National Library of Canada
Canadian Theses Service
Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada
Service des thèses canadiennes

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

NL-339 (r. 88/04) c

Canada

THE UNIVERSITY OF ALBERTA

A SIMULATION APPROACH FOR INTEGRATED RESOURCE PLANNING

by

DALI ZHANG

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF MASTER OF SCIENCE

IN

FOREST OPERATIONS
DEPARTMENT OF AGRICULTURAL ENGINEERING

EDMONTON, ALBERTA

SPRING 1989

National Library of Canada
Canadian Theses Service
Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada
Service des thèses canadiennes

ISBN 0-315-52831-1

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR DALI ZHANG

TITLE OF THESIS A SIMULATION APPROACH FOR INTEGRATED RESOURCE PLANNING

DEGREE FOR WHICH THESIS WAS PRESENTED MASTER OF SCIENCE

YEAR THIS DEGREE GRANTED SPRING 1989

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(SIGNED) Dali Zhang

PERMANENT ADDRESS:

Forest Products Research Institute
Harbin, Heilongjiang Province
The People's Republic of China

DATED April 24 1989

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled A SIMULATION APPROACH FOR INTEGRATED RESOURCE PLANNING submitted by DALI ZHANG in partial fulfilment of the requirements for the degree of MASTER OF SCIENCE.

P. Wayne Anderson
Supervisor

Jack E. Heidt

Date April 24. 1989

ABSTRACT

This study is an attempt to introduce a simulation technique into the decision-making process for multiple-use of natural resources. A multi-objective forest land use problem is evaluated to demonstrate the use of the algorithm developed. The management unit, E8, of the Berland subregion, Edson forest (Alberta) is used as a case study to obtain the data for the problem.

A deterministic, dynamic prototype simulation model was constructed that may help resource managers bridge the gap between strategic planning and implementation. The model can be used for testing the long-term effects of various managerial strategies and thus acts as a tool to provide decision-makers with rapid feedback on the consequences of management alternatives.

The major components of the simulation model are five submodels which predict timber growth, recreation potential, grazing improvement, wildlife populations, and water yield changes using an increment period of one year. The most appropriate biological and physical process principles and available data were used to construct mathematical equations for those submodels. The model does not consider all the management operations, hence it is currently restricted in a number of respects. The model simulates timber harvest, hunting visitor-days, esthetic changes, grazing carrying capacities, moose and caribou populations, and water yields. Simulation results can be displayed in both tabular and graphic forms.

The computer program was written using dBASE IV, and, being menu driven, is very easy to use. Computer time required to simulate the changes on 208448 hectares over one hundred years is in the order of seventeen days on an IBM PS/2 Model 60 computer.

The model was validated by comparing predicted values with the Alberta Forest Service yield tables over a period of a hundred and eighty years. Results indicate that the model predictions are in general agreement with yield table data. Therefore it was concluded that the confidence in the model's ability to predict land use changes was fairly good. To demonstrate

how the model could be used to help resource managers, three simulation runs have been conducted using different management strategies.

ACKNOWLEDGMENTS

I am particularly indebted to my supervisor, Professor A. W. Anderson, who has freely given his advice, guidance, and support throughout the research of the project and the preparation of this thesis. Thanks are also due to the other examination committee members, Dr. H. P. Harrison, Dr. J. D. Heidt, and Dr. S. J. Titus for their valuable comments and suggestions.

Special acknowledgment is due to the Canadian International Development Agency (CIDA) for financial support for my studies and research, to the Foreign affairs office of Heilongjiang General Bureau of Fo[illegible]es, Langxiang Forestry Bureau and to the T. M. Thompson & Associat[illegible] for providing the opportunity for study. In particular, I wish to thank Mr. John Addison and Mr. Hugh Marshall for their considerate arrangement for my study in Canada.

I also would like to convey my gratitude to the Alberta Forest Service for providing data and technical assistance.

This thesis could not have been completed without the cooperative efforts of a number of extremely helpful individuals. I wish to express sincere thanks to everyone involved in this project.

Finally, I am especially grateful to my wife, *Yibing*, for her constant moral support and encouragement, and to my parents, who instilled in me from a very young age the desire for knowledge and wisdom.

Table of Contents

List of Tables

List of Figures

Figure Page

Figure Page

Figure Page

1 INTRODUCTION

1.1 Problem Definition

Integrated (multiple use) forest management is a multi-objective problem. There are a number of interdependent but conflicting objectives that need to be addressed simultaneously in the management planning process. The resource manager may, for example, wish to meet the following objectives: to increase timber production, water yield, and forage production; to maintain wildlife, fisheries, and natural beauty; to enhance recreational use and to decrease any environmental degradation. Moreover, there is no accepted common measure that can be used to satisfactorily compare all of these objectives.

Management of multiple use resources requires complicated decision making. The manager is charged with obtaining a desired mix of goods and services from limited resources, and will usually have several alternative courses of action available. To choose among them, he must know both the tradeoffs between one course of action and another, and the relative desirability of the goods and services. For example, if he wants to provide 25 percent more recreation, what quantity of timber products (if any) must be foregone? He also needs to know which goals are attainable with existing resources. For instance, does he have enough money and land to provide both the desired recreation facilities and the desired timber production quantity?

Questions like these are difficult to answer because some of the goals are complementary (some timber harvesting helps wildlife by improving habitat), but others are competitive (full utilization of forage reduces timber yield). Also, the values of some products are determined by the market, but others, such as wildlife and recreation, are better understood as non-market goods. The market valued goods have some common characteristics that enable one to build models to allocate resource between the production activities required to produce these goods. These characteristics are: 1) a single objective criterion; 2) an objective criterion that can be defined in terms of one measure; 3) goods that are valued by users in terms of that criterion; 4) use of resources also valued using the measures of the objective criterion; and 5) looking for "best" allocation in terms of maximum or minimum value for objective criterion.

Traditional investment models and linear programming models work quite well in allocating resources for market-valued goods because of their characteristics, but they may involve unrealistic assumptions when applied to the kind of decision making situations common in multiple-use forest planning (Schuler et al., 1977). Bell (1976) stated by some of these assumptions. The first assumption is that the management operations on one hectare will not affect the output on a neighboring hectare. In the case of timber management this may be fairly true, but a clearcut next to a campground could conceivably affect the number of visitor-days.

Second, linear programming assumes linearity of coefficients. This means that if 1 hectare provides for 5 visitor-days, 10 hectares will provide for 50 visitor-days. Again, in the case of timber, water, and forage, this seems to be a reasonable assumption. However, in recreation for instance, a certain number of hectares may be required before any visitor-days are produced.

Third, to keep the model simple and within the limits of available data, the effects of time and investments on the alternatives are ignored. That is, the outputs are expressed in terms of average annual potential rather than potential that varies with time or investment level.

Fourth, linear programming allows only one objective or goal to be maximized or minimized. In the multiple use management, however, numerous (often conflicting) objectives exist.

The fourth assumption can be solved by using another mathematical programming technique called goal programming. Goal programming is a variation of linear programming in which the mathematical model is so constructed that the single objective to be maximized or minimized really is composed of several goals (Bell, 1977). Although goal programming may correct the limitations of linear programming in dealing with multi-objective optimization problems, it is still limited by the other assumptions. Goal programming requires, further, the explicit specification of quantitative goals and any preference structure that may be associated with those objectives.

Simulation techniques provide another way of allocating resources efficiently in decision making situations that involve both multiple and incommensurable goals. First of all, in a simulation approach, there is no requirement that the objectives be defined in the same value terms. In fact, multiple goals may be defined in terms of cubic meters of timber, visitor-days of recreation, number of horses, or acre-feet of water, as well as numbers of wildlife. Secondly, simulation models are not tied to particular structures, mathematically, procedurally or otherwise. Therefore, they can avoid some impractical assumptions made by mathematical programming models and permit a less abstract and relatively more faithful representation of a real system. Thirdly, as computer programs, simulation models can flexibly and efficiently process large amounts of data and are quick and inexpensive to execute. Fourthly, simulation approaches study the dynamic behavior of systems so that they may consider the effects of time. Finally, simulation models are user-friendly and will produce the quantitative values that users can compare using subjective frameworks to select the most desirable strategies.

The main disadvantage of simulation is that unlike mathematical programming techniques, it shows what happens under a particular set of assumptions and does not provide the single "best" solution directly and in one quantitative measure. Simulation is a trial-and-error approach to problem solving. As such, it can be a slow method of studying problems (Garner, 1978).

1.2 Objectives and Significance

There are three objectives to be achieved in this project:

a) To explore the feasibility of using a numerical simulation approach for integrated resource planning.

b) To examine the suitability of using spreadsheet/database languages for the development of an integrated resource planning simulation model.

c) To examine the time spent on running such a simulation model on a microcomputer, such as IBM personal system/2 for different planning horizons.

The use of computer models in resource management and planning has developed rapidly in the last two decades. Recently, considerable emphasis has been placed on the introduction of "user-friendly" microcomputers into the workplace. In addition, user-friendly software, designed for professionals with little computer training, has enhanced the manager's ability to implement highly technical projects at the work station (Gray, 1986). The significance of this study lies in the interest of developing such a microcomputer-based simulation model for integrated resource planning of the Berland subregion which would be user-oriented and would allow managers to test different management strategies.

1.3 The Study Area

1.3.1 Location

The study area was the E8 management unit of Berland subregion, Edson forest of the Eastern Slopes of Alberta's Rocky Mountains, located approximately 45 km north of the town of Hinton, and immediately north of the Willmore Wilderness Area (Figure 1.1). Total area of the subregion is approximately 5,700 km^2 (or 2,230 $mi.^2$) in size (Department of Forestry, Land and Wildlife, 1987). The study area possesses a great wealth of renewable and non-renewable resources -- water, scenery, timber, forage, wildlife, fisheries and mineral resources. Demands for use of these resources have increased at a rapid rate in recent years and the fact that the resource base is not unlimited becomes more evident. Growing pressures for resources and land in the area have led to conflicts in land allocation (Government of Alberta, 1984).

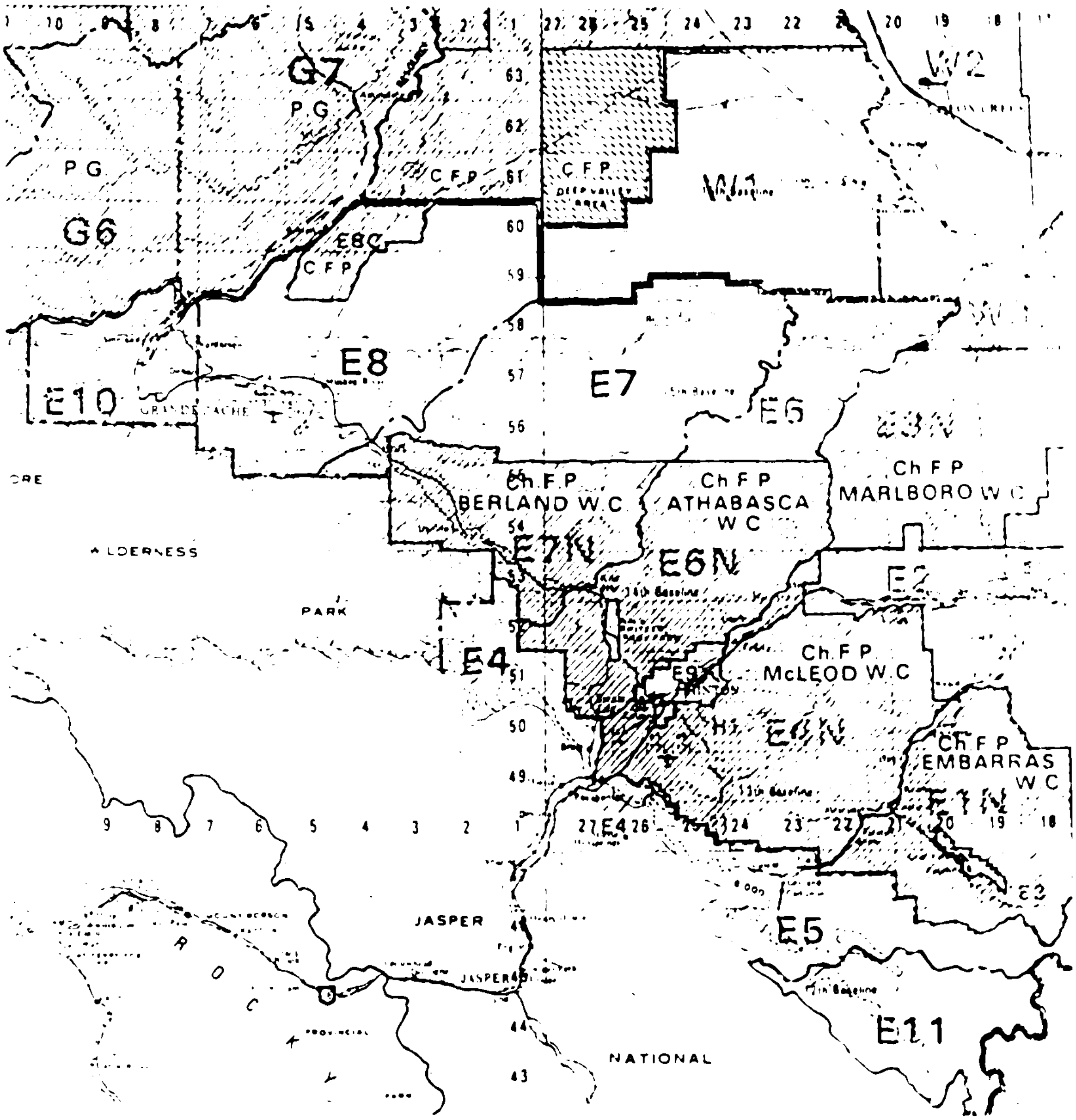


Figure 1.1 Map of Berland subregion indicating the location of E8 management unit.

According to the Regional Plan for the Eastern Slopes, most of the Berland forest is a multiple use zone. The intent of establishing this zone is to provide for the management and development of the full range of forest uses while meeting the objectives of watershed management and environment protection in the long term.

1.3.2 Summaries and Issues of Berland Subregion Resources

Based on the information in the Alberta Forest Service draft report "Berland Sub-regional Integrated Resource Plan" (1987), the timber, recreation, range, wildlife, and watershed resources and relevant issues of Berland subregion may be summarized as follows:

TIMBER

The importance of the timber resource in the Berland subregion is provincially significant. These timber resources are totally committed to existing timber processing facilities and proposed timber developments. Reductions from the existing merchantable timber land base would affect the ability of the Government of Alberta to honor existing commitments in this area.

An annual allowable cut (AAC) of over 480,000 m^3 of timber has been allocated to sustain existing quota and local use dispositions. The quota volume is committed to two quota holders in the Grande Cache area. An additional 76,000 m^3 has been allocated to Canadian Forest Products under a Forest Management Agreement.

British Columbia Forest Product's (BCFP) Grande Cache sawmill, a planer mill complex, has been in operation since 1981. This complex has a mill capacity of over 471947.44 million m^3 (100 million board feet). This sawmill relies entirely on the coniferous timber in the subregion for meeting its mill capacity requirement.

Champion Forest Products (Alberta) Ltd. is also proposing a sawmill expansion. Wood requirements for the proposed expansions will be met from timber resources within the Berland planning area.

The primary issues relative to the timber resource are the following:

a. The impact of aesthetic requirements on the land base on timber management practices in the Grande Cache area and adjacent to Highway 40 and the Forest Trunk Road (SR 734).

b. The impact of wildlife and fish management requirements on future and existing timber commitments, harvesting practices and silviculture practices.

c. Introduction of new practices, activities or changes in land use could create abnormal fire hazard or risk, or might contribute to the spread of forest disease or insects. Change in requirements for forest protection might therefore be needed.

d. Extensive areas of continuous forest cover in the Berland area represent major forest fire hazards for existing and or future settlement areas as well as a continuous supply of the timber resource.

e. Maintaining existing annual allowable sustained harvest levels essential to meet current timber commitments.

f. Maintaining timber resource development opportunities to permit expansion of the forest industry.

RECREATION

Recreational use of the Berland subregion is low when compared to areas of the Eastern Slopes further south. Currently, recreation activities are concentrated along the Forest Trunk Road (Secondary Road 734). However, levels of dispersed recreation occur throughout the area. Popular activities include camping (e.g. at Pierre Greys Lakes), hunting and fishing. Other activities such as canoeing, rafting, jet boating, horse use, hiking, snowmobiling, and cross-country skiing also occur. A considerable amount of hiking and equestrian use in the Willmore Wilderness is staged from the western portion of the Berland area and the Grande Cache area. These recreation activities are major attractions to visitors in the area.

There are two major issues on recreational use of the subregion:

a. Industrial resource developments (existing and proposed) may conflict with recreational developments and growth of the region's tourism industry.

b. Lands that contain high aesthetic qualities for recreation and tourism activities may be affected by other land uses.

RANGE

The Berland subregion supports some domestic grazing in the vicinity of Grande Cache and along Highway 40. Demand occurs largely from local residents and from outfitters for horse holding areas and lands capable of supporting horse grazing. The majority of the area is unsuitable for domestic grazing. The few areas having unimproved grazing potential are already allocated (in proximity to Grande Cache) or are very difficult to access due to their long distance from existing communities and the lack of suitable roads into the areas.

The concerns about the range management in the subregion are listed as follows:

a. There may be a limited supply of available land to accommodate local demand for grazing and horse holding areas in the Grande Cache area by residents and outfitters.

b. Range management requirements for wildlife must be balanced with domestic grazing demand.

WATERSHED

The Berland area has significant levels of resource use and development. Primary uses include timber harvesting, petroleum, natural gas exploration and development, coal development, recreation and access development. The protection and maintenance of environmental quality is a land management commitment of considerable importance. The Victor Lake Basin is the major water supply for the town of Grande Cache.

The major rivers in the area are (from west to east) the Sheep, Smoky, Muskeg, Simonette, Little Smoky, Berland, Wildhay and Athabasca. Lakes (west to east) include Victor, Grande Cache, Peavine, Pierre Greys, Joachim and Donald.

The followings are the two issues on the watershed resource:

a. The impact of resource development (current/future) on existing high water quality and soil productivity in the area, including the Grande Cache water supply

b. The impact of potential water quantity increases from vegetation removal on water quality.

WILDLIFE AND FISHERIES

The Berland area supports a diverse wildlife and fisheries resource which includes a number of sensitive species and habitats. The management of bull trout, Arctic grayling, caribou, grizzly bear, elk, moose, mountain goat and mountain sheep has intensified over the last fifteen years as demands for mineral, timber and recreational resources have escalated over this once remote portion of Alberta. The creation of surface access by the resource industries has warranted the development of extensive measures to protect many sensitive regionally and provincially significant species.

The issues which has to be considered on wildlife and fisheries may be described as:

a. Habitat destruction from resource extraction could occur to a point where regionally significant fish and wildlife populations are critically depleted.

b. Proliferation of vehicle access would have the potential for severely affecting wildlife habitat utilization and viability of fish and wildlife populations, as a result of increased legal harvest (sport hunting and fishing, Native harvest) and illegal harvest (poaching) and harassment.

1.4 Research Approach

The research for this thesis is conducted in three major components. First, a review of the literature was conducted to investigate the use of a simulation approach as a tool for integrated resource planning. Simulation approaches are compared to other forest management models. Second, an analysis of inventory data in the Berland Sub-regional Planning was performed and used in the formulation of the models of timber growth, recreation potential, grazing potential, wildlife productivity, and watershed water yield.

The last portion of the research involves an evaluation of the microcomputer software available for simulation and the development of a computer program. The comparison is used to select an appropriate language to simulate the process of integrated resource planning.

1.5 Thesis Organization

This thesis consists of ten chapters. Chapter one consists of a description of the study, a statement of objectives and an outline of the research methods. The concepts of forest management, integrated resource management, and integrated resource planning are described in Chapter two.

Chapter three is composed of a literature review of the various modeling techniques being used and their applications in integrated resource planning. The use of integrated resource planning in Alberta is discussed in Chapter four.

Chapter five is the explanation of the methodology used. It includes a description of the development of the mathematical models of the resource dynamics. Chapter six contains the discussions on the computer language selection and the model building with dBASE IV programming language.

Validation of the simulation model of the Berland forest is discussed in Chapter seven. Chapter eight presents the applications of the simulation model. A summary of the results and the conclusions are found in Chapter nine and Chapter ten contains recommendations for future work.

2 FOREST MANAGEMENT AND INTEGRATED RESOURCE MANAGEMENT AND PLANNING

2.1 Forest Management

Forest management is concerned with the sustained production of forest products and services and the maintenance of high quality forest environment. There is an increasing demand for timber and associated products. Man's ability to manipulate the processes of nature is increasing through improved technology and in so doing he is altering the balance of relatively stable ecosystems by polluting the atmosphere, by draining wet-lands, by clearing forests, by afforesting using monocultures of exotic species, and by building towns and roadways. Forests can play an important role in the preservation of those ecosystems and in the conservation of the environment as well as providing wildlife refuges for endangered species of fauna and flora. The growing demand for outdoor recreation and hunting has made its impact upon the forest often resulting in special areas being re[illegible]r their exclusive use, and particularly in times of recession t[illegible] the forest in providing meaningful employment opportunities shoul[illegible] be ignored.

While it is sometimes possible to reserve a forest area to one exclusive use, the inevitable trend is towards a number of simultaneous uses. Pressure on the forest to meet the needs of the people is rapidly increasing everywhere and particularly so in the more densely populated countries. For the forest manager this means striking a balance between competing use or multiple use (Joyce, 1981).

2.2 Integrated Resource Management

Integrated resource management as a forest management concept is as old as forest management itself but the term came to enjoy its great popularity only in the mid-sixties, when it became a popular slogan meaning different things to different people according to their interest (Davis, 197[illegible]). Marczyk (1985) defined integrated resource management as a management strategy which endeavors to optimize use of the resource base to achieve maximum benefits for the people now and in the future. The concept recognizes that management and use of any resource inevitably affects management and use of other resources. The approach attempts to minimize existing and potential conflicts and promote positive interactions among uses. Integrated resource management, according to Watts (1983), refers to the exploitation of two or more natural resources compatibly on a given land area. A simple example of this idea is the cooperative use of a watershed for both forest and fisheries resources. Duerr, et al. (1982) describe integrated resource management as a program of managerial inputs rationally selected to produce a desired set of forest services. According to this view, integrated resource management is the result of a decision-making process.

2.3 Integrated Resource Planning

Social conflicts over integrated resource management in forestry tend to be expressed in conflicts regarding land use and the processes of integrated resource planning. Planning is the process; plans are its outcome. A land use plan depicts the way land may be allocated during a future period and explains the objectives sought, the expected flow of services, the projects needed to implement the plan, the costs and benefits, and other matters, including environmental-impact analysis (Duerr, et al., 1982).

Camp (1973) suggests that, for the forester, land use planning is organizing the development and use of a forest and related lands and their resources in a manner that will best meet the needs of people over time, and to maintain maximum flexibility for a dynamic combination of resource outputs for the future.

Integrated resource planning is an integral component of integrated resource management and shares the same principles. Its relationship to integrated resource management as a means to implement that concept has been recognized for nearly two decades. Its major distinguishing feature is that it provides future-oriented direction for action by means of a dynamic, continuous and sometimes repetitive process of decision-making

(Marczyk, 1985).

Some vital questions which must be considered during the planning process are the following:

a. Where are the resources?
b. What is their condition?
c. How accessible are they?
d. What are the opportunities to improve production of individual resources?
e. How will changes in one resource affect others?

