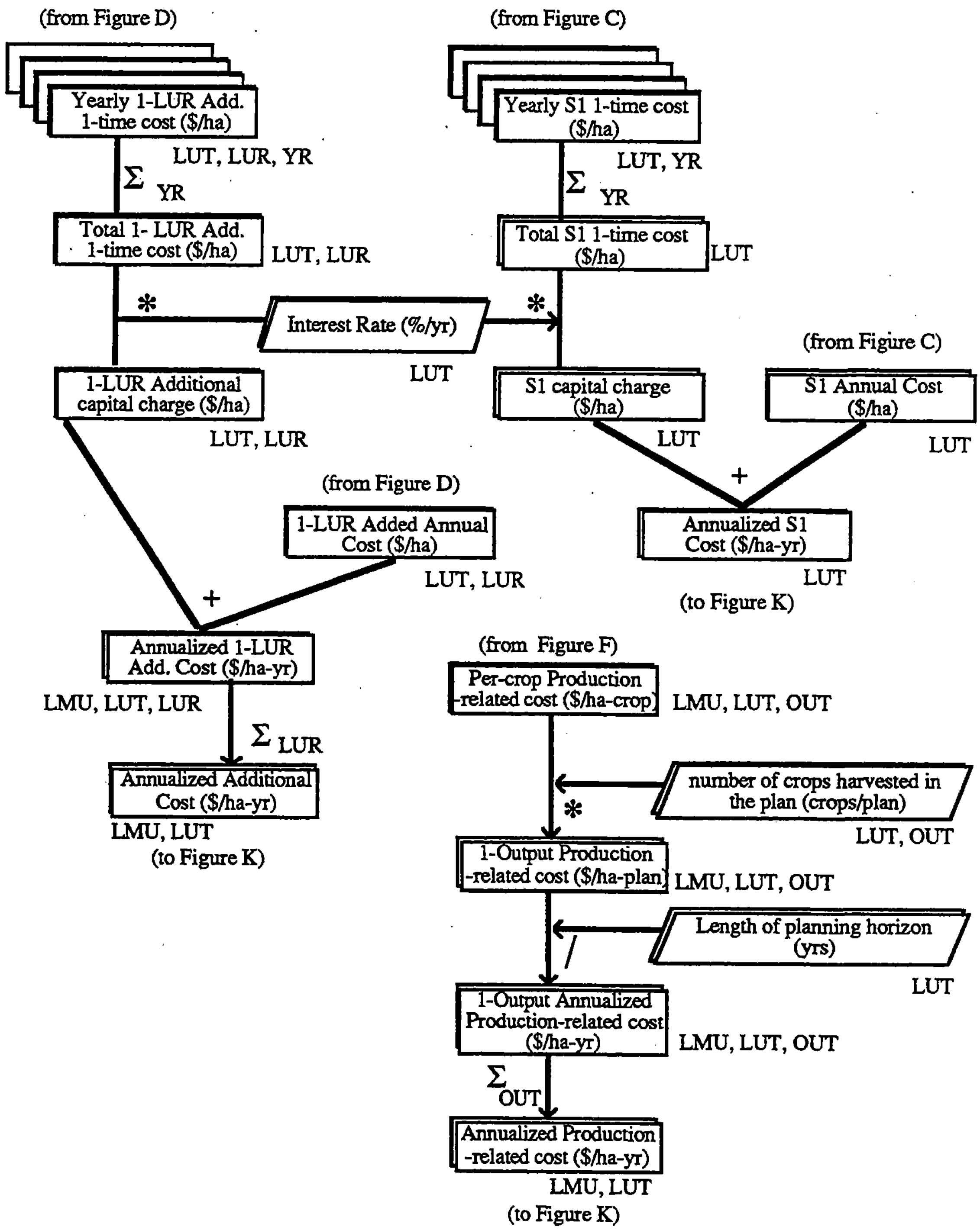


Fig. J. Computation of annualized costs for gross margin analysis

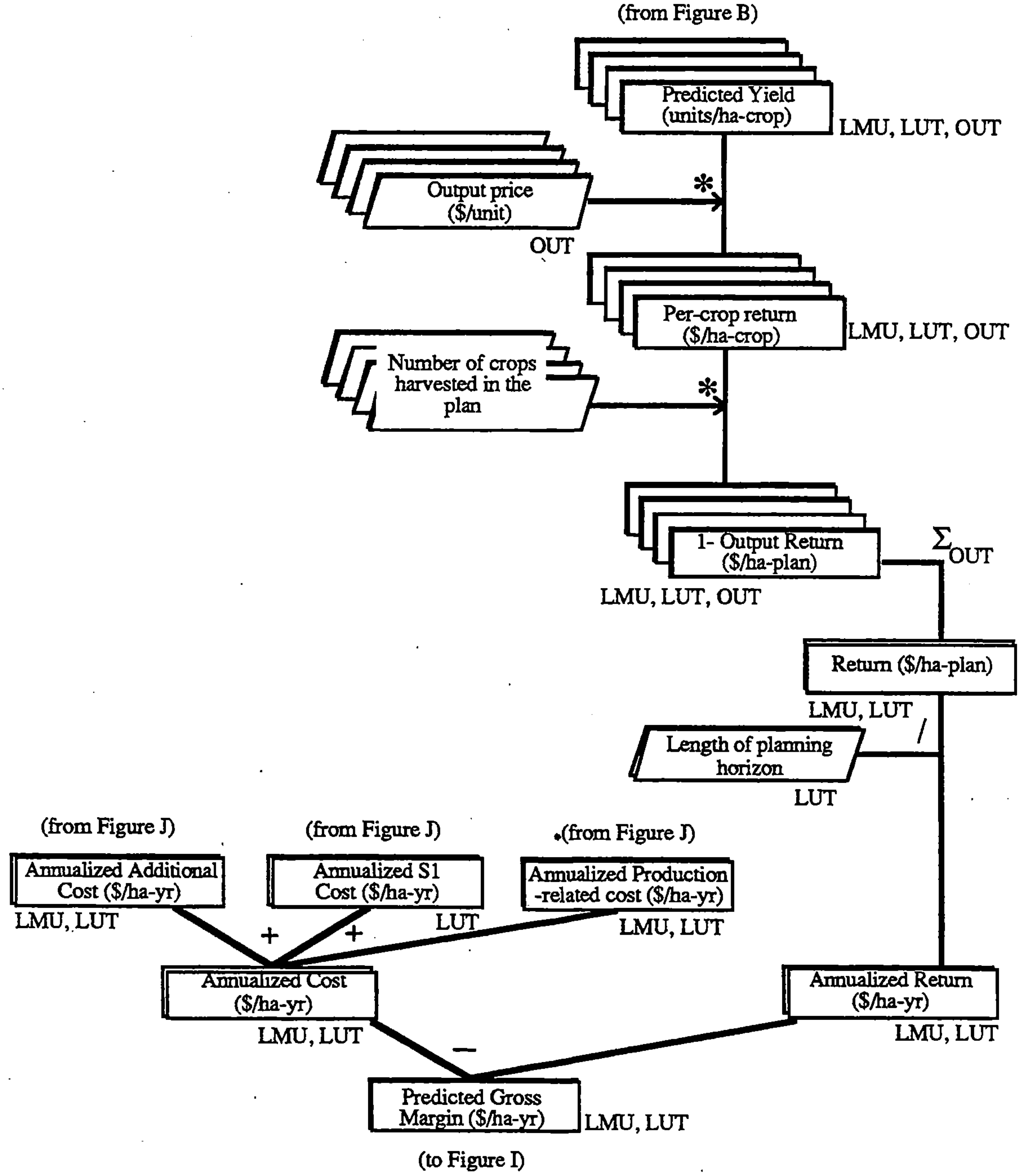


Fig. K. Computation of the gross margin computed value keys

direction of data flow

model parameters keys

logical decision

MIN minimum of a list of entities

Σ sum over a list of entities

Π product over a list of entities

\+ addition

— subtract left from right

* multiply

abbreviations used in keys:

IN = Input to a Land Utilization Type
LC = Land Characteristic
LMU = Land Map Unit
LUR = Land Use Requirement
LUT = Land Utilization Type
OUT = Output from a Land Utilization Type
SEV = Severity Level of a Land Use Requirement
YR = Year within the planning horizon

'$' represents any unit of currency
'ha' represents any unit of land area

DISCOUNT discount as: $\text{Present Value} = \text{Amount} * \left[\frac{100}{100 + \text{DiscountRate}}\right]^{\text{Year in plan}}$

Figure L : ALES computation flow diagram - legend