Although answering all of the questions completely may not be possible, the most current information and technology can help describe the complex interactions among the resources involved.

Brown (1981) argues that the process for deciding how public land should be managed involves a focus on tradeoffs. A tradeoff is a relationship between two or more effects of a change in some condition (such as the condition of the forest). The relationship signifies a difference between the initial condition and the new one which the change would bring about. The advantage of examining tradeoffs between forest conditions is that attention is focused on actual quantities and values of those quantities rather than on abstract values.

Abstract values, characterized by statements such as "Timber is more important than range." generally are of little use in decision making because they do not apply to specific management options in given locations. Much more important are comparisons of, for example, the value of an increase of a certain quantity of timber with condition A to the value of an increase of a given amount of forage with condition B. A process incorporating this focus involves ten general steps:

a. assessing management concerns, public issues, and resource use and development opportunities;
b. deciding on planning criteria;
c. collecting relevant data and information;
d. analyzing the management situation;

e. formulating feasible, realistic alternatives responsive to such assessments;

f. estimating the physical, biological, social, and economic effects of the alternatives;

g. comparing the alternatives and evaluating their differences;

h. choosing a preferred alternative;

i. implementing the alternative; and

j. monitoring the implementation and its effects.

3 INTEGRATED RESOURCE MANAGEMENT AND PLANNING MODELS

The techniques used in multi-objective management and planning models are usually referred to as either simulation models or mathematical programming. Mathematical programming is the term applied to a group of optimization techniques including linear programming, goal programming, integer and mixed integer programming, quadratic programming, geometric programming, and dynamic programming. All of these techniques are designed to select an optimal solution for a set of variables, often called activities. The optimal outcome is the numerical maximum or minimum of some specified performance criterion or objective function (Joyce et al., 1983). Simulation is another operations research technique. In contrast with mathematical programming models, simulation models have no built-in algorithms, like the simplex, that lead to an optimal solution.

3.1 Linear Programming

Linear programming (LP) is a mathematical programming technique which can be used to maximize or minimize a linear objective function, subject to a set of linear constraints. Linear programming models have three major components (Joyce, et al., 1983):

a. A set of all possible activities under consideration. These are also called the decision variables.

b. A set of limitations on the resources needed to carry out the activities. These are called the constraints and, as linear combinations of the decision variables, make up the rows in the LP matrix. The sum of resources used by the activities must be constrained to the total resources available.

c. A performance criterion for selecting the optimal set of activities and the level of each activity from all possible activities.

A simple representation of a linear programming problem follows:

Maximize:

$$Z = \sum_{j=1}^{n} C_j X_j \tag{3.1}$$

Subject to:

$$\sum_{j=1}^{n} A_{ij} X_j \leq B_i \quad i = 1 \ldots m \tag{3.2}$$

$$X_j \geq 0 \quad j = 1 \ldots n \tag{3.3}$$

In this example, there are n activities, and m constraints. The total amount available of a specific resource m is represented by the coefficient B_i, the activity level for a decision variable j is represented by the X_j and the amount of resource used by one unit of activity X_j is A_{ij}. Additionally, activities $[X_j]$ may not take on a value less than zero.

Several assumptions about the real system being modeled are necessary before a linear programming model of that system can accurately describe the system. First, one assumes that both the objective function and the resource constraints can be represented as linear combinations of the decision variables. This is not as restrictive as it would seem initially as many nonlinear relationships can be represented using piecewise approximations of nonlinear functions. Linearity is assured by two requirements--proportionality and additivity. Each activity's contribution to the objective function and its rate of resource use is proportional to the level of that activity in the solution. That is, coefficients in both the objective $[C_j]$ and constraints $[A_{ij}]$ are constant for all levels of X_j. The total contribution to the objective function and the total resource use of engaging in two or more activities at the same time must equal the sum of the individual contribution, or resource use, of each activity engaged in separately.

Second, one assumes that activity levels are restricted to only positive values. In the real world, a negative activity level does not make sense. Third, one assumes that all decision variables must be divisible; that is, they can take on fractional values.

Fourth, one assumes that values for coefficients $[C_j$ and $A_{ij}]$ can be specified before the model is run and the values of these coefficients are known with certainty. Thus, a linear programming model is a deterministic model.

Finally, one assumes that only one criterion is used at any one time in selecting the optimum combination of activities.

3.1.1 The advantages and limitations of linear programming

Linear programming provides a useful approach to the decision maker seeking to optimize some specific objectives. If the benefit from each forest use can be given a monetary value the technique can be effective in providing opportunity cost information. The important thing to remember is, however, that "optimal" solutions are best only in terms of the objective function which is optimized. Nevertheless, the method provides useful information for making a choice and is a positive step towards evaluation of multiple use objectives (Joyce, 1981).

Although the assumptions incorporated in linear programming models are well known, the implications of these assumptions are not usually consciously considered by operations research analysts or the general users of linear programming models. However, the interpretation of results and their implementation by resource managers depend heavily on these assumptions and their relationship to the "real world". Bare and Field (1987) give a detailed discussion on this topic.

First, because of the linearity assumption, linear programming models are often criticized as being too simplistic to model many real world phenomena. "The world is not linear!" This is an often heard battle cry, and is generally a valid observation. But, the implication of linearity that is most often challenged is proportionally. This deficiency can be overcome with proper formulation of the linear programming model. However, the more insidious implication of linearity often is overlooked. This is the additivity assumption. Additivity means that any combination of feasible values for the decision variables produces a consequence which is the sum of the individual values. No interactions between decision variables are permitted to cause variations from this total.

Secondly, linear programming does not guarantee integer solutions because of the divisibility assumption. In many linear programming models, however, integer-valued decision variables are required. Examples of such decision variables include numbers of campgrounds, animals, etc. To permit a noninteger value implies infeasibility in the real world, making such a solution difficult to implement. This limitation is being overcome by use of mixed integer solution packages.

Thirdly, a commonly cited criticism of the linear programming method is that it is deterministic. This is because probability functions, which are often nonlinear, are impossible to incorporate into linear programming models. Hence, all coefficients in a linear programming model are assumed to be known with certainty. In attempting to incorporate uncertainty into a linear programming model, both model size and solution time increase dramatically.

A final weakness of linear programming is that it explicitly assumes there is a single decision criterion to be optimized and implicitly assumes there is a single decision maker. Neither of these conditions hold for public decision making in general.

3.1.2 Early applications of linear programming to integrated resource planning

Over the past two decades, great progress has been made in developing computer models as a tool for decision-making in forest management and multiple use planning. There are a number of such applications described in forestry literature but no great indication that linear programming is being adopted by individual forest managers or planners (Joyce, 1981). Chappelle (1977) comments that the situation seems to be one of researchers talking to other researchers and suggests a number of reasons; the type of mathematics involved, the difficulty of isolating a primary goal that could be expressed as an objective function, the high data requirements and the difficulty of getting expert advice. Because of those obstacles, the operational application of linear programming occurred only in the larger organizations where the scale of operations was sufficiently large to justify specialized staff. A number of "prepackaged" models have been developed for operational use in forest management and multiple use planning Notable among those are Timber RAM and FORPLAN (Walker, 1982).

3.1.3 Major problems of linear programming planning models

Although linear programming models have played an important role in forest planning, the definite shortcomings of those models should be recognized. Rose (1984) gives a detailed discussion on several major problems of linear programming models. A serious limitation of linear programming in forest planning is model size. The decision variables in the model represent specific management alternatives for specific units of land. The decision variables can be defined in a number of different ways. However, a basic problem is that the number of distinct land units that can be recognized in practice is limited.

The problem of uncertainty surrounding economic values, yield values, demand projections and other entities is another area of concern, since linear programming models have the distinct disadvantage of not permitting extensive sensitivity analysis except at very high cost. Poor data are problematic in any planning model. The problem with the forest planning process is the use of data with varying amounts of precision in one overall model structure. Data are treated as deterministic and the same degree of confidence or weight is assigned to them.

Several factors stand in the way of actually achieving the optimal solution. One problem is that different managers can develop different implementation plans because of the aggregated nature of the solution. More seriously is the requirement that attainment of the optimal solution requires that the resource management schedule must be followed throughout the planning horizon. Since aggregation can lead to increased sensitivity of solutions to changes in assumptions, already taken actions will be more likely to turn out to be non-optimal and often irreversible (Rose, 1981).

The cost effectiveness of the FORPLAN (Forest Planning Model) process is an important, but difficult question to answer. Such an evaluation would require a quantification of model inputs and outputs and comparison with other alternative evaluation procedures. Such an analysis and comparison are not possible to date due to lack of information.

It would be misleading to not point to the positive aspects of the linear programming planning models. FORPLAN certainly has provided a structure for addressing the forest management problem. Considering, however, all the problems of applying linear programming models, it is reasonable to question whether or not another operations research technique like goal programming or simulation might be a more useful tool.

3.2 Goal programming

Linear programming, in its simplest formulation, is unable to handle multiple goal situations which are extremely common in resource management. "Multiple use" solutions are not derivable from simple linear programming models since only one objective function can be specified at a time. Multiple goal situations have been handled by some analysts using linear programming by using the equation that contains the quantitative measures of the major goal as the objective function and relegating equations quantifying the lesser goals to the constraint area of the problem. This is not the same as calculating a solution that will satisfy, to the extent feasible, the entire set of goals given a relative preference ranking of the alternative goals. This limitation can be overcome by using the mathematical programming technique of goal programming.

Goal programming, probably, the most widely used technique for general multi-objective programming, is certainly the one which has been most extensively applied to natural resource management problems. It is an extension of linear programming. Developed by Charnes and Cooper (1961), goal programming minimizes deviations from multiple goals, or objectives, subject to some constraints that are goal statements and others that are physical constraints. Mathematically, the program can be expressed as:

Minimize:

$$\sum_{i=1}^{n}(u_{1i}d_i^- + u_{2i}d_i^+) \qquad (3.4)$$

Subject to:

$$AX + Id^- - Id^+ = G \qquad (3.5)$$

$$BX \leq C \qquad (3.6)$$

$$X_j \geq 0 \qquad j = 1, \ldots, m \qquad (3.7)$$

$$d_i^-, d_i^+ \geq 0 \qquad i = 1, \ldots, n \qquad (3.8)$$

where

d^-, d^+ - the vectors of the negative and positive deviational variables, respectively. These represent the solution's deviation from the goal vector, G,

w_{1i}, w_{2i} - the weights and/or priorities for the deviational variables,

A - the matrix representing the relation between the decision variables vector, X, and the goal vector, G,

I - the identity matrix,

B - the matrix of coefficients which relate the decision variables to the constraint vector, C, and

n, m - the number of goals and decision variables, respectively.

In contrast to physical constraints, the goal constraints are satisfied as closely as possible but need not all be met completely. This flexibility is obtained at a cost: goal programming requires the explicit quantification not only of objectives but also of goal levels associated with these objectives and a preference structure defining the decision makers' preferences for each objective relative to the others. The difficulty of specifying the goal levels and preference ratings in a satisfactory and objective way has surrounded goal programming with considerable controversy.

3.2.1 The advantages of goal programming over linear programming

The main advantages of goal programming over linear programming is that goal programming attempts to correct the limitations of linear programming in dealing with multi-objective optimization problems, while retaining its useful basic structure and numerical solution.

Linear programming focuses on the problem of determining an optimal allocation of scarce resources to meet a given set of objectives. Goal programming, in a similar format, seeks a plan that comes as close as possible to attaining specified goals. Both procedures deal with constrained optimization. Both are limited by the assumptions that model variables are infinitely divisible and connected only by linear relations. Goal programming requires, further, the explicit specification of quantitative goals and any preference structure that may be associated with those objectives. It is this orientation that provides the technique with the flexibility necessary to circumvent two major weaknesses of ordinary linear programming. A conventional linear programming model may incorporate two classes of objectives: (1) an overall, single-dimensional, optimization criterion such as profit maximization or cost minimization; (2) a set of secondary requirements imposed by the decision-maker (distinct from absolute physical or economic constraints) such as the attainment of certain minimal production levels. Ordinary linear programming procedures yield an optimal solution to a quantitative allocation problem only if a feasible solution exists. Feasibility is assured if the requirements specified by the analyst and the constraints imposed by the problem environment are all mutually consistent. But, inconsistencies are not always readily apparent. For example, it may not be obvious, prior to the analysis, that limited resources preclude the simultaneous satisfaction of a minimum desired timber yield goal and a watershed management objective. In contrast, the objectives specified in a goal programming format are approached as closely as possible but need not all be met completely. This flexibility allows the specification of a problem in terms of multiple conflicting goals and the allocation of resources according to subjective priorities.

Given the existence of a feasible solution to an ordinary linear programming problem, a second shortcoming is the requirement of a single-dimensional optimization criterion. Whatever the measure associated with the objective specified by this criterion, the outcomes of the several activities included in the solution plan must be expressed in common units This requirement has two particularly serious effects. First, analysts eager to apply linear programming to problems involving incommensurable values are tempted to search for indirect measures of relatively intangible results in terms of those more easily valued. Thus, for example, vacation expenditures are used as a surrogate gauge of outdoor recreation benefits, and a wilderness preserve is valued in terms of timber harvests foregone. Secondly, even when a clearly valid relationship between the optimization criterion standard and a particular activity does exist, that relation may be very difficult to specify. Goal programming allows not only the simultaneous consideration of resource allocations to activities whose outcomes cannot be valued in like terms but it also permits the analyst to specify directly activities whose levels can be associated with a common measure. For example, the consequences of a shortage of pulpwood at a mill can be expressed in cords rather than requiring the difficult estimate of the overall dollar impact of such a shortage on the firm's operating costs and sales revenues (Field, 1973).

3.2.2 Applications of goal programing to integrated resource planning

Goal programming is probably the most extensively used operations research technique in the area of forest management and land use planning. The number of published applications of goal programming in this area has been substantial, especially in multiple use contexts. Field (1973) introduced goal programming to the forestry literature. Bottoms and Bartlett (1975) applied goal programming to integrated resource management of 9,050 acres of the Colorado State Forest. Their formulation used ordinal priority ranking of goals. Bell (1976) further discussed transformation of a linear program into a goal program with a composite weighted objective function. Dane, Meador, and White (1977) used the composite weighted objective form of goal programming on a 158,000-acre planning unit on the Mt. Hood National Forest, Oregon. Schuler, Webster, and Meadows (1977) reported on their pilot application of goal programming on a 10,000-acre subunit of the Mark Twain National Forest, Missouri, for the evaluation of tradeoffs between timber management, outdoor recreation, grazing, and production of game animals for hunting.

3.3 Simulation

Although the optimization methods such as linear programming and goal programming have been widely used in solving integrated resource planning problems, their limitations discussed previously have motivated the search for alternative planning techniques. A simulation approach using heuristic rules (trial and error) appears to be promising (Rose, 1984).

3.3.1 Principle of simulation

Any attempt to discuss the overall field of simulation is a formidable task simply because the term "simulation" has been so ill-used and abused. Simulation appears to apply to almost anything unreal which reports to be something like reality. It, therefore, has been used widely in industry, agriculture, forestry, and government. However, simulation has different definitions for different people. Simulation, according to Phillips, et al. (1976), is the process of duplicating the essence of a system without attaining the reality of that system. Simulation refers to mathematical and statistical models that have been implemented on a computer (Joyce, et al., 1983). Joyce (1981) indicated that to simulate is to copy the process of an operation over a time period. Based on those definitions, one may infer that any kind of process of developing computer models to predict the future world could be called as "simulation". In fact, this is not a precise definition of simulation. "Computer modeling" can be a better name for this process rather than "simulation". Simulation is a particular technique of operations research. Simulation models deal with the study, essentially by trial and error, of **dynamic** problems which can not be solved by common mathematical methods. For example, a simplistic mathematical model of the growth of a deer population in a forest would state that the growth proceeds at a rate proportional to the number of animals. This relationship can be expressed as a simple equation whose solution gives the population size as a function of time. In fact, however, the growth of the population is also a function of the amount of food available in the forest, which itself changes at a rate that depends on the way the forest is managed, and so on. To solve such problem properly, one needs a system of equations for which there are no exact solutions, only approximate ones. Buongiorno and Gilless (1987) described simulation as the process of developing a model of a real system and conducting experiments with the model.

As described by Pritsker (1984), in the general sense, simulation deals with the study of (dynamic) systems over time, and is the representation of the dynamic behavior of the system by moving it from state to state in accordance with well-defined operating rules. Similarly, Grant (1986) defined simulation as the process of using a model to mimic, or trace through, step by step, the behavior of the system under studying. He stated the following:

> "Simulation models are composed of a series of arithmetic and logical operations that together represent the structure (state) and behavior (change of state) of the system of interest. The system of interest exists in different states at different points in time and there are rules governing the manner in which the state of the system changes as time passes. The rule governing change also may change from time to time, since they are themselves a function of the state of the system. If we choose appropriate variables to describe the system and appropriately represent the rules governing change, then we should be able to trace the state of the system through time, which is to say we can simulate behavior of the system."

3.3.2 The advantages and limitations of a simulation approach

Simulation models are designed to sample the characteristics of the system they represent by "observing" the system over time and subsequently gathering pertinent information. In essence, this would be very much the same as observing a real system but with the advantage that the analyst is controlling the simulated system rather than being controlled by it This means that the analyst can experiment with a system and study its performance while changing its parameters and decision rules at will. Another important advantage is that the actual time span over which the real system is simulated can be compressed so that its performance over a one-year span may be examined in minutes on the microcomputer (Taha, 1976).

Simulation is perhaps the most versatile tool for dealing with complex (or nonconforming) systems. Simulation models have a degree of flexibility not available in mathematical programming and have the great advantage that individual solutions are inexpensive to obtain. The system can be used to indicate the results of various courses of action thus allowing the decision maker to select the most desirable strategy.

Simulation models are not tied to particular structures, mathematically, procedurally or otherwise. As computer programs, they can flexibly and efficiently process large amounts of data and are quick and inexpensive to execute. The reluctance of forest managers to adopt simulation models in forest management planning may be attributed to a number of reasons including lack of expertise in constructing models, unfamiliarity with computers and difficulty in obtaining access to computer terminals (Joyce, 1981). With the rapid technological developments in microprocessors, relatively inexpensive microcomputers and mini-computers are now available which will accommodate sizable simulation packages.

Simulation answers questions of the type: "What would happen if I did this...?" Once a forest system has been modeled and the model found to be acceptable, a manager has great flexibility in imposing changes on his forest model. He can vary his management goals and the conditions of his stands with no ill effects to the real stands from any undesirable alternatives. With adequate mathematical relationships, the manager can predict probable future performance and yield of his forest (Myers, 1973). For the multiple use zones, in seconds, he can get an estimate of the long-term effects of proposed timber harvesting on recreation, grazing, wildlife and fishery, and watershed, etc.

The main disadvantage of simulation is that, unlike other operations research techniques, it does not provide optimal problem solutions directly. Simulation is a trial-and-error (heuristic) approach to problem solving. As such, it can be a slow method of studying problems.

3.3.3 Simulation model building

Joyce, et al. (1983) indicates that the process of constructing a simulation model can be broken into the following five stages: conceptual, diagrammatic, mathematical, computer programming, and validation/verification.

In the conceptual stage, the modeler's experience and intuition suggests important features about the system's structure to be modeled, given the questions being asked about the system. In the diagrammatic stage, word models and diagrams are used to structure the model.

The complexity represented in the diagrammatic model is referred to as the degree of aggregation. Because a model can never completely replicate the system, compartments of the real system may be combined in the model. One example would be combining all minor tree species into one compartment instead of modeling each tree species separately.

In the mathematical stage, the structure and function of the system are described in equations. The relationship between the state variables and/or flows is determined from experimental work, previous models, or the modeler's intuition. The fourth stage in the modeling process is the computer programming stage where the mathematical equations are written into a computer program.

The last stage in the modeling process is the validation/verification stage. The model is tested under a variety of situations. An error analysis may be performed to determine the magnitude of error propagation in the model. A sensitivity analysis may be performed to determine how sensitive the model is to changes in the parameters of the equations. Finally, the model output is compared with different field data that were not used in constructing the model to determine how well the model mimics the real world.

3.3.4 Applications of simulation models

Simulation has been applied to almost every aspect of forestry and forest planning operations. Perhaps its most widely used application has been in the development of models which simulate the growth of forest stands. This enables the manager to study the problems arising because of decisions to harvest and helps him to compare the outcome of alternative strategies. Examples of such models are given in Fries (1974) and Fries, et al. (1978). Simulation has been extended to an examination of the physical, economic and environmental consequences of alternative wildland use decisions by Bare and Schreuder (1975) in which they use a set of simulation models each of which describes a major component or subsystem of a forest ecosystem.

Some of the earliest applications of ecosystem simulations were those of Odum (1960) and Olson (1963) on an analog computer, and Garfinkel (1962) on a digital computer (Wiegert, 1975). Since these attempts, simulation modeling has diversified greatly. Simulation models of ecosystems range from large, complex systems of equations to small sets of differential equations.

Models constructed for natural resource management differ in structure from modeling approaches in ecological simulation. This difference was pointed out by Spofford (1975) for aquatic systems models and Reed and Clark (1979) for forest growth models. Hoganson and Rose (1984) describe a simulation approach for optimal timber management scheduling.

Applications of simulation modeling vary greatly; in most cases, the simulation model was built for a specific ecosystem, and often for a specific set of management problems. Van Dyne and Abramsky (1975) compiled a list of models used in agricultural and natural resource fields, including both simulation and optimization models. Weigert (1975) and Frenkiel and Goodall (1978) reviewed the development of simulation modeling and critiqued simulation model building. Odum (1971) and Holling (1978) presented modeling frameworks and several case examples. Potential users of simulation models need to critique carefully the assumptions underlying the simulation models.

As Odum (1976) noted, no theoretical framework previously existed to guarantee the same structure in each ecosystem model.

There are no simulation models for integrated resource planning found in the literature, although some models (Myers, 1977; Betters, 1975) had taken two different management activities into account.

3.4 The Role of Computer Models in Decision Making

A common misconception about computer models is that they provide objective, error-free optimal solutions. This implies that decision making can be automated if all variables are properly defined. However, decisions are required throughout the process of developing models and, while the assumptions are intended to be reasonable, these decisions are subjective. Often a problem is influenced by non-quantifiable factors. So while a solution may be found, modifying the solution to fit these other considerations, and then successfully implementing the solution, continues to require the experience, intuition and judgement of decision makers. Therefore, models can only be used as an useful tool in decision making.

A simulation model can be an *aid* to use management planning, but it does *not* design use-management systems or select an optimal use strategy. A simulation model will give a manager the probable results of any use level and pattern that he might want to consider. The manager might use the results of earlier simulations to help guide the design of a new, potential use strategy, but the initiative still comes from the manager, not the computer. In designing a use strategy for simulation, the manager must be guided by what is possible. A simulation model will present results for any use strategy. The manager, not the computer, also evaluates the results. Using professional judgment, aided where possible by research knowledge, the manager decides if the amount of timber harvests are acceptable in light of management objectives. This also is what would be done with an on-the-ground trial, except more information is available much faster from a simulation model. The manager can consider many different possible use-management plans, compare results both in terms of use and encounter levels, and also consider how much total use would take place under each plan. With this sort of foreknowledge, better choices of use-management plans should be possible (Shechter and Lucas, 1978).

3.5 Criteria for Evaluating Planning Models

In order to clearly delineate limitations of any model, it is necessary to develop evaluative criteria that relate to the specific type of problem that the model is designed to solve. However, it is difficult to define one criterion to evaluate any type of planning model. More realistically, a large number of factors need to be considered. One might base an evaluation on a number of questions (Rose, 1984):

a. Does the model generate solutions that are at least theoretically sound (valid)?
b. Does the model develop an implementable plan or can results be used to develop an implementable plan?
c. How can the model fit into the general planning process?
d. Will the planning process be cost effective with this model?
e. Would different planners reach different results?
f. Can the plan be made flexible and responsive to questionable model assumptions about the future?
g. Is the model large enough to recognize most of the pertinent data?
h. Is the planning model understandable or viewed as a black box?
i. Does the model deal effectively with uncertainty aspects?
j. Is the model transferable to other users?
k. Does the model allow for adjustments to specific situations?
l. Can model results help evaluate specific alternatives not recognized in the model?

Furthermore, Chappelle et al. (1976) offered the following evaluation criteria which ought to be met by all models if they are to be useful in comprehensive multiple use forest management planning:

a. the model should be capable of accepting data as inputs for the physical, biological, economic and social variables that bear most importantly on the problem;

b. the model, even if constructed for a single-resource functional analysis, should be capable of linkage with a comprehensive multiple-use planning model;

c. the model should be capable of handling both the temporal and spatial dimensions of resource production and management;

d. outputs of the model should be presented in a format that can provide ready guidelines to planners and managers; and

e. the computer program should be efficient (i.e., compute model outputs at least cost), be transferable to computer centers with a variety of machinery, and be documented to the extent that it can be readily modified by a skilled computer programmer employed outside of the group responsible for development of the program.

The integrated resource planning problem is too large and too complex to expect that a single run of any computer model can produce the desired plan. Therefore, to expect too much from any model can be dangerous. The key to success is in recognizing both the advantages and the limitations of a model.

4 INTEGRATED RESOURCE PLANNING IN ALBERTA

Approximately two-thirds of Alberta is public land, most of which is forested. This land is the focus for many activities -- logging, trapping, grazing, exploration for oil and gas, coal mining and many forms of recreation, such as hunting, fishing, hiking, camping and use of off-highway vehicles. In the past, land-use decisions were often made by individual agencies without consideration for other existing uses or possible future uses. As the demands on Alberta's public land and natural resources have increased, the decision-making has also changed. Integrated resource management has become the fundamental approach to decision-making for public land and resources in Alberta. The key is shared decision-making. Through this process, a balanced set of decisions result. Land-use conflicts are minimized and the best uses for the forest resources are identified (AFS, 1988a).

4.1 History of Integrated Resource Planning

Integrated resource planning in Alberta originated as a component of resource management in the Eastern Slopes region of the Rocky Mountains. In 1973, the Environment Conservation Authority conducted major public hearings on all aspects of land use and resource development in the Eastern Slopes and in subsequent recommendations to the government strongly stressed the need for integrated resource policies and land use planning for the area. The government response in 1975 by forming the Eastern Slopes Interdepartmental Planning Committee which reported to Cabinet and was responsible for making recommendations on an integrated resource planning approach for this strategic region of Alberta. In 1976, the Resource Planning Branch of the Resource Evaluation and Planning Division of Alberta Energy and Natural Resources was created to provide integrated resource plans to facilitate the optimum use of public lands in Alberta (Fardoe, 1984).

4.2 Characteristics of Integrated Resource Planning

Integrated resource planning attempts to integrate resource planning activities through the following (Fardoe, 1984):

a. Providing coordination of resource planning activities. This is accomplished by applying an organized, systematic approach to planning.

b. Promoting comprehensive consideration of resource values. This is accomplished by providing a wide range of opportunities for participation in integrated resource planning.

c. Promoting cooperation and communication of resource values. This is accomplished by providing a forum for discussion and development of resource planning initiatives and a mechanism for resolution of identified conflicts. A team or committee approach to plan preparation and approval is used extensively.

4.3 Levels of Integrated Resource Planning

In an attempt to provide a hierarchy of planning direction with progressive refinement of direction from broad to specific, four levels of integrated resource planning have been identified. Each level of planning provides resource management direction and a framework for more detailed levels of planning. The levels of planning are based on the geographic scale of the planning areas and the distinctions between levels are somewhat arbitrary. The four levels of planning and the approximate scale of planning areas are provincial (1:1 000 000), regional (1:250 000), sub-regional (1:100 000) and local (1:50 000 - 1:10 000) (Marczyk, 1985).

Pı .ncial level resource planning involves the development of strategic, long-term policy statements or objectives for natural resource throughout the province. This level of planning interprets resource management legislation and political direction and is usually the responsibility of senior executive levels within resource management agencies.

Regional level resource planning deals with the development of planning direction for broad areas of the province. Regional plans provide a general allocation of resources towards meeting provincial objectives and identify planning priorities within the region.

Sub-regional level resource planning involves the development of planning direction for sub-regional throughout the province. Sub-regional plans identify desirable patterns of land use and an appropriate mix of resource use activities. Field level resource managers are heavily involved in development of sub-regional plans. Over the past several years, integrated resource planning has focused mainly on sub-regional plans.

Local level resource planning provides direction for smaller planning areas throughout the province by developing detailed land and resource use parameters. The main participants in developing local plans are field-level resource managers.

4.4 Integrated Resource Planning Process

The integrated resource planning process is a systematic series of interrelated actions that leads to development of integrated resource plans. The process is based on the decision-making fundamentals of information-choice-action. The planning process is basically sequential but may be interrupted to return to a previous step if required. Documents are produced at key stages of the process and are approved by the appropriate government committee. Although there are slight variations associated with the four levels of planning, the process generally consists of the following steps (Fardoe, 1984):

a. initiation of a plan,
b. collection and analysis of resource data,
c. development of integrated resource management policies and objectives,
d. development of integrated resource management guidelines for action,
e. consolidation of direction into a final plan,
f. development of a plan implementation strategy, and

g. review and revision of plan.

The planning process is rational in the sense that alternatives and consequences of planning decisions are considered as fully as possible given available time and resources. The best available information is used as a basis for development of planning initiatives. Resource management agencies are asked to provide information in the general categories of present use, demand, potential and capability for the resource sectors under their jurisdiction. Integrated resource planning relies heavily on existing information in order to shorten the time frame for preparation of plans. Information is applied throughout the planning process to assist in relation of identified conflicts.

4.5 Berland Sub-regional Integrated Resource Planning

A regional plan for the Eastern Slopes area was completed in 1977 and updated in 1984. The regional plan requires further refinement to meet its stated objectives (Government of Alberta, 1984).

Sub-regional integrated resource plans express the resource management policy of the Alberta Government for provincially owned resources within geographically defined planning areas. Sub-regional integrated resource plans identify desirable patterns of land and resource use. Plans provide integrated direction to field level resource managers to assist in their day-to-day management activities. Therefore, to address potentially conflicting uses the provincial valuable resources, the Berland Sub-regional Integrated Resource Planning was initiated in the summer of 1986.

The purpose of the Berland Sub-regional Integrated Resource Planning is to (Department of Forestry, Land and Wildlife, 1987):

a. identify land and resource issues,

b. provide guidance for the resolution of these issues,

c. identify resource development opportunities in this sub-region,

d. indicate the best uses for the sub-region by determining and evaluating the natural resource potential of the area, and

e. accomplish the above through integration of information received from all participants in the planning exercise.

5 METHODOLOGY

5.1 Development of the Mathematical Models of Forest System Dynamics

If forestry is defined as the utilization of forest resources and their management for the benefit of man, it certainly has a longer history than what is commonly called the oldest profession (Sweda, 1984). In accordance with this long history, through most of which forestry has been practiced empirically, forest mensuration, which provides management with vital statistics, has been using yield tables in one form or another. The methods of mensuration, accumulated empirically over centuries, have been effective and useful so far as they are mathematically easy to provide estimates of stand parameters. For example, the stand height growth rate is only a function of age. In fact, however, the increment of stand height is also a function of current stand height, [illegible] it may not follow a simple mathematical relationship. For many years there were no easy ways to approach such problems from a theoretical point of view, because of the difficulty of implementation. As the development of computer and quantitative modeling techniques, such as simulation, to study the forest dynamics by using interrelated set of growth equations will be an alternative.

The development of the mathematical models is directed at the following problem: Given observational data on a dynamic system, what are the equations governing its behavior? This is known as system identification and in its broadest sense involves both unknown algebraic form and numerical constants in the governing equations. Clearly, knowledge of the exact, or even approximate, equations governing a system makes effective control a realistic possibility. The ability to control a dynamic forest system is of concern in the many instances where man has developed preferences for certain system states and is prepared to take action to ensure their existence (Leary, 1970). The development of mathematical models of forest system dynamics is the crucial step and is usually more time consuming in the whole simulation procedure, since the adequate mathematical models make the simulation results meaningful. The acronym GIGO or "garbage in ... garbage out" is a precise statement of the situation (Ingle, 1985).

An algorithm is derived to represent timber growth using the basic biological principles that govern tree growth. In this project the Berland forest is divided, by quarter section, into a number of planning units. Only four main species, white spruce, black spruce, pine, and aspen, within each planning unit are considered. The numerical integration technique of Euler's method is used for solving the differential equations. The time interval used can be small or large. Here, the time interval is one year since trees grow annually.

The mathematical models derived here are based on the following assumptions:

a. The forest within each planning unit is even-aged and distributed uniformly.

b. The site quality within each cell remains constant over time.

c. The trees of the same species have the same taper form.

d. The height of each cell is the site height, which is the average height of dominant and codominant trees weighted by basal area.

e. The diameter of each cell is the quadratic mean diameter, that is, the dbh of the tree with average basal area for all trees above minimum stump d.o.b. (diameter outside bark).

f. The stump diameter is equal to the dbh.

5.1.1 Diameter growth model

The first phase of the study develops equations to predict diameter growth. Studies have shown that tree diameter is one of the important factors affecting tree volume growth. The methods employed are based on the work done by Leary (1973) for predicting forest dynamics.

In this study, each quarter section was considered to be a forest stand and it is changing over time. Leary (1973) describes a forest stand as an aggregate of dynamic interactive energy transformers and accumulators. This description, as opposed to some others, provides a useful frame work to derive the mathematical equations governing forest dynamics. Specifically, it shows:

a. a stand is an aggregate (set),

b. the set is dynamic,

c. the set is interactive, and

d. elements of the set function as both transformer and accumulator.

How can these help in deducing the governing equations? One way they help is as follows--consider the diameter as a dynamic variable which changes from one state (t_0) to another (t_1) as in Figure 5.1.

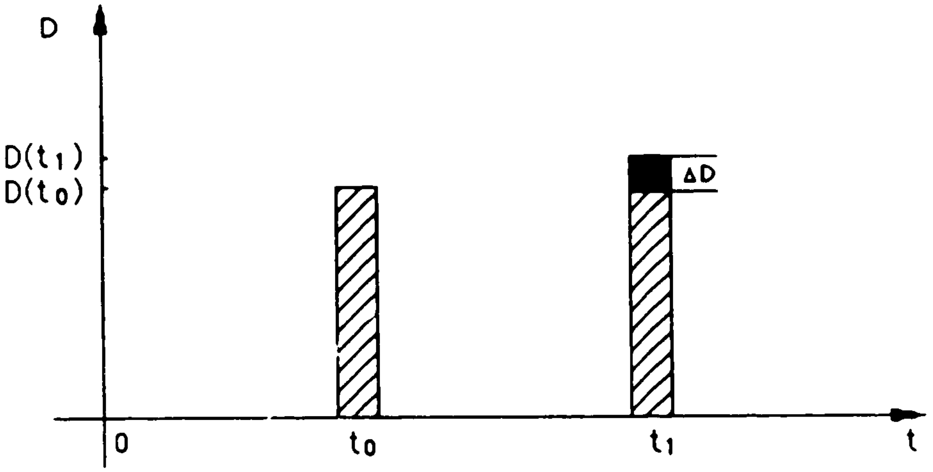


Figure 5.1 The tree diameter at time t_0 and time t_1

The mathematical relationship of diameter at two states is

$$D(t) = D(t_0) + \Delta D \tag{5.1}$$

where

ΔD is the growth in diameter from state t_0 to state t_1,

and the difference between this mathematical prediction and the real diameter growth is presented in Figure 5.2.

Thus, a necessary, but not sufficient condition for having identified the governing equations is that the ϵ (error) be small for all t.

One problem common to much of biology is that when rationalizing the

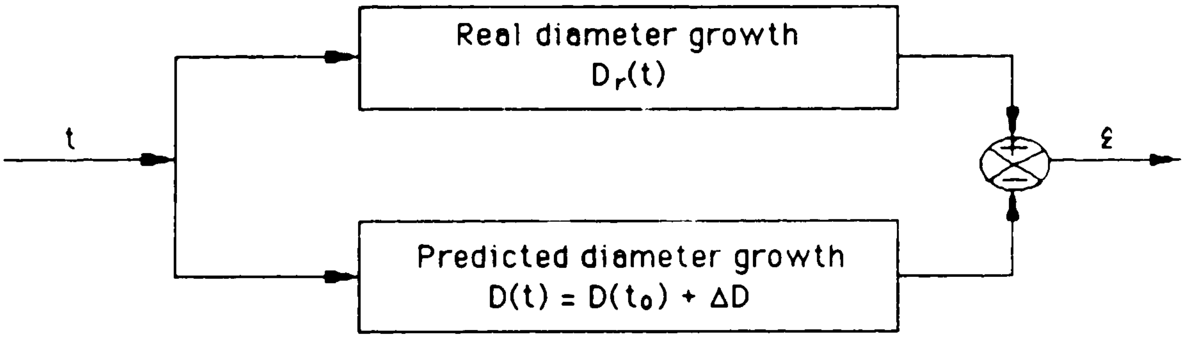


Figure 5.2 The comparison between predicted and real diameter growth

form of governing equations, "biological laws" are not known as they are in the physical sciences. There are, however, some useful biological "principles" (Leary, 1973). One is as follows:

> "That which results from biological growth is itself typically capable of growing..."

Another useful "principle" is:

> "In a fixed physical space the natural ecosystem resources necessary for plant life are limited (as is the flow into the physical space occupied by the plants)."

The third "principle" is:

> "The biological growth rate varies over time."

These principles might be graphically showed as in Figure 5.3, Figure 5.4, and Figure 5.5.

The vertical arrow, in Figure 5.3, denotes a potential increase in standing crop. In Figure 5.4, the downward slanting arrow indicates that the potential increase may not be realized. The length of the vertical arrow in Figure 5.5, presents the increment of standing crop per unit time.

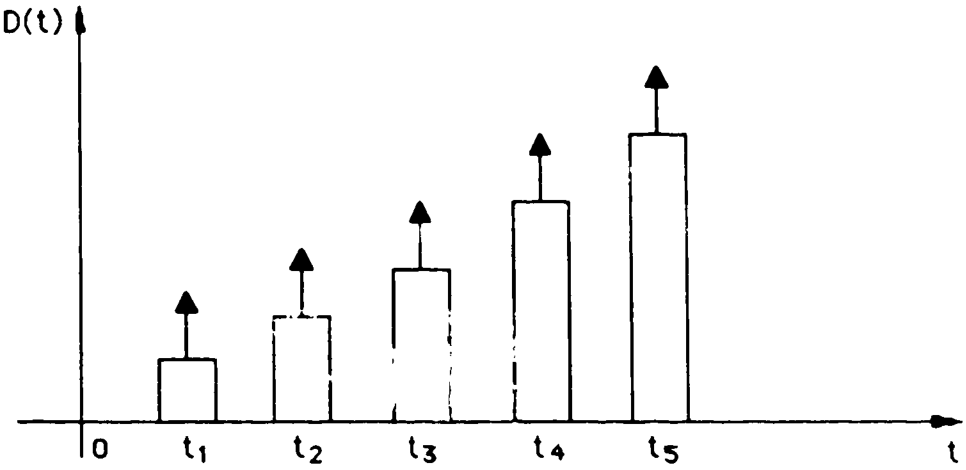


Figure 5.3 The first "principle" of standing crop growth (adapted from Leary, 1973)

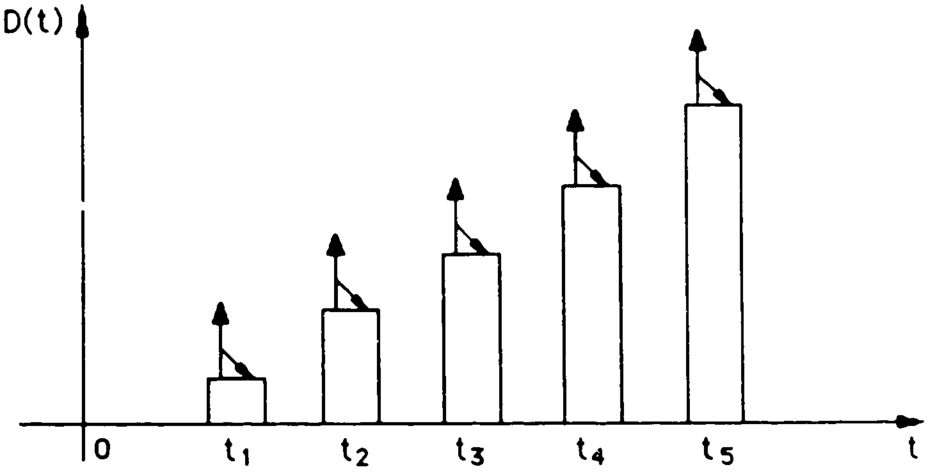


Figure 5.4 The second "principle" of standing crop growth (adapted from Leary, 1973)

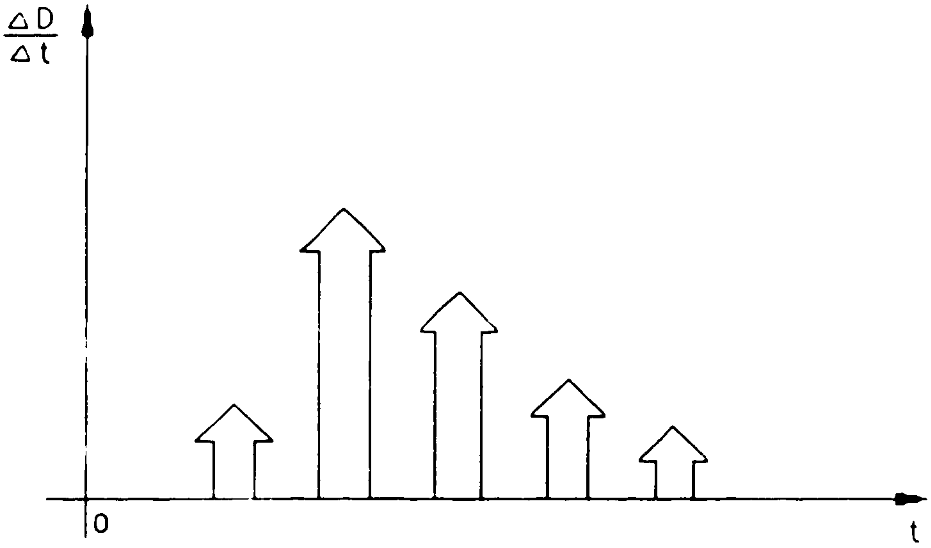


Figure 5.5 The third "principle" of standing crop growth

Based on these "principles", the basic mathematical model of diameter growth might be derived. For simplicity, consider the length of the arrow to be indicative of the rate at which diameter (D) is increasing, then the rate of growth may be expressed as follows:

$$diameter\ \ growth\ \ rate = g(D.t) \tag{5.2}$$

or

$$\frac{\Delta D}{\Delta t} \approx \frac{dD}{dt} = g(D.t). \tag{5.3}$$

The natural question at this point is "what form should the function $g(D.t)$ take?" Again by applying the three biological "principles" it is possible to identify the potential forms of the function $g(D.t)$. First:

> "That which results from biological growth is itself typically capable of growing..." (Figure 5.6)

that is

$$\frac{dD}{dt} = a \cdot D \tag{5.4}$$

where

a is a constant.

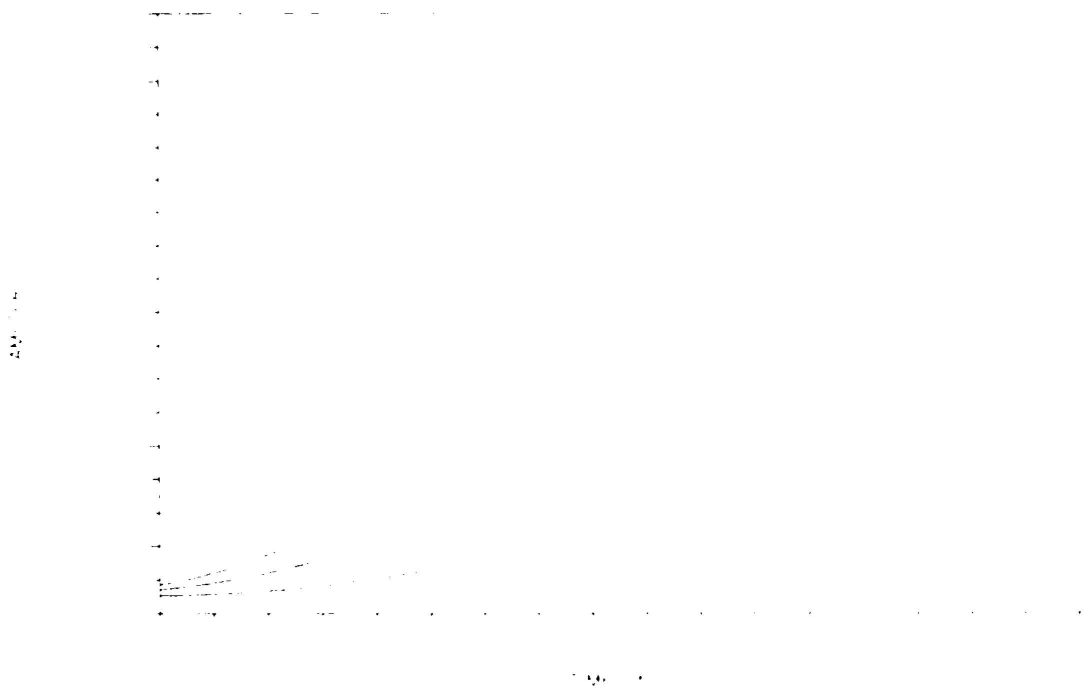


Figure 5.6 The potential diameter growth curve derived by using the first "principle" (adapted from Leary, 1973)

Second:

> "In a fixed physical space the natural ecosystem resources necessary for plant life are limited." (Figure 5.7)

that is

$$\frac{dD}{dt} = p(D) \cdot D \tag{5.5}$$

where

$p(D)$ = a function of diameter D.

A question which has received considerable attention in the literature concerns the algebraic form of $p(D)$ (Leary, 1973; Clutter et al., 1983):

$$\frac{1}{D} \cdot \frac{dD}{dt} = p(D) \tag{5.6}$$

Possible forms for $p(D)$ are the following:

a. The logistics function

$$\frac{dD}{dt} = -aD^2 + bD \tag{5.7}$$

Figure 5.7 The potential diameter growth curve derived by using the second "principle" (adapted from Leary 1973)

$$p(D) = -aD + b \tag{5.8}$$

where

a and b are constants.

b. The Chapman-Richards function

$$\frac{dD}{dt} = aD^{c} - bD \tag{5.9}$$

$$p(D) = aD^{c-1} - b \tag{5.10}$$

where

a, b and c are constants.

c. The Mitscherlich function

$$\frac{dD}{dt} = k(M - D) \tag{5.11}$$

$$p(D) = kMD^{-1} - k \tag{5.12}$$

where

k and M are constants.

d. The exponential function

$$\frac{dD}{dt} = aD(1 - e^{b/D}) \tag{5.13}$$

$$p(D) = a(1 - e^{b/D}) \tag{5.14}$$

where

a and b are constants

e = base of natural logarithms.

In many cases, $p(D)$ should "begin large and then decrease," such as in Figure 5.8.

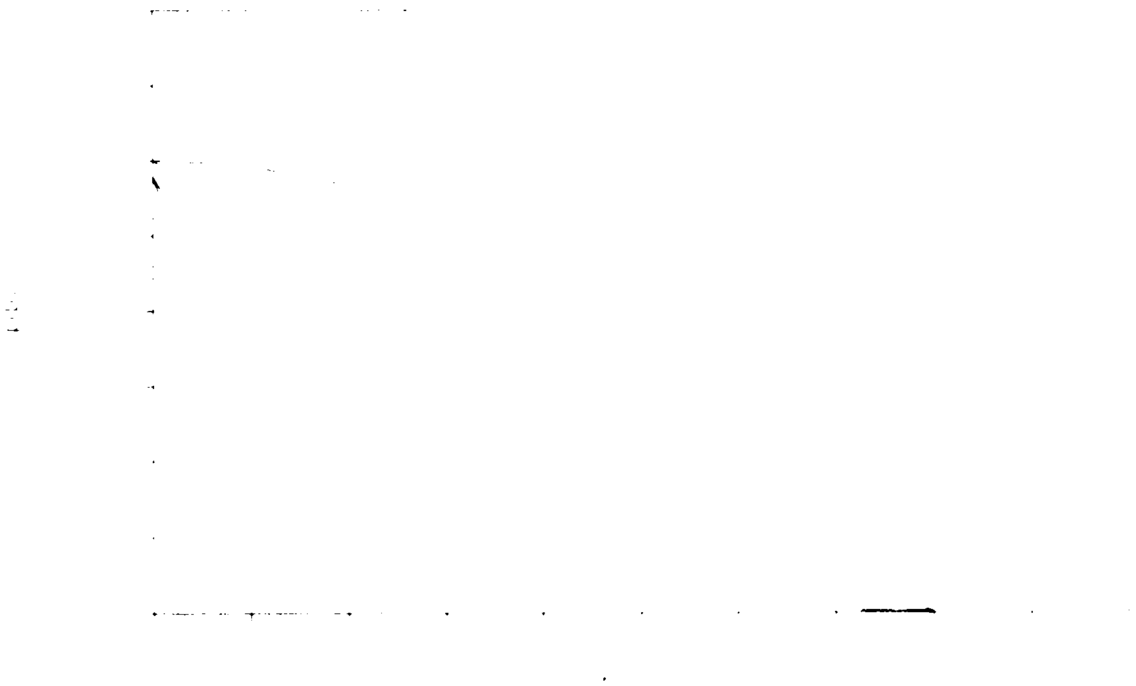


Figure 5.8 The potential curves of function $p(D)$ (adapted from Leary, 1973)

The discussion to this point has dealt with some of the simplest types of differential equation models; i.e., those with constant coefficients which are time (age) independent. According to the third "principle" (i.e. that biological growth rate varies over time), the equation coefficients are thought to be time (age) dependent. A case in point is the so-called generalized logistics equation (Leary, 1973):

$$\frac{dD}{dt} = -R(t)D^{k(t)} + Q(t)D \tag{5.15}$$

For simplicity, assuming that only coefficient Q and R are time dependent, the diameter growth equation can be formulated as follows:

$$\frac{dD}{dt} = -R(t)D^{k} + Q(t)D \tag{5.16}$$

This formulation of mathematical models, where a potential -for change-component is multiplied by a time dependent modifier to give a realized change, is now becoming more widely used in modeling forest growth. It is, of course, an old idea, dating back at least to Gause (1934) and Lotka (1925).

The above discussions involved only the natural diameter growth of trees and none of the management activities, such as thinning. These operations, however, play important roles in managing the forest resource to maximize the management objectives. Although thinning can certainly be used for other objectives of managing forest vegetation, timber production is the chief purpose.

Empirical evidence from thinning experiments has consistently indicated increases in diameter growth with decreasing stand density (Johnstone, 1981, 1982). Stands with wider spacings or stands previously thinned, in time, have larger average diameters than similar stands with closer spacings or comparable unthinned stands (Clutter, et al. 1983). Therefore, stand density is one of main factors affecting the diameter growth. There are several parameters which can be considered as measures of stand density, such as the number of trees or the amount of basal area, wood volumes, leaf cover, or any of a variety of less common parameters (West, 1983). The simplest parameter is number of trees per unit area. But, this takes no account of either the sizes of trees or the space they occupy. Therefore, number of trees per unit area may not be used in the diameter growth model as a measurement of stand density since it cannot reflect the effects of different thinning methods, e.g., low thinning versus crown thinning. Furthermore, it may confuse the thinning effect with the effect of density changes on subsequent stand diameter growth.

Basal area is by far the most commonly used parameter although more tradition may be involved for this practice than any biological reason. Its main virtue is that it is a kind of integrated expression of numbers of trees and their sizes (Smith, 1986).

The thinning effect on diameter development can be graphically illustrated as in Figure 5.9:

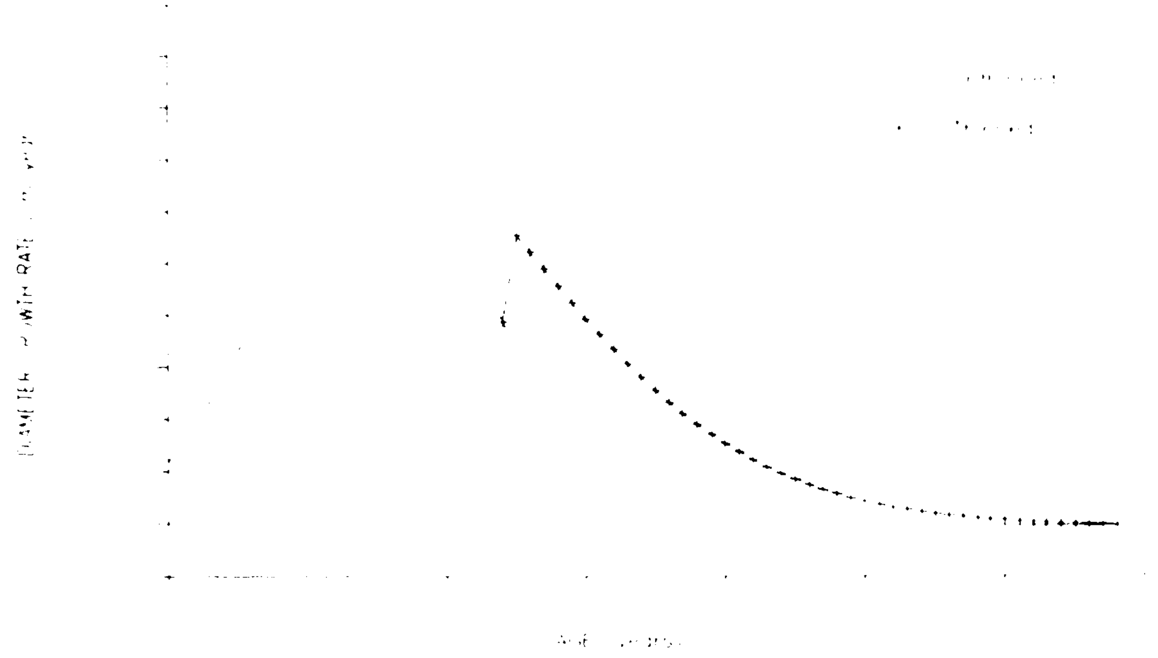


Figure 5.9 Possible thinning effect on diameter growth

The functional form for this relationship may be obtained by including the ratio of basal area before thinning and basal area after thinning in the equation (5.16), that is:

$$\frac{dD}{dt} = -R(t)D^2 + Q(t)D\,\frac{BA_b}{fBA_a} \qquad (5.17)$$

where

BA_b = the basal area before thinning,

BA_a = the basal area after thinning, and

f = model parameter, depending on the thinning method used.

The equation (5.16) and equation (5.17) give the increment rate of diameter of a stand. To obtain the diameter at certain time t, the equation has to be integrated. However, this equation can not be integrated by common mathematical methods. One way to solve such a problem is to obtain an approximation to the solution through the use of the numerical integration methods and computers. The most widely used numerical integration methods are Euler (Rectangular), Simpson, and Runge-Kutta. The selection of the method depends on the use of computing time and the requirement for accuracy. In this study, Euler's method is used because of the following considerations:

a. Since this is a prototype numerical simulation model, it is easy and reasonable to run initially using the simplest Euler's method.

b. The Euler's method involves evaluating only the first derivative once per time step and requires less calculations, therefore, it can result in less amount of computing time to run the model on a microcomputer than other integration methods.

c. The simulation using this method with a proper time intervals can provide sufficient accuracy for most of biological systems (France and Thornley, 1984).

Consider the mathematically unsolvable function $D(t)$ graphed in Figure 5.10. Assuming that an initial value of the function $D(t)$ at time n (D_n) and the instantaneous rate of change at time n (dD_n/dt) are known, the objective of numerical approximation is to estimate the value of $D(t)$ at some future time $n+1$ (D_{n+1}). This estimate is calculated by assuming that the instantaneous rate of change at time n is representative of the rate of change over the entire finite time interval (Δt) from t_n to t_{n+1} Graphically this means that the tangent to $D(t)$ at point (t_n, D_n) Δt is projected into the future to t_{n+1}, thus obtaining an estimate of D_{n+1} -- call it D_{n+1} If dD/dt actually is changing from t_n to t_{n+1} as is the case in Figure 5.10, the estimate of D_{n+1} is wrong. However, the size of the error can be controlled by adjusting the size of Δt. The smaller Δt is the smaller the error will be. This procedure of projecting the instantaneous rate of change over a finite time interval to estimate the value of the function in the future is repeated.

Thus, by the Euler's method, the diameter equation can be described as:

$$D(t) = D(t_0) + \Delta D \approx D(t_0) + \Delta t \frac{dD}{dt} \tag{5.18}$$

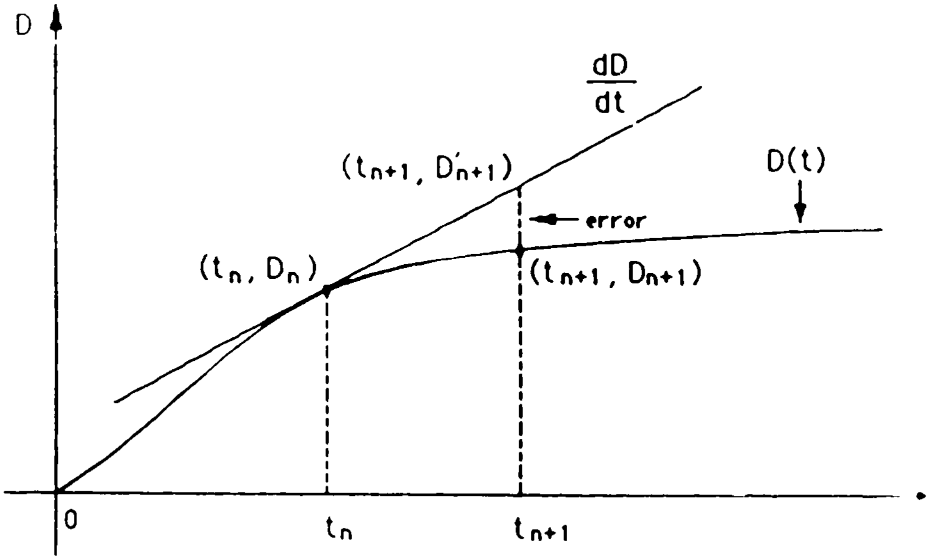


Figure 5.10 Graphical representation of the Euler's method of numerical approximation for differential equations.

5.1.2 Height growth model

Tree height is another important variable affecting tree volume growth. Since the height growth shows the same trend as diameter growth (Husch, et al., 1982), the height growth function may be assumed to follow the same growth equation. The only difference from diameter growth is that variations in stand density, such as those induced by thinning, cause remarkable little change in height growth (Smith, 1986). Thus, the mathematical model of height growth is obtained as

$$\frac{dH}{dt} = -N(t)H^2 + M(t)H \tag{5.19}$$

and

$$H(t) = H(t_0) + \Delta H = H(t_0) + \Delta t \frac{dH}{dt} \tag{5.20}$$

where

$M(t)$ and $N(t)$ are model coefficients which are time dependent.

5.1.3 Mortality model

At a certain point in time, the number of trees is finite, and each tree is either alive or dead. The dimensions of the living trees in the stand are changing continuously. Hence, if the rate of change in dimensions at the given point in time can be estimated, these estimates can be applied to obtain the current rate of change in net volume produced

In homogeneous all-aged stands and in unthinned even-aged stands of given age and site quality, the average number of trees per hectare is a useful measure of stand density. It is a necessary variable in the development of volume prediction equations. The change in the number of trees in a stand is usually expressed by mortality functions. The development of empirical mortality prediction equations generally requires data from the remeasurement of monumented plots. However, some kinds of mortality rate prediction equations may be derived from basic biological considerations. One such equation, derived by Clutter et al. (1983) relates to age and site index in the following way

$$\frac{dN}{dt} = \left(\beta_0 + \frac{\beta_1}{t} + \beta_2 S\right)N = \left(\beta + \frac{\beta_1}{t}\right)N \tag{5.21}$$

where

N = number of trees per unit area at age t,

S = site index,

β_0, β_1 and β_2 are constants, and

$\beta = \beta_0 + \beta_2 S$.

Past work (Leak, 1969a and 1969b) has also shown that numbers of trees of a stand decrease as the stand average diameter D_a increases according to a negative exponential distribution, which has the form:

$$N = ke^{aD_a} \tag{5.22}$$

and

$$\frac{dN}{dt} = -k' e^{k e_1}\left(\frac{dD_a}{dt}\right) \qquad (5.23)$$

where

k is a constant.

In addition, some previous studies on tree mortality have also indicated that the average diameter growth rate dD_a/dt and the mortality rate are highly correlated. Yang (1988) has developed such a generalized logistic mortality model, which can be expressed as

$$\frac{dN}{dt} = -\frac{N}{1 + e^{a_1 + a_2 D_a + a_3 \frac{dD_a}{dt}}} \qquad (5.24)$$

where

D_a = average diameter,

dD_a/dt = increment of average diameter, and

a_1, a_2 and a_3 are constants.

By the Euler's method, the number of trees can be expressed as

$$N(t) = N(t_0) + \Delta t\frac{dN}{dt} \qquad (5.25)$$

5.1.4 Basal area model

Basal area is the cumulative cross-sectional area per hectare, of tree stems at breast height (1.3 m above ground level). It is a useful characteristic of a forest stand. For example, basal area is directly related to stand volume and is a good measure of stand density (Husch, et al., 1982).

As with number of trees per hectare, the utility of basal area per hectare as an indicator of crowding is limited when the prior stand history is unknown. However, when used in the proper context, such as for a given age and site in unthinned even-aged stands (like the Berland forest) or plantations, or in all-aged stands with a reasonably stable age distribution, basal area has proved useful in volume estimation. Since basal area per hectare and the number of trees per hectare specify average tree size, the use of both will often give improved volume estimates in comparison to those obtainable with the use of only one of these measures (Clutter, et al., 1983).

Basal area, b_i in square meters (m^2) for a tree of breast height diameter d_i cm is

$$b_i = \frac{\pi d_i^2}{4(10000)} = 0.00007854 d_i^2 \tag{5.26}$$

and the total basal area B for a stand of N trees is

$$B = \sum_{i=1}^{N} b_i = 0.00007854 \sum_{i=1}^{N} d_i^2 \tag{5.27}$$

with a mean per-tree basal area of

$$\frac{B}{N} = \frac{1}{N} \sum_{i=1}^{N} b_i = 0.00007854 \sum_{i=1}^{N} d_i^2 / N \tag{5.28}$$

The quadratic mean breast height diameter D is defined as

$$D = \sqrt{\frac{1}{N} \sum_{i=1}^{N} d_i^2} \tag{5.29}$$

so that

$$\frac{B}{N} = 0.00007854 D^2 \tag{5.30}$$

and

$$B = 0.00007854 N D^2 \tag{5.31}$$

This equation can be used to calculate basal area but the results could be overestimated since diameter and number of trees are estimated using Euler's method. Therefore, basal areas may be integrated from the increment rates which can be expressed as a function of diameter growth rate and mortality rate. If the equation (5.31) is differentiated with respect to time t, the increment rate of the basal area of a stand is

$$\frac{dB}{dt} = 0.00007854 \left(2DN \frac{dD}{dt} + D^2 \frac{dN}{dt} \right) \tag{5.32}$$

and the basal area at time (age) t is

$$B = \int_0^B dB = \int_0^t \left(0.00007854 \left(2DN \frac{dD}{dt} + D^2 \frac{dN}{dt} \right) \right) dt \tag{5.33}$$

By Euler's method, the above equation can be expressed approximately as

$$B(t) = B(t_0) + \Delta t \frac{dB}{dt} \tag{5.34}$$

where

$B(t_0)$ is the initial basal area at time t_0, and

Δt is the time increment.

5.1.5 Form factor model

Volume is a statistical estimate of the production of a forest stand. To make more accurate estimates, studies have been done on developing a physical relation between the tree volume and stand variables such as diameter and height. Because a tree diameter varies from the base to the tip, there are no simple ways to calculate a tree volume based on general geometrical solid forms. Thus, the most commonly used method is using regression techniques to define relations between volume and other stand variables. In this study, however, another method of employing tree form factors was applied.

As described by Husch et al. (1982), a form factor is the ratio of tree volume to the volume of a geometrical solid, such as a cylinder, a cone, or a cone frustum, that has the same diameter and height as the tree. (The diameter of the geometrical solid is taken at its base; the diameter of the tree is taken at stump or breast height.) A form factor is different from other measures of form in that it can be calculated only after the volume of the tree is known. In formula form the form factor F is

$$F = \frac{\text{volume of tree}}{\text{volume of geometrical solid of same diameter \& height}} \tag{5.35}$$

Early in the nineteenth century the form of tree stems was recognized as approaching that of the solids mentioned above. But that there were many variations in form, and that a tree was rarely of the exact form of one of these solids were also recognized. Thus, the form factor was conceived as a method of coordinating form and volume. That is to say, the main objective of the early work was to derive factors that would be independent of diameter and height, and by which the volume of standard geometrical solids could be multiplied to obtain the tree volume (Husch et al., 1982).

The cylindrical form factor, which has been widely used, may be expressed by the equation

$$F = \frac{V}{\pi d^2 h} \tag{5.36}$$

where

V - volume of tree, m^3,

d - diameter of cylinder whose diameter equals tree diameter, m, and

h - height of cylinder whose height equals tree height, m.

The results of analyzing the data provided by the "Alberta Phase 3 Forest Inventory: Yield Tables for Unmanaged Stands" (1985) show that the form factors decrease as the ages of trees increase. That is, a form factor is a function of the age of tree. This function can be linear or non-linear, depending upon several factors. For simplicity, the mathematical relationship between a form factor and age of a tree was assumed to be a linear function

$$F = at + b \tag{5.37}$$

where

F - form factor,

t - age of tree, and

a and b are constants.

5.1.6 Volume model

Volume is an important variable of forest dynamics because it is a integrated measure of stand variables. Volume increment rates have been expressed in terms of net total volume growth. The periodic net increment of volume is the subtraction of periodic gross increment and periodic mortality. The rate of change of net volume with respect to time (dV/dt) is, approximately,

$$\frac{dV}{dt} \approx \frac{\Delta V}{\Delta t} = \frac{(periodic\ gross\ increment) - (periodic\ mortality)}{(periodic\ length)} \tag{5.38}$$

Since geometrically, the volume of a stand is a function of its height diameter, number of stems, and form factor, this rate can be expressed as follows:

$$\frac{dV}{dt} = f(height, diameter, form\,factor, stems) \tag{5.39}$$

$$dV = f(height, diameter, form\,factor, stems)dt \tag{5.40}$$

Then, for a certain form factor, and with height, diameter, and number of stems as variables expressible as functions of age and previous state, the cumulative net volume (V) is closely approximated by the integral of the rate equation, evaluated from zero to the present age of the stand:

$$V = \int_0^t dV = \int_0^t f(height, diameter, form\,factor, stems)dt \tag{5.41}$$

The problem now is to identify the form which the function f can take. If v is the average volume of individual tree at time t, then it may be expressed mathematically as the following equation.

$$v = \frac{\pi F D^2 H}{4} \tag{5.42}$$

where

F = average form factor,

D = quadratic mean diameter of the trees in the stand, and

H = site height of live trees in the stand.

Thus, for a stand having N stems, the total volume of live trees is:

$$V = \frac{\pi N F D^2 H}{4} \tag{5.43}$$

When the above equation is differentiated with respect to t,

$$\frac{dV}{dt} = \frac{\pi}{4} N F D^2 \left(\frac{dH}{dt}\right) + \frac{\pi}{2} N F D H \left(\frac{dD}{dt}\right) + \frac{\pi}{4} F D^2 H \left(\frac{dN}{dt}\right) \tag{5.44}$$

In this equation, $(\pi F D^2 H/4)$ is the volume per tree; dN/dt is the mortality rate (<0), the change in number of trees per unit time; and $(\pi F D^2 H/4)(dN/dt)$ is the volume of mortality per unit time (a negative quantity). Therefore, the function f may be written in the form:

$$f = \frac{dV}{dt} = \frac{\pi}{4}\left(N F D^2 \left(\frac{dH}{dt}\right) + 2 N F D H \left(\frac{dD}{dt}\right) + F D^2 H \left(\frac{dN}{dt}\right)\right) \tag{5.45}$$

and,

$$V = \frac{\pi}{4}\int_0^t \left(N F D^2 \left(\frac{dH}{dt} \right) + 2 N F D H \left(\frac{dD}{dt} \right) + F D^2 H \left(\frac{dN}{dt} \right) \right) dt \tag{5.46}$$

By Euler's method, the total volume V can be expressed as:

$$V(t) = V(t_0) + \Delta t f(H, D, N, t) = V(t_0) + \Delta t \frac{dV}{dt} \tag{5.47}$$

where

$V(t_0)$ is the initial stand volume at time t_0, and

Δt is the time increment.

5.1.7 Determination of the model coefficients

By summarizing the equations derived above, the mathematical models of forest dynamics are the following equations:

a. Diameter

$$\frac{dD}{dt} = -R(t) D^2 + Q(t) D \tag{5.48}$$

$$D(t) = D(t_0) + \Delta t \frac{dD}{dt} \tag{5.49}$$

b. Height

$$\frac{dH}{dt} = -N(t) H^2 + M(t) H \tag{5.50}$$

$$H(t) = H(t_0) + \Delta t \frac{dH}{dt} \tag{5.51}$$

c. Number of trees

$$\frac{dN}{dt} = -\frac{N}{1 + e^{a_0 + a_1 D + a_2 H}} \tag{5.52}$$

$$N(t) = N(t_0) + \Delta t \frac{dN}{dt} \tag{5.53}$$

d. Basal area

$$\frac{dB}{dt} = 0.00007854 \left(2 D N \left(\frac{dD}{dt} \right) + D^2 \left(\frac{dN}{dt} \right) \right) \tag{5.54}$$

$$B(t) = B(t_0) + \Delta t \frac{dB}{dt} \tag{5.55}$$

e. Volume

$$\frac{dV}{dt} = \frac{\pi}{4}\left(N F D^2\left(\frac{dH}{dt}\right) + 2 N F D H\left(\frac{dD}{dt}\right) + F D^2 H\left(\frac{dN}{dt}\right)\right) \qquad (5.56)$$

$$V(t) = V(t_0) + \Delta t \frac{dV}{dt} \qquad (5.57)$$

Those growth equations are the general representation of forest dynamics used in this research. For different regions and different species, the coefficients of the equations will differ. To develop an appropriate growth model for the study area, the Alberta Phase 3 Forest Inventory Yield Tables for Unmanaged Stands (1985) were used to determine the coefficients of the mathematical models. Since the possible mathematical forms of $R(t)$ and $Q(t)$ could not be found in the literature and since representative functions could not be derived using a single equation over the time period required, the piecewise approach was employed. Theoretically, the growth rate curve can be plotted as shown in Figure 5.11.

To determine the mathematical forms of $R(t)$ and $Q(t)$, let us consider the special case where $R(t)$ and $Q(t)$ are constants. In this case, the equation (5.48) becomes

$$\frac{dD}{dt} = -R D^2 + Q D \qquad (5.58)$$

and graphically it can be shown as in Figure 5.12. Obviously, the comparison of Figure 5.11 and 5.12 shows that equation (5.58) with constant coefficients R and Q cannot be used to represent the changes of growth rate over the whole growth period. However, it may be used to mimic part of the growth rate curve by dividing the curve shown in Figure 5.11 into several parts and representing each part using the equation (5.58) with different constant coefficients R and Q. Thus, $R(t)$ and $Q(t)$ are discrete function of time for the whole curve and are constants for each part of the curve. Based on the characteristics of the growth rate curve and yield table data, the equation (5.58) may be rewritten as

$$\frac{dD}{dt} = -R_1 D^2 + Q_1 D \qquad (0 \le t < t_1)$$

$$\frac{dD}{dt} = -R_2 D^2 + Q_2 D \qquad (t_1 \le t < t_2)$$

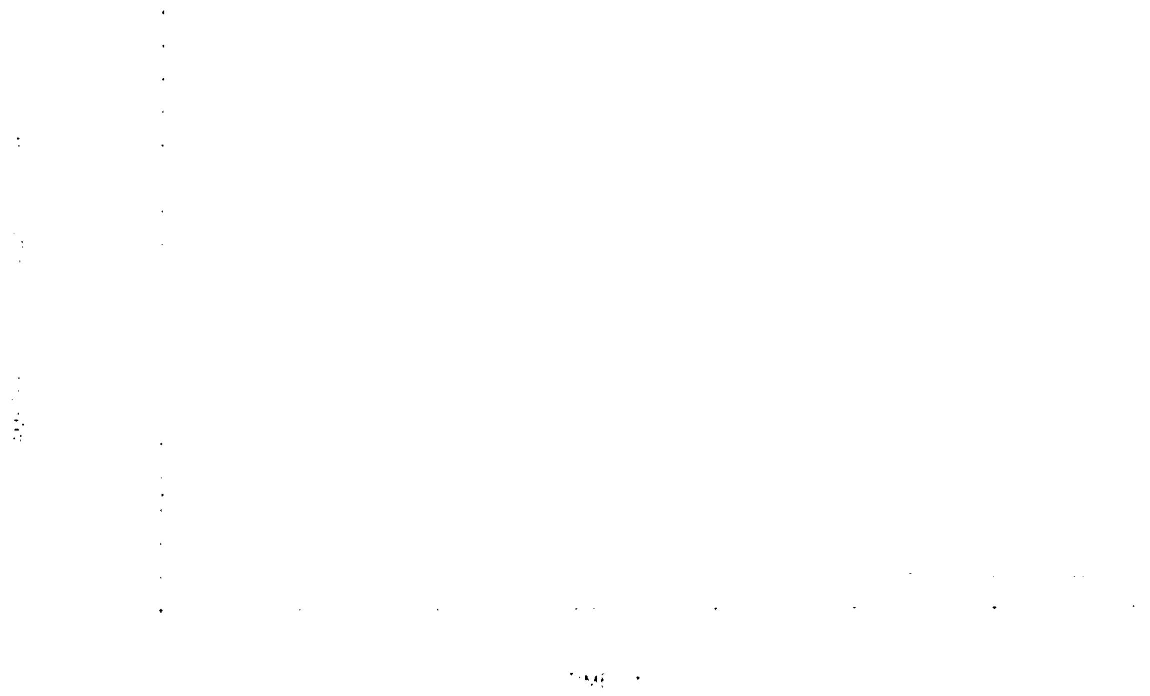


Figure 5.11 The theoretical diameter growth rate curve

Figure 5.12 The diameter growth rate curve represented by the equation (5.58)

$$\frac{dD}{dt} = -R_3 D^2 + Q_3 D \qquad (t_2 \leq t < t_3)$$

$$\frac{dD}{dt} = C \qquad (t \geq t_3) \qquad (5.59)$$

where

t_1 = 20 (or 30),

t_2 = 90,

t_3 = 170, and

C is a constant.

A growth rate curve developed by using the above equation is shown in Figure 5.13

Figure 5.13 The diameter growth rate curve represented by the equation (5.59)

The shape of the curve may be improved by dividing the growth period into more parts. For simplicity, the growth rate curve was split into four parts in this study. The coefficient *R* and *Q* for each part of c were estimated by the following steps:

1. For each part, find the quadratic mean diameter and diameter growth rate at starting point (t_0) and ending point (t_1), D_0 (D_1) and dD_0/dt (dD_1/dt).
2. Replace dD/dt and D in equation (5.58) using dD_0/dt, D_0 and dD_1/dt, D_0 respectively to get the following equations,

$$\frac{dD_0}{dt} = -RD_0^2 + QD_0$$

$$\frac{dD_1}{dt} = -RD_1^2 + QD_1$$

3. Solve the above equations, then

$$R = \frac{\frac{dD}{dt} D_1 - \frac{dD}{dt} D_0}{D_1^2 D_0 - D_0^2 D_1}$$

$$Q = \frac{\frac{dD}{dt} + RD_0^2}{D_0}$$

The predicted coefficients for equation (5.48) is shown in Table 5.1. Similarly, the coefficient N and M of height growth model may be estimated as shown in Table 5.2. As for the coefficients of mortality models, they were determined by trail and error approach using equation (5.48), (5.49), (5.52), and (5.53) and based on the coefficients of the model developed by Yang (1988). The initial densities not available in the AFS Yield Tables were estimated by the author from the data in the table. The results are listed on Table 5.3.

Table 5.1 THE PREDICTED COEFFICIENTS OF THE DIAMETER GROWTH MODELS

Species-Site	R_1	Q_1	R_2	Q_2	R_3	Q_3
White Spruce-G	0.09048	0.32589	0.00299	0.07479	0.00030	0.01288
White Spruce-M	0.37531	0.51259	0.00513	0.09430	0.00027	0.01043
White Spruce-F	0.06519	0.18860	0.00693	0.08606	0.00045	0.01309
Black Spruce-G	0.21807	0.39876	0.00528	0.08169	0.00041	0.01217
Black Spruce-M	0.41250	0.48250	0.00790	0.07790	0.00074	0.01444
Black Spruce-F	0.25735	0.31287	0.02158	0.11246	0.00107	0.01564
Pine-G	0.05971	0.27999	0.00328	0.07908	0.00026	0.011[illegible]
Pine-M	0.05331	0.20031	0.00519	0.09107	0.00035	0.011[illegible]6
Pine-F	0.12508	0.23041	0.00641	0.07852	0.00053	0.01221
Aspen-G	0.26492	0.74614	0.00257	0.07714	0.00025	0.01177
Aspen-M	0.40373	0.73075	0.00392	0.08706	0.00032	0.01104
Aspen-F	0.87912	0.88791	0.00590	0.09328	0.00044	0.01097

Table 5.2 THE PREDICTED COEFFICIENTS OF THE HEIGHT GROWTH MODELS

Species-Site	N_1	M_1	N_2	M_2	N_3	M_3
White Spruce-G	0.01981	0.16952	0.00249	0.06720	0.00040	0.01590
White Spruce-M	0.01687	0.14114	0.00264	0.05585	0.00053	0.01786
White Spruce-F	0.00893	0.08084	0.00307	0.04535	0.00075	0.02014
Black Spruce-G	0.06715	0.20682	0.00361	0.06702	0.00041	0.01288
Black Spruce-M	0.03526	0.13534	0.00415	0.05229	0.00063	0.01521
Black Spruce-F	0.01179	0.06213	0.00422	0.03783	0.00111	0.01825
Pine-G	0.01657	0.14265	0.00319	0.07638	0.00042	0.01341
Pine-M	0.02760	0.1 399	0.00364	0.06963	0.00033	0.00989
Pine-F	0.02383	0.11960	0.00498	0.07322	0.00092	0.01946
Aspen-G	0.23453	0.90369	0.00313	0.08453	0.00036	0.01242
Aspen-M	0.28577	0.80715	0.00306	0.06930	0.00066	0.01857
Aspen-F	0.50335	1.05033	0.00421	0.07203	0.00057	0.01319

Table 5.3 THE PREDICTED COEFFICIENTS OF THE MORTALITY MODELS

Species-Site	a_0	a_1	a_2
White Spruce-G	3.2700	0.0655	-1.5290
White Spruce-M	3.3250	0.0900	-3.2000
White Spruce-F	3.5080	0.1110	-4.0000
Black Spruce-G	3.2450	0.1450	-6.0000
Black Spruce-M	2.2500	0.2800	-2.8000
Black Spruce-F	3.5000	0.1700	-7.4500
Pine-G	3.4800	0.0700	-3.5000
Pine-M	3.2500	0.1250	-4.7300
Pine-F	3.1080	0.1500	-5.2000
Aspen-G	3.1050	0.1350	-3.8000
Aspen-M	3.0750	0.1600	-4.4000
Aspen-F	3.1560	0.1600	-5.4000

Table 5.4 lists the form factors of four species by stand quality. They were predicted by using equation (5.36) and (5.37) and the data in the "Alberta Phase 3 Forest Inventory: Yield Tables for Umanaged Stands" (Alberta Forest Service, 1985). The procedure used was:

a. Determine gross volume, quadratic mean diameter, and site height at age t_1 and t_2 from the AFS Yield Tables.

b. Calculate F_1 and F_2 using equation (5.36).

c. Replace F and t in equation (5.37) by F_1 (F_2) and t_1 (t_2), that is

$$F_1 = at_1 + b$$

$$F_2 = at_2 + b$$

d. Using those two equations, then

$$a = \frac{F_1 - F_2}{t_1 - t_2}$$

$$b = F - at$$

Table 5.4 THE FORM FACTORS OF WHITE SPRUCE, BLACK SPRUCE, PINE, AND ASPEN

Species	Site		
	Good	Medium	Fair
White Spruce	-0.00067t+0.41	-0.01t+0.5	-0.0018t+0.708
Black Spruce	-0.00078t+0.41	-0.0012t+0.55	-0.0019t+0.75
Pine	-0.00047t+0.44	-0.00059t+0.5	-0.0012t+0.6
Aspen	-0.00046t+0.41	-0.00053t+0.46	-0.00068t+0.53

5.2 Development of the Mathematical Models of Other Resources

There are several non-timber values or activities that must incorporated in an integrated resource plan. These activities may be incorporated separately or they may be in combined form. Most often they are combined as they tend to be interactive, for example hunting and wildlife (Marlow, 1985). The activities considered in this study are the following:

a. recreation,

b. grazing,

c. wildlife, and

d. watershed.

To develop a simulation model for total integrated resource planning, the mathematical relationships of these increments with timber growth need to be established.

5.2.1 Recreation

Recreation is a vital part of integrated resource planning. It is, probably, the one non-timber value which is most readily identified with by the general public. Recreational use of forest land areas has increased dramatically in recent decades. A number of forest management activities alter the forest and impact on the recreation activities. The most criticized management activity is timber harvesting, which has been considered the most harmful to forest recreations. However, harvesting operations are not totally bad. Construction of logging roads may improve the access to wilderness and provide routes for skiing and hiking (Chappelle 1988). Quantitatively approaching these problems may be useful for resource managers to select the "best" management strategies.

FORESTRY AND NATURAL SCENIC BEAUTY

A major concern of forest land managers is the visual impact that activities, such as timber harvest and road building, have on the forest landscape. Management of the visual resource is a regular part of planning and many public and private forest land managers are increasing their efforts to protect and enhance this resource.

People unfamiliar with sound forestry and logging practices have generally been critical of logging. In fact, logging only causes a temporary loss of natural beauty (Cliff, 1965). A number of studies have been done that attempt to measure and compare visual quality changes after harvesting. Many psychometric techniques have been used in attempting to measure viewers' responses (Arthur and Boster 1976). The technique used in these studies is the Scenic Beauty Estimation (SBE) Technique (Daniel and Boster, 1976). This procedure consists of showing a series of randomly selected slides of an area to panels of viewers who make a numerical rating between 0 (dislike) and 9 (like). An SBE score and a mean rating (raw arithmetic mean) for each scene are developed from these ratings.

The SBE score is a sophisticated measure of viewers' response based on mathematical transformations that take into account the fact that some viewers use the rating scale differently than others. For homogeneous groups of observers, the mean ratings and the SBE scores are usually closely related. Benson and Ullrich (1981) used mean ratings in a study measuring public response to various types of timber harvesting activities. They hypothesized that changes in viewer ratings after harvest are related to growth in vegetation, since this is the principal visual change on the site. The results of their study for lodgepole pine (LPP) and Douglas-fir/Larch (DF/L) after harvest in clearcuts are shown in Figure 5.14.

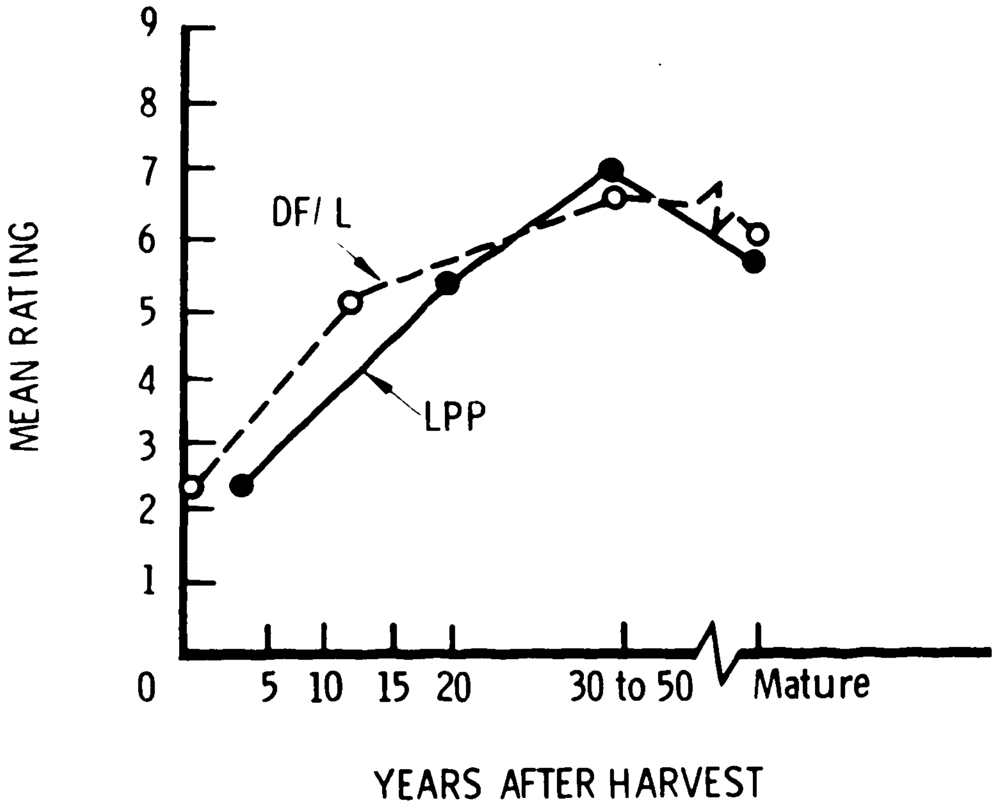


Figure 5.14 Esthetic rating over time following clearcutting (adapted from Benson and Ullrich, 1981)

Both the Douglas-fir/Larch and the lodgepole pine harvest areas were rated low initially. About 10 years after harvest the DF/L had reached a point on the "like" portion of the scale, probably because these stands are on moist sites that "green up" quickly and trees begin to grow rapidly. LPP took longer to reach this point reflecting the generally sparser vegetation and slower tree growth. When stands had reached heights of about 25 to 75 feet and crowns were green and vigorous, ratings were the highest. In mature stands aged 150 years or more, rating were lower. This is speculated to be due to more dead material and debris, and also that the mature stands with a high dense canopy are darker and more enclosed than younger stands.

Since the regeneration species in the study area are pine and white spruce, the scenic beauty model can be derived based on work by Benson and Ullrich (1981). Let mean rating (MR) of a unit area be expressed by

$$MR = f(A) \qquad (5.60)$$

where

A = age from regeneration.

The equation form was assumed to be

$$MR = aA + b \qquad (5.61)$$

where

a and b are constants.

Considering the site quality of the Berland forest, it may be assumed that young trees begin to be distinguishable at age 15, reach the highest rating at age 70, and are mature at age 180. Then the equation (5.61) can be represented by the following three equations

$$MR = \frac{2}{15}A \quad (A < 15) \qquad (5.62)$$

$$MR = \frac{1}{11}(A - 15) + 2 \quad (15 \le A \le 70) \qquad (5.63)$$

$$MR = -\frac{1}{55}(A - 70) + 7 \quad (70 < A < 180) \qquad (5.64)$$

HUNTING

Sport hunting is one of most popular wildland recreation activities. Previous research has not indicated any direct impact of timber growth on hunting. However, hunting has been restricted by its effects on wildlife which depends on the forest for food. The relationship between the amount of hunting use and wildlife impacts is not well understood. Very few studies have systematically examined the effects of varying numbers of visitors on wildlife. Even fewer wildlife studies have determined an accurate population count of wild animals prior to the introduction of recreation. Thus, it has been difficult to document a perfect relationship between number of visitors and wildlife impacts. While human-wildlife interactions are too complex to classify, an attempt to generalize some kinds of relations may be useful. Hammitt and Cole (1987, pp84) stated as the following:

> "Recreational activities directly associated with the harvesting of animals can lead to three major changes in the size of wildlife populations that, in turn, affect the quality of these recreational activities. These changes are (1) near elimination of a game species on a local level, (2) reduction beyond a viable breeding population, and (3) reduction beyond a viable hunting or fishing population."

This statement implied that recreation use, such as hunting, should be regulated and managed to avoid all three of these situations. From such a consideration, it may be assumed that the visitors for sport hunting must be proportional to the wildlife population. The equation form was assumed to be

$$V_h = \frac{(N - b)}{a} \qquad (5.65)$$

where

V_h = visitors for sport hunting,

N = wildlife population,

a = successful harvest rate, and

b = minimum reserved population.

The parameter a and b would vary from one species to another according to the increasing rate of population growth, the minimum population protected and some environmental factors. The 1986 statistical data of the successful harvest rate for the study area was provided by the Alberta Forest Service as shown in Table 5.5. Based on this Table, the average successful harvest rate is about 10%. Thus, the coefficient a was assumed to be equal to 0.1 in both the moose and the caribou models. The coefficient b was assumed to be 0.064 animal per quarter section (10 animals/100 km^2) for caribou and 0.384 animal per quarter section (60 animals/100 km^2) respectively, n are the AFS wildlife resource objectives.

Table 5.5 THE 1986 STATISTICAL DATA OF SUCCESSFUL HARVEST RATE PER VISITOR-DAY FOR HUNTING

Game species	Successful harvest rate
Moose	7%
Mule deer	9%
White tail deer	12%

5.2.2 Grazing

Grazing is another activity involved in producing an integrated resource plan. Grazing by domestic livestock is an important land use in the Eastern Slopes of Alberta. But, because of the old growth forest and difficulty of access in the study area, there is currently a limited supply of available land to accommodate local demand for grazing. In general, logging operations tend to increase grazing capacity significantly for a few years.

In managing forest areas, it is important for the production of timber and forage that the stocking level for livestock be at a level to minimize damage to tree reproduction. The number of livestock permitted to graze should depend upon the kind and abundance of forage available--that is the carrying capacity. Carrying capacity is usually expressed in terms of animal unit months (AUMs). An animal unit is the equivalent of 454 kg of animal live weight, or a cow and a calf (Avery, 1975). The carrying capacity of a forest range depends largely on the number of openings in which grasses can become established.

Timber management activities and changing land use affect the quality and quantity of land available for forage production (Grelen, 1978). To assess the impact of timber management and land area changes on the future supply of forage, forage production relationships on forestland must be quantified and linked to the timber growth model.

Management of livestock under forests has traditionally been extensive, rather than intensive (USDA, Forest Service, 1980), and models used to estimate forage production under the changing overstory canopy have been less detailed than the pasture models (Wolter et. al., 1982). Models for both pasture and forest have focussed on specific sites for application, and were typically regression models based on long-term data collected at one site (Joyce and Baker, 1987).

According to Joyce (1986), a theoretical curve of forage production as a function of timber stand ages on a specific site can be drawn (Figure 5.15). As the forest stand age (the x-axis could be age, timber volume, basal area, or canopy closure), a dramatic drop in forage production is observed. Factors such as forest type, site class, stocking level, and timber management shift the slope and/or the intercept of this theoretical curve.

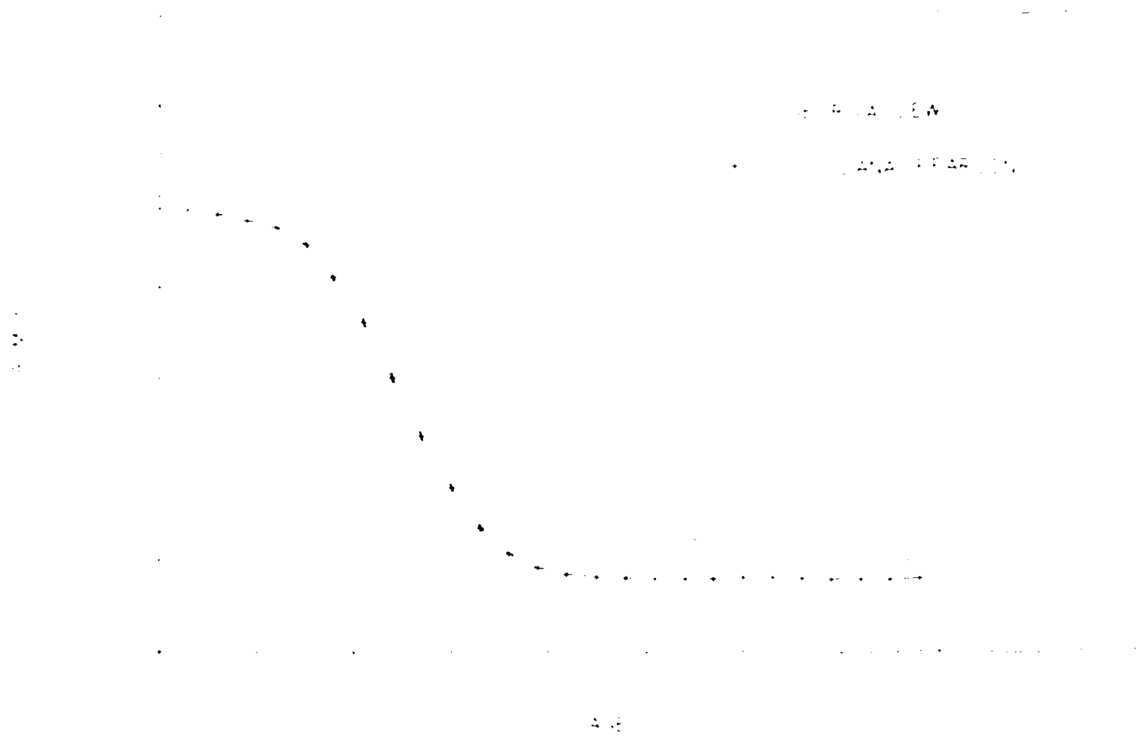


Figure 5.15 Theoretical forage production under forested stands (adapted from Joyce, 1987)

Mathematically, this graph is of the following form:

$$Y = ae^{b(X-c)^2} + d \tag{5.66}$$

where

$a, b, c,$ *and* d are constants

e = base of natural logarithms.

Assuming that the study area has forage production similar to that shown on Figure 5.15 and if Y represents the annual number of AUMs instead of forage production, and X is the basal area (BA) of a forest stand, then the mathematical model for grazing is

$$AUM = ae^{b(BA-c)} + d \qquad (5.6)$$

where

a = 4.94,

b = 0.1,

c = 1.5, and

d = 0.99.

That is

$$AUM = 4.94e^{0.1(BA-1.5)} + 0.99 \qquad (5.7)$$

and it can be graphically shown as in Figure 5.16

Figure 5.16 The grazing capacity of forested stands in the study area

5.2.3 Wildlife

A third resource to consider in integrated planning is wildlife, or more specifically the identification of wildlife population status. Wildlife is a product of the land. Forests provide a home, or habitat, for many kinds of wildlife. Hundreds of kinds of plants make their home under the forest canopy and could not exist without it. The important elements of wildlife habitat are food and cover, and the combination and balance of these factors determines the kinds of wildlife to be found in any forest area (Stoddard and Stoddard, 1987). The management of the wildlife resource implies an effort to attain a degree of balance between the food and cover available and the animal populations that are favored.

Some of the well-known game animals, found mainly in the study area, are moose, elk, bear, and caribou. The shade of dense old-growth forests is often [illegible] great to provide the shrubs and herbs needed for food by animals, but a well-managed forest with openings can produce abundant supplies. Wildlife tends to prefer the edges, between the woods and the openings, because food lies on one side and escape cover on the other.

The evaluation of wildlife resources is more difficult (and often less precise) than similar assessments of rangelands because many game species obtain a large proportion of their forage from browse species (woody plants), fruits, and nuts. Also, many wild animals are more mobile than domesticated livestock, and their movements are ordinarily beyond human control. There is no simple and reliable technique for determining population size, and only rough estimates can be made of the number of animals being supported within a given area (Avery, 1975).

The widely used differential equation representing sigmoid population growth is (Grant, 1986)

$$\frac{dN}{dt} = rN - \left(\frac{r}{K}\right)N^2 \qquad (5.69)$$

where

N = wildlife population per ha,

dN/dt = instantaneous rate of change in size of population N,

r = intrinsic rate of increase of population N (a constant), and

K = carrying capacity of the environment for population N.

The carrying capacity of a forested area for wildlife depends largely upon the size of trees and stand density, therefore it can be heavily influenced by harvesting operations. For example, lichen, the main food for caribou, grows mostly in stands over a hundred years old, and if the rotation age is 90 years, then there will be little food left for caribou after one rotation period. In this case, the carrying capacity will decrease dramatically.

Avery (1975) indicated that carrying capacity for various habitats could be computed by the same procedure used for determining [illegible] grazing capacity. Thus, a simple mathematical model of carrying capacity for wildlife such as moose, elk, and bear can be expressed

$$K = a_u e^{b_u(BA - c_u)} + d_u \qquad (5.70)$$

where

BA = basal area per unit area, and

a_u, b_u, c_u, and d_u are constants.

If assuming that the forested lands provide game species with the same amount of forage as that with livestock, and that the dry-weight forage required to provide for one unit month on forested lands is approximately 410 kg, the model parameters can be obtained as follows:

a_u = 5.47,

b_u = 0.099,

c_u = 1.5, and

d_u = 1.09.

Different from other game species, the carrying capacity for caribou could change as the forest stand age increases following the curve shown in Figure 5.17. Again, for the mathematically oriented, the graph (Figure 5.17) can be presented by the following equation:

$$K = a_c\left(\frac{Age^{b}}{c^{b} + Age^{b}}\right) + d_c \qquad (5.71)$$

Figure 5.17 Theoretical carrying capacity change for caribou under forested stands

where

[illegible] = forest stand age

a_c, b_c, [illegible] and d_c are constants,

and

a_c = 2.92,

b_c = 8

c_c = 100,

d_c = 0.72.

Hanson (1975) investigated the conflicts between wildlife and domestic grazing on forest lands in Alberta. He indicated that "On the grazing land examined, there was relatively little serious damage to game range. The total land grazed by livestock is not great and the use on most areas was light enough that the browse remained almost unused and other forage was not damaged." Based on this work, one may consider that there are no competitive relations between grazing and wildlife management.

5.2.4 Watershed

Watershed is another resource output of land use. It is one of most dynamic natural resources, and at times it may completely dominate or limit the use and management of other resources on the land. In both liquid and solid forms, water influences the production rates of timber, forage and wildlife. In the form of snow, water provides winter recreational opportunities, and liquid water on and from wild lands affects fishing, swimming, boating, and other water based recreational activities (Avery, 1975).

The E8 management unit of the Berland sub regional planning area includes three major drainages. The Little Smoky River, the Simonette River, and Bolton Creek flow north into the Smoky River. As an integrated resource planning area, vegetation manipulation to increase water yield is one land use alternative to be assessed. Water yield or run-off, is affected by hydrometeorological factors such as precipitation (quantity and type), solar radiation, and wind speed. Yield is also affected by evapotranspirative losses from existing vegetation. The hydrologic cycle (Figure 5.18) can be expressed by a simple water balance equation:

$$Q = P - E_t - S \qquad ([illegible])$$

where

Q - water yield or run-off,

P - precipitation,

E_t - evapotranspiration, and

S - change in storage.

Since with the change in soil moisture storage assumed to approach zero over long-term, run-off is the difference between precipitation and evapotranspiration as in the following:

$$Q = P - E_t \qquad ([illegible])$$

Vegetation manipulation done by harvesting can therefore increase yield by reducing evapotranspirative losses. Silvicultural activities cannot increase the amount of precipitation entering the system but it can influence the disposition of rain or snow in both time and space in a small local

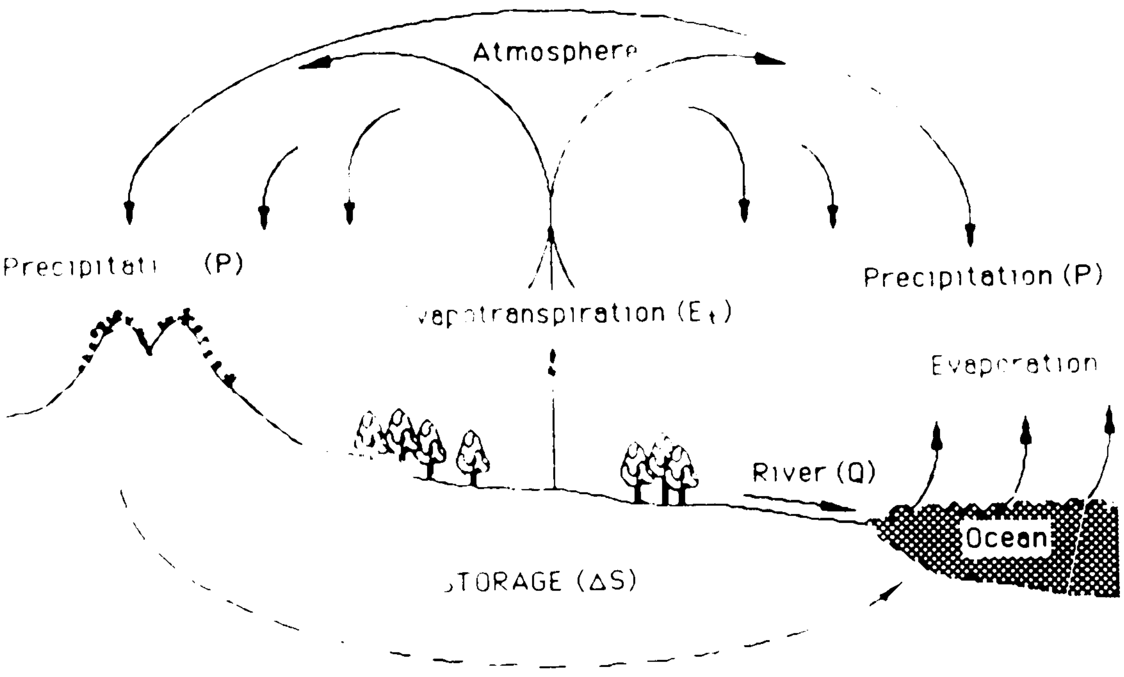


Figure 5.18 The general water cycle (adapted from Hewlett, 1982)

scale. Overall, the removal of vegetation may increase the net precipitation received by reducing the amount of interception storage (Troendle and Leaf, 1980).

Numerous studies throughout North America have been conducted to determine the effects of forest cover removal on water yield. A number of factors affecting changes in water yield have been identified and considered by Troendle and Leaf (1980) in the WRENSS (Water Resources Evaluation of Non-Point Silvicultural Sources) procedure. These factors are the following:

a. Vegetative cover, expressed as cover density, is an index of the capability of the stand to utilize incoming radiant energy to transpire water. It is also significant in defining the energy transmitted to the ground. Both vary according to crown closure, vertical foliage, distribution and species. The removal of forest cover may change the aerodynamic flow of air over the forest opening, leading to a change in pattern of snow accumulation. Studies have shown (Golding and Swanson, 1978) significant increases in snow accumulation near the center of the forest opening are largely offset by decreases in snowpack in the adjacent forest so that the total snow storage on the watershed subjected to cutting is not changed. However, when openings are greater than 14H (H = height of surrounding trees) in diameter, the total snow may decrease through sublimation losses and transport of snow out of the basin by wind scour. This represents a net loss of precipitation, thus a net decrease of water available for streamflow.

b. Water equivalent of the snowpack provides useful information on the amount of liquid water stored in a winter snowpack. Any large retention of snow as a result of forest cutting can be important in determining the amount of spring runoff. It is expected that the change in snow accumulation patterns produced by timber harvesting will persist until the regeneration of new trees approach the height of the surrounding forest.

c. Evapotranspiration, referring to the moisture loss through evaporation and transpiration, provides the greatest mechanism for potential changes in the quantity of water available for streamflow as compared to the other water balance components.

d. The aspect of sites is of important significance in terms of the energy available to melt snow and transpire water.

e. The condition of the analysis by WRENSS (Water Resources Evaluation for Non-Point Silvicultural Sources) refers to the hydrologic state of the watershed. Two conditions are available: baseline and proposed. The baseline condition assumes complete hydrologic utilization is achieved. This is usually thought as a fully forested watershed (primarily coniferous trees) capable of maximum evapotranspiration. The proposed condition is the hydrologic state following a proposed silviculture activity.

Most hydrologic models (Leaf and Brink, 1973; Troendle and Leaf, 1980) have dealt with time increments of an hourly or daily [illegible]. In such cases, processes of the hydrologic cycle such as soil evaporation, infiltration rates, transpiration, interception, and throughfall become significant in developing individual functional relationships.

This watershed model will use a yearly time scale for model parameters and variables. In this time span, prediction of many of these individual hydrologic processes is neither possible nor of major importance. The watershed model therefore concentrates on developing only those processes which are pertinent to predicting long-term timber-water yield relationships. Due to the lack of data, the precipitation, evapotranspiration, and water yield data generated by Alberta Forest Service using WRENSS (AFS, 1988b) were used to determine the model parameters.

Based on the first principle, the evapotranspiration can be derived as a function of the timber stand characteristics, such as basal area and height. Simply, this relationship can be assumed to be linear,

$$E_t = a_1 + a_2 BA + a_3 H \qquad (5.74)$$

where

BA = basal area,

H = height, and

a_1, a_2, and a_3 are constants.

And similarly, precipitation in the forest can be expressed as a linear function of real precipitation and average tree height:

$$P_f = b_1 P + b_2 H \qquad (5.75)$$

where

P = real precipitation,

P_f = precipitation in forest, and

b_1 and b_2 are constants.

Thus, the water yield model can [illegible] as:

$$Q = P_f - E_t \qquad (5.76)$$

Tables 5.6, 5.7 and 5.8 present the WRENSS output for the Little Smoky, Simonette and Bolton basin. Based on these data, the precipitation and evapotranspiration models of the three basins were developed respectively.

For Little Smoky basin:

$$P_t = 0.843P + 4.565H \quad (H < 0.071P) \tag{5.77}$$

$$P_t = 1.167P \quad (H \geq 0.071P) \tag{5.78}$$

$$E_t = 284 + 1.58A + 4.84H \tag{5.79}$$

For Simonette basin

$$P_t = 0.843P + 4.664H \quad (H < 0.069P) \tag{5.80}$$

$$P_t = 1.167P \quad (H \geq 0.069 \tag{5.81}$$

$$E_t = 245 + 1.58A + 5.57H \tag{5.82}$$

For Bolton basin:

$$P_t = 0.843P + 4.318H \quad (H < 0.075P) \tag{5.83}$$

$$P_t = 1.167P \quad (H \geq 0.075) \tag{5.84}$$

$$E_t = 245 + 1.58A + 5.56H \tag{5.85}$$

Table 5.6 Water balance for Little Smoky basin per WRENSS model

Season	Precipitation(mm)		Evapotranspiration(mm)		Flow(mm)	
	Forest	Open	Forest	Open	Forest	Open
1	131	96	46	27	85	69
2	205	144	165	133	40	11
3	199	199	226	124	-27	75
Annual	535	439	437	284	98	155

Table 5.7 Water balance for Simonette basin per WRENSS model

Season	Precipitation(mm)		Evapotranspiration(mm)		Flow(mm)	
	Forest	Open	Forest	Open	Forest	Open
1	131	96	36	21	96	75
2	207	144	149	124	58	20
3	199	199	226	100	-27	99
Annual	537	439	411	245	127	194

Table 5 8 Water balance for Bolton basin per WRENSS model

Season	Precipitation(mm)		Evapotranspiration(mm)		Flow(mm)	
	Forest	Open	Forest	Open	Forest	Open
1	130	96	36	21	94	75
2	201	144	147	124	53	20
3	199	199	226	100	-27	99
Annual	530	439	409	245	120	194

6 IMPLEMENTATION OF THE SIMULATION MODEL

Although mathematical models have been formulated, mathematics alone cannot make them work. The reason is that only very simple mathematical models have exact analytical solutions. Therefore, an appropriate computer program must be developed to do the simulation.

6.1 Selection of Computer Programming Language

Many computer simulation models in resource management and planning have been developed in the last two decades. Most of them were written using a general purpose programming language such as FORTRAN. Recently, considerable emphasis has been placed on the introduction of "user-friendly" software into the workplace. The reason for this is well stated by Weisz (1988)

> "Our region is operating in a multiple level, multiple purpose multi-resource, multiple use, multi-year environment that has both vertical and horizontal hierarchies and processes. We need a fast, cheap, simple, efficient, and effective means of acquiring, analyzing, interpreting, displaying, and transmitting information to planners and decision makers at each level of the organization. We need uniform and consistent information which can be used in all related processes--data which is credible and comprehensible to both ourselves and to our interested publics. The software utilized for processing this information should be compatible with existing office automation software utilized".

Designed for professionals with little computer training, user friendly software has enhanced the manager's ability to implement highly technical projects at the work station (Gray, 1986).

A spreadsheet program like LOTUS 1-2-3 is one example of a commercially available software package which can be used to address a variety of resource planning issues. Gary and Keith (1988) used a spreadsheet program to model the habitat-species evaluation procedures in the Northwest Territories, Canada. Weisz (1988) described the use of spreadsheets in multiple use planning instead of mathematical programming.

At the beginning of this project, a spreadsheet software, LOTUS 1-2-3 Release 2.01, was used to develop the simulation model. However, the major disadvantage of spreadsheet packages of insufficient computer memory soon become a limitation on the model. If the whole study area were to be represented on a LOTUS 1-2-3 worksheet 195,420 cells would be required but the available computer memory (640K) can only hold up to 77,056 cells. One method to overcome this memory limitation is to provide additional memory to the computer; a second is to split the data into two or more files. However, database management packages have the capability to manipulate large databases on disk files so the research was direct toward developing the simulation model using a database management program.

The database management package called dBASE IV is a very powerful and user-friendlv program that helps users collect and manage information or data. The data are stored in a database on disk storage. dBASE IV is not only a complete database management system but it also includes a programming language called dBASE (dBASE IV manual, 1988). With this programming language, the user can create customized applications for specific needs, allowing for control of the database management tasks. The greatest advantage of dBASE IV over LOTUS 1-2-3 is that it has no memory problems because it does not load all the database into the computer memory at once. Another advantage of dBASE IV is that it can load and run assembly language programs without leaving the program. This feature can increase the speed of executing simulation runs.

6.2 Establishment of Simulation Database

6.2.1 The Data

The study area consisted of 3257 quarter sections of old-growth trees, most of which have never been harvested. A section, on basis of survey, is an area of approximately one square mile (250 ha.). Expressed in metric system, one quarter section is approximately sixty-four hectares. In 1981, regular harvesting in the study area started by British Columbia Forest Products (BCFP). The reforestation began two years later.

The timber data used in this study were provided by the Alberta Forest Service (AFS) from their Phase 3 inventory database. Since the height data provided by AFS are classified by height classes, the average values of each class were selected as average tree height. The diameter data were only available for some cover types. Therefore estimates were made for the rest. The tree form factors were estimated based on the Alberta Phase 3 Forest Inventory: Yield Tables for Unmanaged Stands (1985). The site indexes also came from these tables. In addition, the number of trees per hectare for each quarter section were adjusted using the diameter, height, and volume data and the equations developed in this study since some data were unrealistic when compared to the volume.

As for other non-timber resources, there were no field data available for the study. In order to make sample simulation runs, the data were estimated based on available statistical information and existing research results. The esthetics mean rating scale was developed by Benson and Ullrich (1981). It is a numerical rating between 0 (dislike) and 9 (like) selected by different panels of viewers based on the visual quality of lodgepole pine and Douglas-fir/larch forests. The rating decreases to the minimum when the forest has been harvested (clearcut) and increases since then until it is mature. After that, the rating will decrease slowly.

The grazing carrying capacity data were from Joyce's (1986) theoretical forage production curves under pine stands. The curves illustrate that the herbage biomass production varies from 2692.3 *kg*/ha. (2400 *lb*/acre) to 471.1 *kg*/ha. (420 *lb*/acre) as stand age increases. Because grazing carrying capacity is usually expressed in terms of animal unit months (AUMs), the above values can be converted by assuming one animal unit month to be equivalent to 475 *kg* (1046 *lb*) so the grazing capacity varies from 5.67 AUMs/ha to 0.99 AUMs/ha.

The moose and caribou populations were estimated based on the Alberta Forest Service wildlife resource statistics. The objective of AFS is to increase caribou densities to 10 animals/100 km^2 (0.064/64 ha.) and to increase the moose density to 60 animals/100 km^2 (0.384/64 ha.). The current moose and caribou population were estimated to be 50 percent of the desired population, that is 5 animals/100 km^2 (0.032/64 ha.) for caribou and 30 animals/100 km^2 (0.192/64 ha.) for moose. Recreational hunting use depends on the wildlife population. For simplicity, the assumption was made that no hunting activities occur presently in the study area.

The watershed for the study area consists of three basins; the Simonette River basin, the Little Smoky River basin, and the Bolton Creek basin. The water yield data for these basins were summarized from the predicted results of the Alberta Forest Service using WRENSS. Only the two end values, water yields of forested (old forest) areas and water yields of open areas were used Any values between them were computed by the equations developed in this study which are functions of the tree height and average basal area. Precipitation values used were the recorded annual total for precipitation.

6.2.2 The structure of simulation database

The database file constructed for the simulation includes 90 fields and 3257 records. The order and contents of these variables are listed in the Appendix B.5. Fields 1 to 11 contain the site and location information. Fields 12 to 71 contain the timber data for the four main species, in turn, white spruce, black spruce, pine, and aspen. Fields 72 through 77 are recreation activities. Fields 78 and 79 contain domestic grazing data. Fields 80 to 83 contain wildlife variables. The last four fields (84 to 87) contain watershed information.

Area control method was used to decide the annual allowable cuts and a two pass harvesting system was employed to identify the percentage of land area to be harvested. For each planning unit (quarter section), 50 percent of the area can be harvested in the first pass, and the other 50 percent would be left for the second pass. To simulate these operations, in setting up database, each unit was divided into two subunits to represent the areas for different passes.

6.3 Building the Simulation Model with dBASE IV

The computer program (IRPM), which is listed in Appendix B.2, consists of a main menu, five subprograms and a number of subroutines written in dBASE IV programming language. The program can be executed on a microcomputer with 640K memory and a hard disk drive. The inputs of the management strategies and the model control parameters (i.e., the time interval and the total length of the simulation time) can be entered by the user following the screen menus. A sample of the model input screen is displayed in Appendix B.3.

Three alternatives of simulation available for users are SIMULATA, SIMULATB and SIMULATC. The features and algorithms of these subprograms are described below.

Main program (menu)

The main program (Figure B.1) makes no computations, but serves only to coordinate the flow of the program. It performs the following operations:

a. Modify the database structure to add or delete the management activities.

b. Add to or delete from the database.

c. Call the program SIMULATA to perform the calculations required to simulate operations of harvesting, regenerating, growing, and thinning for different rotation lengths.

d. Call the program SIMULATB to simulate the same operations as the SIMULATA does but only for the particular management units defined by the users.

e. Call the program SIMULATC to simulate the resources changes over time without any management operations.

Program SIMULATA, SIMULATB and SIMULATC

SIMULATA calls seven subroutines to perform computation and other operations in the proper sequence (Figure B.2). Each subunit is processed for a given period (1 year preferred), and results for all subunits are combined into a report which can be printed. Seven routines (HARREG, GROW, THIN, GRAWLD, WATER, HUNTING and BEAUTY) are used to calculate the annual changes in the area being simulated. Annual changes include tree growth, thinnings, and regenerations, carrying capacity changes for grazing and wildlife, water yield changes, hunting visitor-days changes, and natural scenic beauty changes. The number and location of cutting blocks are determined by the program based on the annual allowable cuts and following a sequential order from quarter section 1 to 3257.

SIMULATB performs the similar operations as SIMULATA, except it allows the user to determine the number and location of areas to be harvested and regenerated. As such, SIMULATB provides the user with the choice of testing the management strategies as what exactly happens in the real world. The disadvantage of this approach is that the input can be tedious and time consuming.

SIMULATC , different from SIMULATA and SIMULATB, is designed to simulate the changes of resources over time with no management operations and without any uses. This alternative can be used to test the behavior of dynamic growth equations of individual resource with different mathematical structures and coefficients.

Subroutine GROWTH

GROWTH (Figure B.3) is called by program SIMULATA, SIMULATB, or SIMULATC to compute diameter, height, basal area, number of trees, and volume. Subroutine GROWTH contain growth equations for white spruce, black spruce, pine, and aspen on three different sites. Based on the user's input, GROWTH can also modify growth equations to compute the post-thinning increase in average tree diameter.

Subroutine HARREG

HARREG (Figure B.4) is called by program SIMULATA to simulate clearcutting and planting/seeding. HARREG consists of six subroutines Harreggl, Harregg2, Harregml, Harregm2, Harregfl, and Harregf2. They perform the same operations but for different sites and cutting passes. The following operations are performed:

a. Compute volume removed. (Volume is expressed in cubic meters.)

b. Simulate clearcutting and regenerating (planting or seeding) operations. Average DBH, site height, basal area, and volume are reduced to zero. The unit is assigned a new number of trees per hectare (equal to the planting density selected by the model user). A new seed/seedling stand is established after each clearcut. Stand age is set to a new value, depending on the delay period of regeneration. This value varies from 0 to -7.

Subroutine THINNING

THINNING is called by program SIMULATA to simulate thinning operations. Thinning will be made at the age specified by the model user, and options include not thinning during the rotation. The number of trees and the basal area to be reserved after thinning are determined by the user. The planning area will not be thinned if its density is already at or below the appropriate residual.

Subroutine GRAWLD

For each time interval, GRAWLD is called by the program SIMULATA, SIMULATB, or SIMULATC to compute potential grazing carrying capacity, and moose and caribou population of each planning unit. The planning subunit is first examined to see if any trees are present in the area. If not, the unit is considered as a clearcut block and having the maximum carrying capacity for grazing and moose, and the minimum carrying capacity for caribou.

Subroutine ESTHETIC

This subroutine, called by the program SIMULATA, SIMULATB, or SIMULATC, computes the impact of tree growth and harvest on the natural scenic beauty. The mean rating of esthetics is considered only as a function of years after harvest.

Subroutine HUNTING

HUNTING is called by program SIMULATA, SIMULATB, or SIMULATC to compute potential visitor-days of hunting use available from each subunit. Potentials are based on the wildlife population and on the successful harvesting rate which was assumed as 10 percent of total visitor-days. The present subroutine estimates only the recreational use of moose hunting and caribou hunting. It can be expanded to include other hunting activities. In HUNTING, the number of animals harvested per year is deducted automatically from the total population.

Subroutine WATERYLD

The subroutine WATERYLD is called by program SIMULATA, SIMULATB, or SIMULATC to compute the effects of timber growth and harvest on the watershed. It calculate the annual water yields separately for each of the three basins. The potential precipitation for forested and open areas is estimated from the annual average precipitation.

Subroutine REPORT

REPORT is called at the end of each year simulated to print the simulated results of each year. Array values computed and stored in the dBASE file IRPRPT.DBF are printed using the dBASE report form ReportA and ReportB (Appendix B.4), and transferred to LOTUS 1-2-3 file IRPRPT.WKS to draw graphs.

Simulation execution steps

The flowchart of the simulation model can be found in Appendix B.1 and the execution procedure of SIMULATA can be detailed as follows:

a. Input the rotation length, the regeneration time, the species to be regenerated, the method of regeneration, simulation control parameters, and thinning operation parameters if applicable.

b. Calculate the annual allowable cuts (number of areas) according to the area control method.

c. Calculate the annual timber harvests.

d. Harvest trees from subunit 1 of each area to be cut if the simulation time is less or equal to the half of rotation length and from subunit 2 if the simulation time is greater than the half of rotation length and less or equal to rotation length. Until simulation time reaches the half of rotation length or the total rotation length, all areas left will be harvested even though there may be more volume harvested than specified by the annual allowable cut.

e. Regenerate all areas harvested. For different quality sites, the different initial stand parameters will be assigned. If the regeneration is not carried out in the same year as harvesting, the tree age will be given a negative values based on the regeneration time entered by the user.

f. Calculate the timber growth for the whole areas. Dependent on the site quality and age of stands, diameter, height, number of trees, basal area and volume are calculated by employing the different equations.

g. Calculate the esthetics changes, hunting visitor-day changes, water yield changes, grazing carrying capacity changes and wildlife population changes for all subunits.

h. If a thinning operation is to be used, compare stand ages of the harvested areas to decide whether or not a thinning operation should be performed.

i. Summarize the annual timber harvests, esthetics changes, hunting visitor-days, grazing carrying capacity changes, wildlife populations and water yield. The results will be written in a temporary database file so that they can be output later.

j. Simulation time is advanced and the above steps are repeated until the simulation time reaches the input value.

k. Results for the whole simulation period are transferred from the temporary database file to Lotus 1-2-3 for making graphs and printing in a tabular form.

SIMULATB and SIMULATC follow the similar procedure. A sample simulation report of SIMULATA is included in Appendix B.4.

Time interval

Unit time interval for the simulation may be varied by model users. The choice is made in the program SIMULATA. All variables will be computed for each interval. Theoretically, the smaller the time interval is, the more accurate the simulation results are. But a smaller time interval also will result in a longer computing time. Therefore, for most simulation models, the problem is choosing the proper integration step-size or time interval. It should be chosen such that it is sufficiently small to insure an accurate solution and not too small to result in excessive run time.

For the forest dynamic growth problem, the most often used time intervals are from one month to ten years. In this project, the same range was expected to be used. However, because of the difficulty of estimating the coefficients for mortality models, the Euler's integration procedure with time interval of one year was used to make predictions. This step determines that those coefficients are dependent on integration time intervals. Figures 6.1 to 6.8 show the influence of time intervals on main stand variables of pine on good site and black spruce on good site.

From Figure 6.1 to Figure 6.4, the simulation results of site heights and average tree diameters for both species are rarely affected by the integration intervals which are less than 1 year. This means that a time interval of 1 year will be small enough to provide accurate solutions for site height and average DBH growth.

Figure 6.5 shows that the simulated number of trees of pine with a small time interval (0.2 year) decreases slightly faster than that with a large time interval (1 year) as trees get older. Thus, the estimates from the simulation model for mortality of trees will be greater as time interval decreases. Correspondingly, the gross volume estimated using the simulation model will decrease after age 90 as shown on Figure 6.7.

In general, simulation interval of 0.2 years is appropriate for most computer runs and for the level of precision of the equations and constants currently in the model, except for some number of trees equations. This result indicates that the number of trees equations are sensitive to the change of time interval. The simulation runs with different time intervals suggested that the simulation model with the current coefficients would provide reasonable results under the simulation time interval of 1 year. A major corcern about the size of the time interval is that a smaller time interval would result in a great amount of run time.

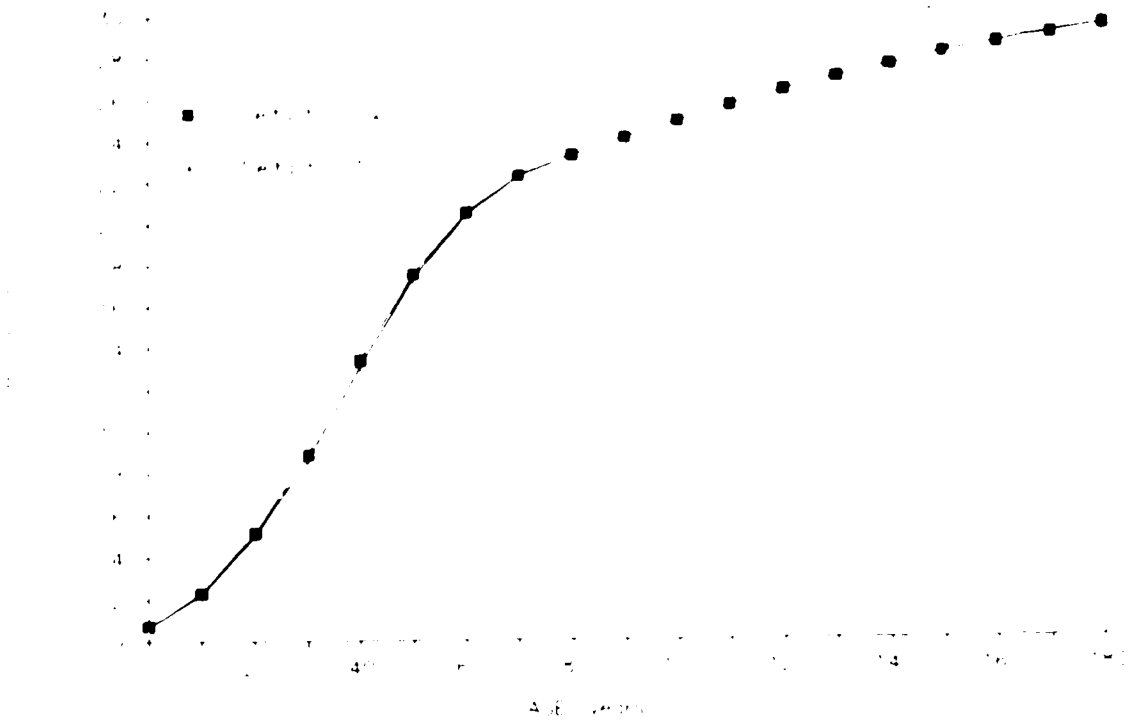


Figure 6.1 Simulated site heights of good site pine with different integration intervals

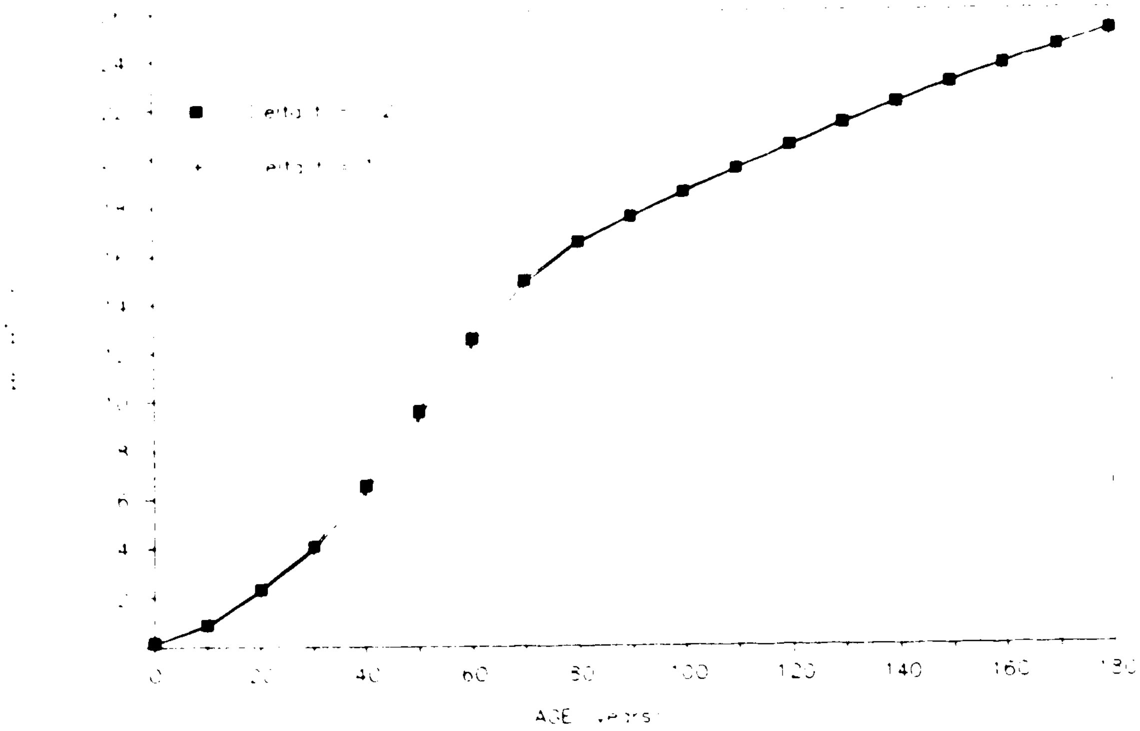


Figure 6.2 Simulated site heights of good site black spruce with different integration intervals

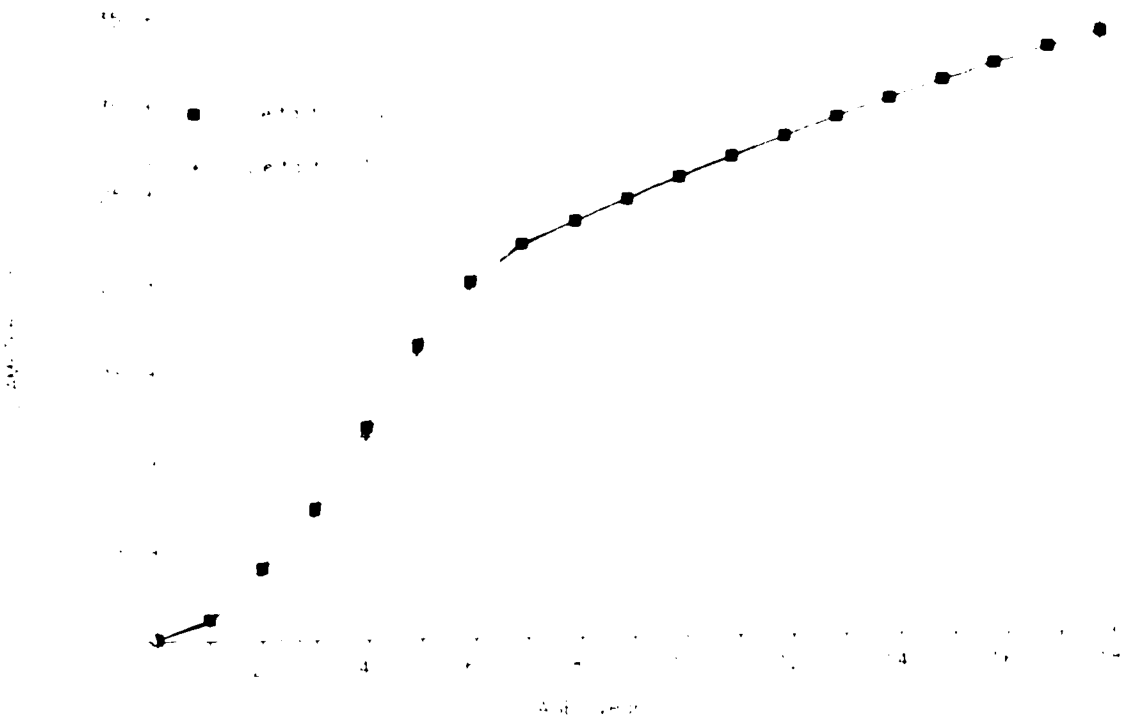


Figure 6.3 Simulated average DBHs of good site pine with different integration intervals

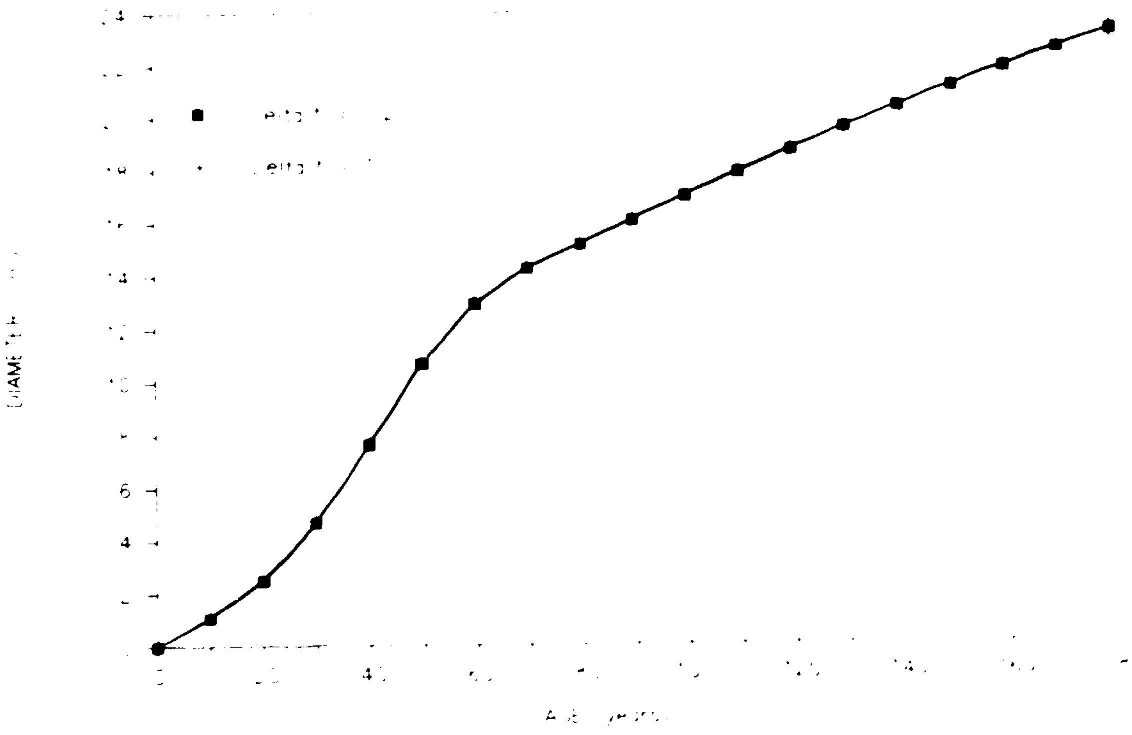


Figure 6.4 Simulated average DBHs of good site black spruce with different integration intervals

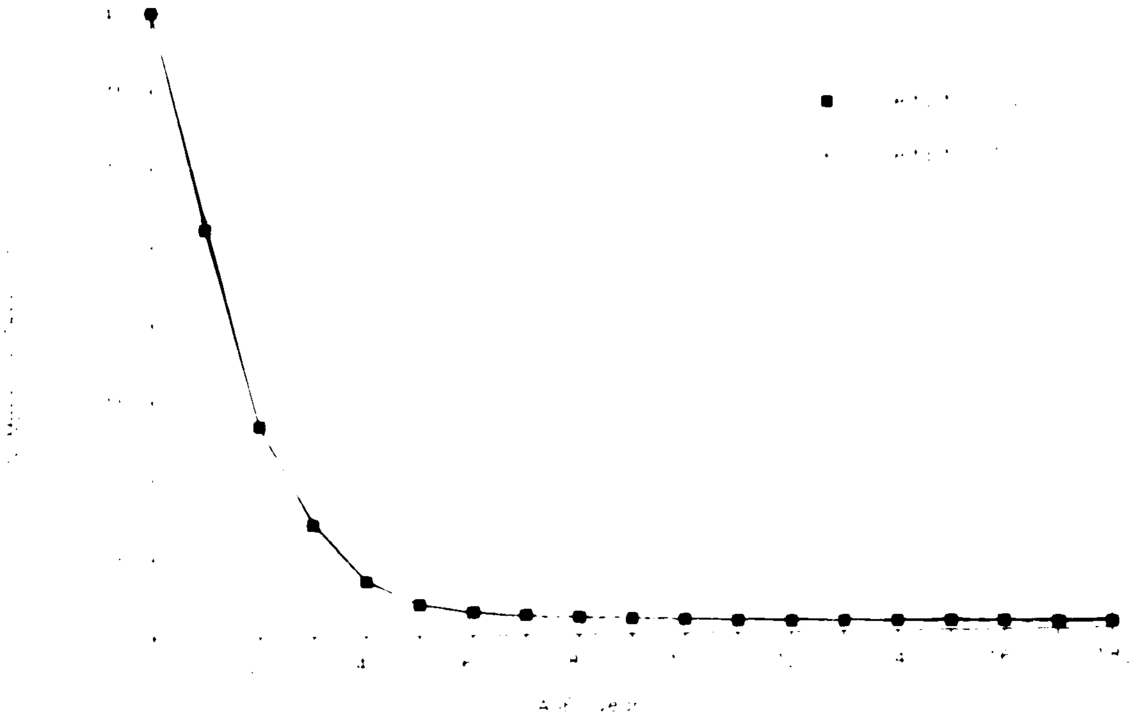


Figure 6.5 Simulated number of trees of good site pine with different integration intervals

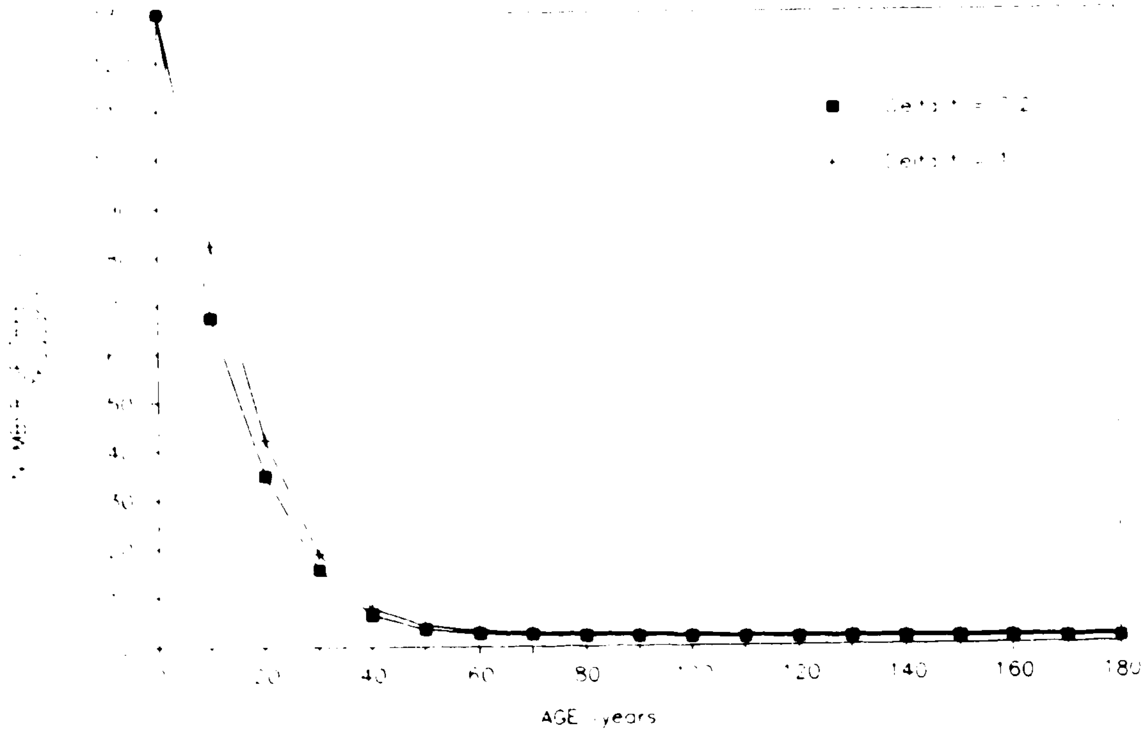


Figure 6.6 Simulated number of trees of good site black spruce with different integration intervals

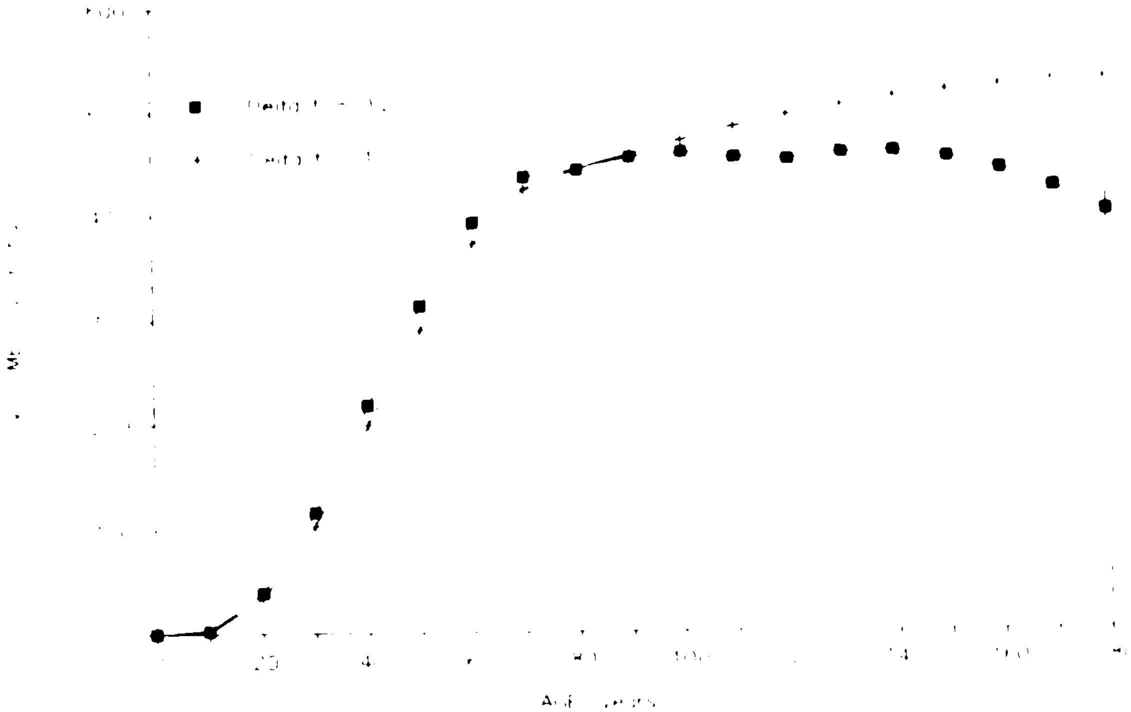


Figure 6.7 Simulated gross volumes of good site pine with different integration intervals

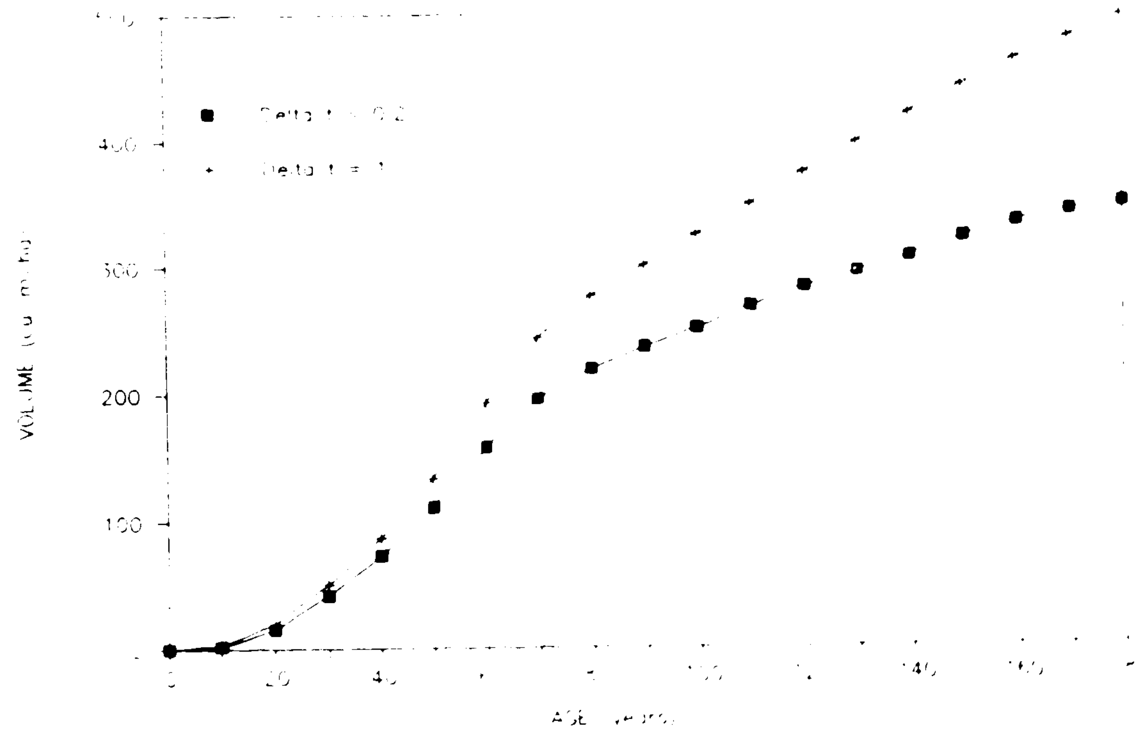


Figure 6.8 Simulated gross volumes of good site black spruce with different integration intervals

7 VALIDATION OF THE SIMULATION MODEL

Validation has been defined by various authors as any process which examines the correspondence between the model and the system under study. In this context, it is difficult to validate projection models in the normal way because the system under study is the future, and the real data is unobtainable within a short time and at a low cost. However, testing a model with historical data or by the first principles increases the accuracy of the model as a decision making tool and identifies some of its weak points.

Model validation was conducted in four separate parts. First, the timber component of the model was validated by comparing estimates for height, diameter, number of trees, basal area, and gross volume with other published data. Second, the estimates in the model for grazing and wildlife carrying capacity was compared with the theoretical curves. The third part was the validation of the esthetic part of the model with Benson and Ullrich's (1981) model. The final part of the validation involved the comparison of the water yield estimates with the estimates produced by the WRENSS model.

7.1 Comparison of the Simulated Tree Growth with the AFS Yield Tables

In order to validate the tree growth models, the AFS Yield Tables were assumed to be accurate indicators of tree growth for the four main species in the study area. Checking the estimates from the model against the AFS Yield Tables provided a simple and reliable method of judging the validity, reliability and accuracy of the model. The AFS Yield Tables with the following utilization standards: minimum DOB (Diameter Outside Bark) at 0.30 *m*, minimum top DIB (Diameter Inside Bark) = 0.0 *cm*, and stump = 0.0 *cm*, were used for each site in order to estimate the model coefficients. The data on simulated pine growth on three different sites (good, medium, and fair) are shown in Figure 7.1 through to Figure 7.15 together with the yield table data. The comparisons of the data from the simulation model and yield table data for white spruce, black spruce, and aspen are included in Appendix C.

7.1.1 Height

Figures 7.1, 7.2, and 7.3 show the comparison of the simulation model data and the AFS Yield Table data for site heights of pine on good, medium, and fair sites for a growth period of 180 years, respectively. The yield table data for site heights and average height growth rate at age 20 (or 30), 90, and 170 were used to determine the coefficients of height growth functions in the simulation model. In cases where yield table data were unavailable for age 10 (or 0), the author estimated values for those years based on an interpolation between age 0 and the first data in the table.

A reasonable agreement of the two curves for each site was seen throughout the growth period. The site heights estimated using the simulation model, however, were observed to be higher than the yield table site heights after age 40. For good sites (Figure 7.1), the height increases quickly after age 40 and the deviation reaches a maximum (2.4 *m*) near age 65. Then, height growth decreases gradually as the data from the simulation model approaches the yield table height data. For the medium site (Figure 7.2), the site height estimated using the simulation model is above the yield table data starting from age 40 and by year 70 the deviation reaches the maximum value 1.65 *m*. Following that, the error begins to decline and two curves join at age 110. As age increases again, the site height estimated using the simulation model is slightly less than the yield table data. For the fair site (Figure 7.3), the comparison of site heights is different from that of good and medium sites. Before age 55, the simulation model produces estimates of site heights that are below the data in the yield tables and beyond that point it starts to produce estimates that are greater than those in the yield tables with a constant deviation of 1 meter.

The deviations of the data from the simulation model from the yield table data may be due to inaccurate growth rates for young forest stands that were assumed in the development of the simulation model. These estimated rates affect the whole growth curve since error is accumulated into the later years. The effect of the initial height included in the model may also contribute to the deviations (the model is capable of simulating the growth by starting from different initial heights). In addition, the few data points for the simulation model taken from the yield tables to determinate the coefficients may be responsible for some error.

7.1.2 Diameter

The comparisons of the estimates on the average diameter of [illegible] from the simulation model with the AFS Yield Table data for three different sites for a growth period of 180 years are presented in Figure 7.4, 7.5 and 7.6, respectively. Good agreement of the two curves representing the data for each of three sites was observed, especially for the young and old forest. However, all three graphs have the same trend, showing some deviation between the two data sets during the middle ages. For the good site, the maximum deviation (about 3 *cm*) occurs between age 60 and age 70. For the

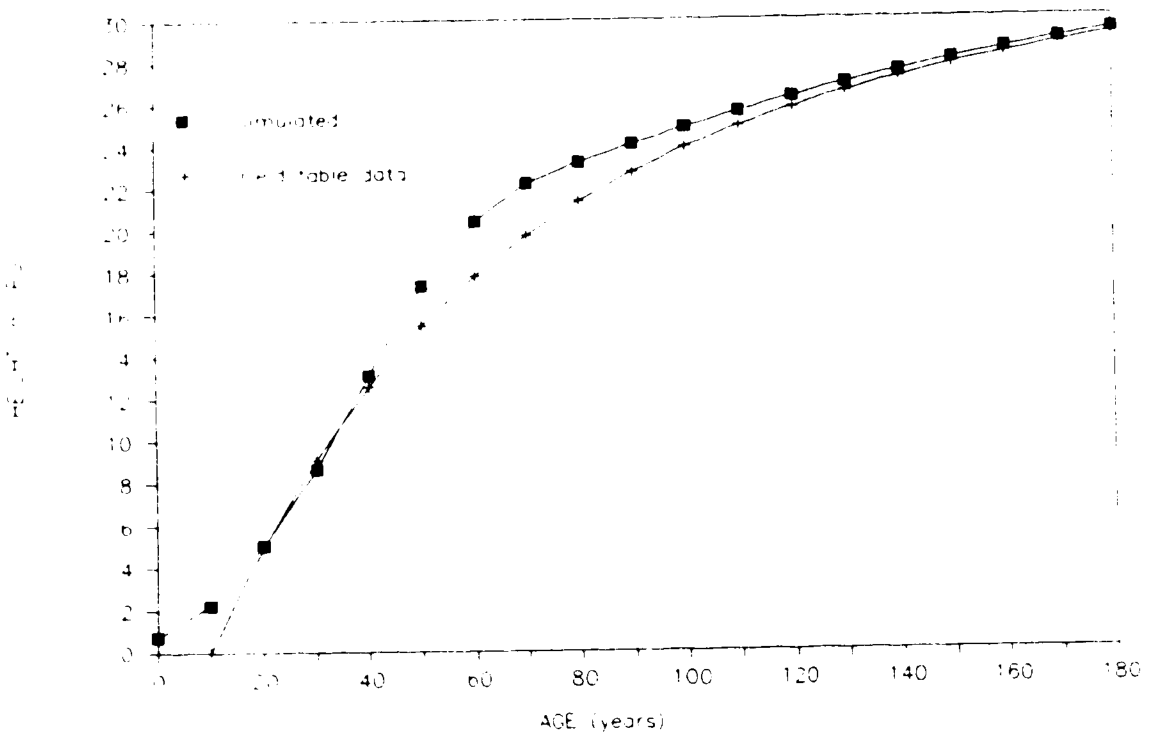


Figure 7.1 Simulated and yield table site height of pine on good sites

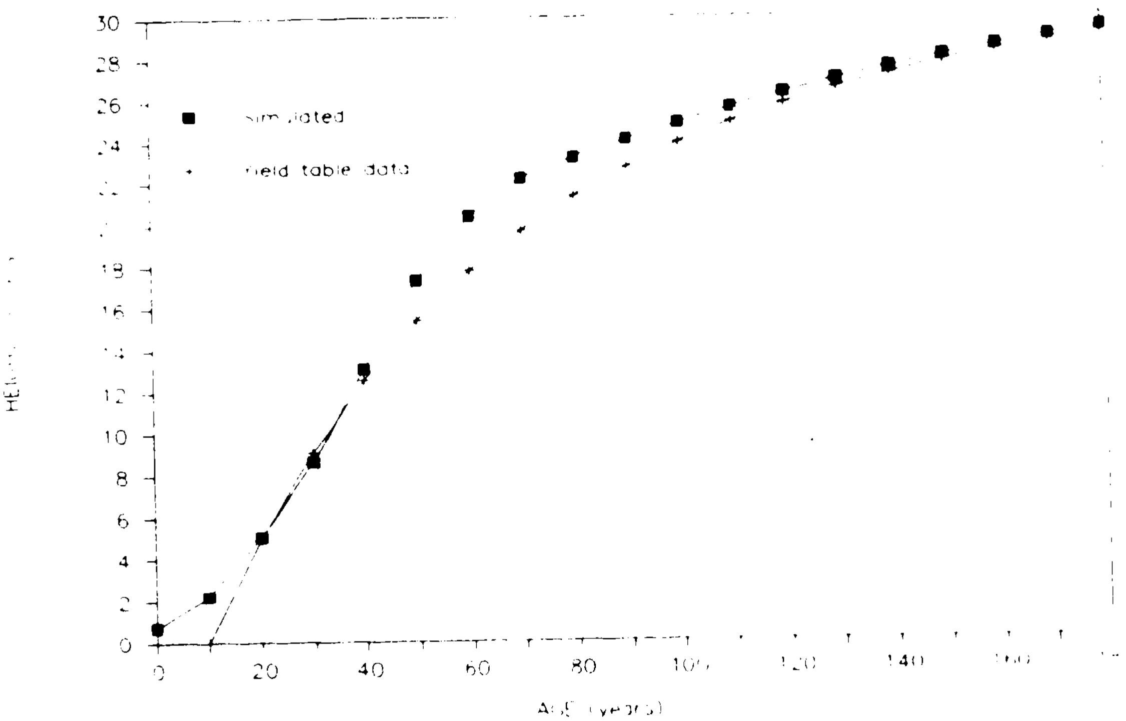


Figure 7.2 Simulated and yield table site height of pine on medium sites

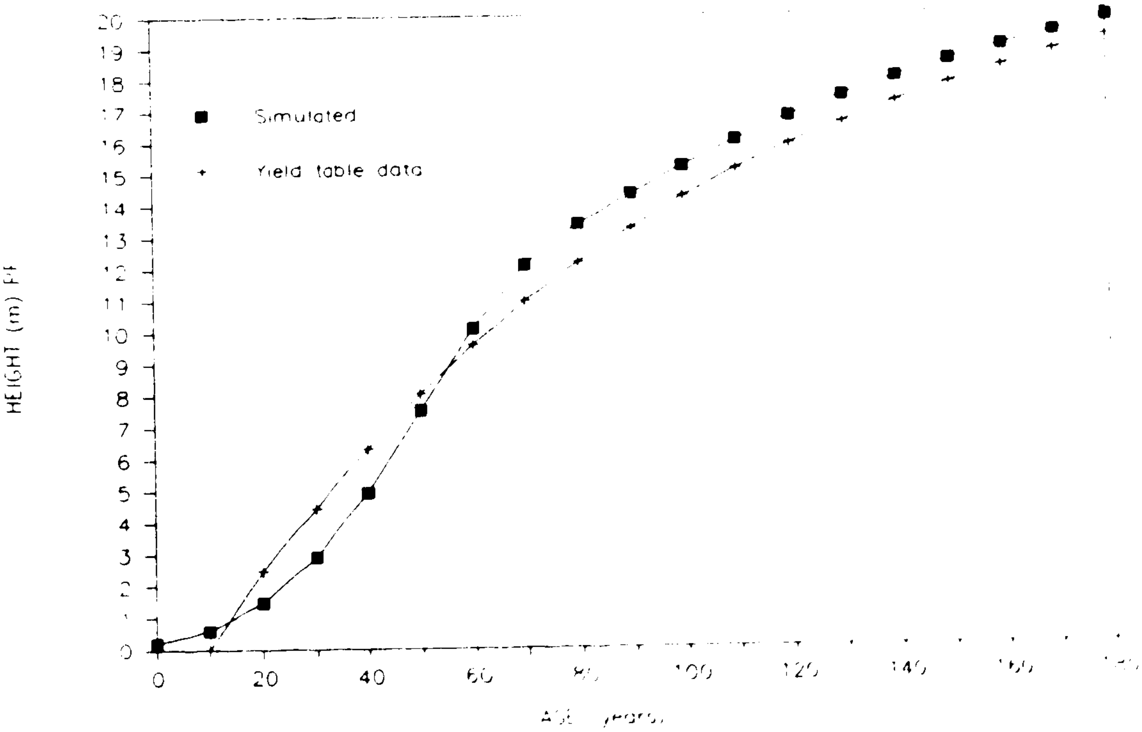


Figure 7.3 Simulated and yield table site height of pine on fair sites

medium site, the deviation gets to the maximum (2.5 *cm*) at age 60. For the fair site, the maximum deviation is 1.9 *cm* which appears at age 65.

The overestimates of diameter when using the simulation model (as shown in Figures 7.4 to 7.6) occur for the same reasons as cited for the differences in the estimates for site height, since both models have the same mathematical structure. The other possible reason is that the consideration has been made to develop sigmoid-shaped growth curves for height and diameter growth and at the same time the effort has been made to keep the most parts of two curves close to each other.

In summary, the diameter growth model used in the simulation model exhibits satisfactory behavior. Depending on the availability of field data, both height and diameter growth models can be redeveloped to mimic the real growth curves. The major difference of this simulation model from widely used regression models is that past diameter (or height) and diameter (or height) growth rate are needed for predicting future diameter growth.

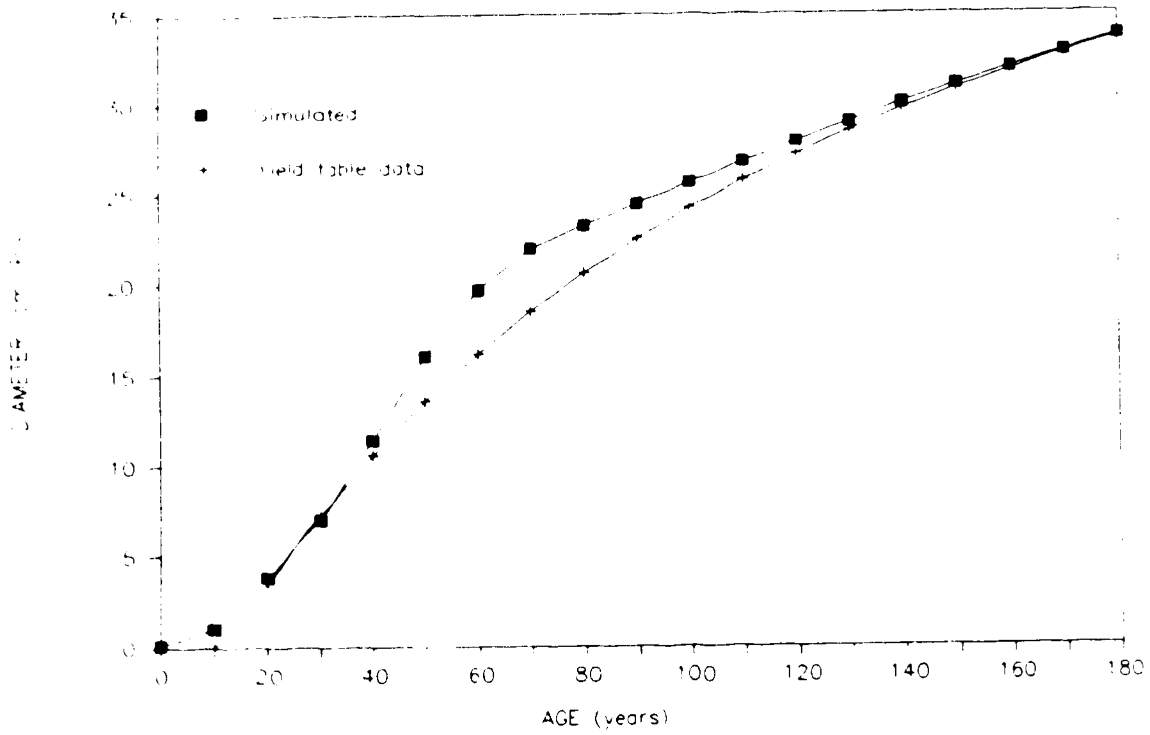


Figure 7.4 Simulated and yield table average DBHs of pine on good sites

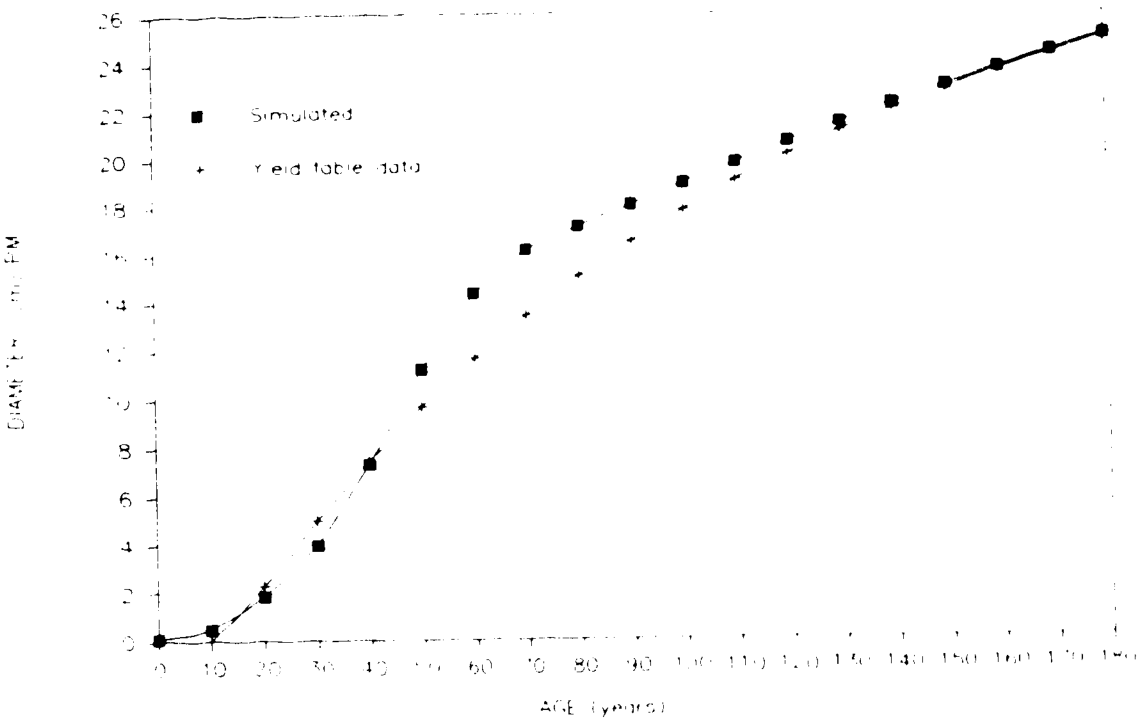


Figure 7.5 Simulated and yield table average DBHs of pine on medium sites

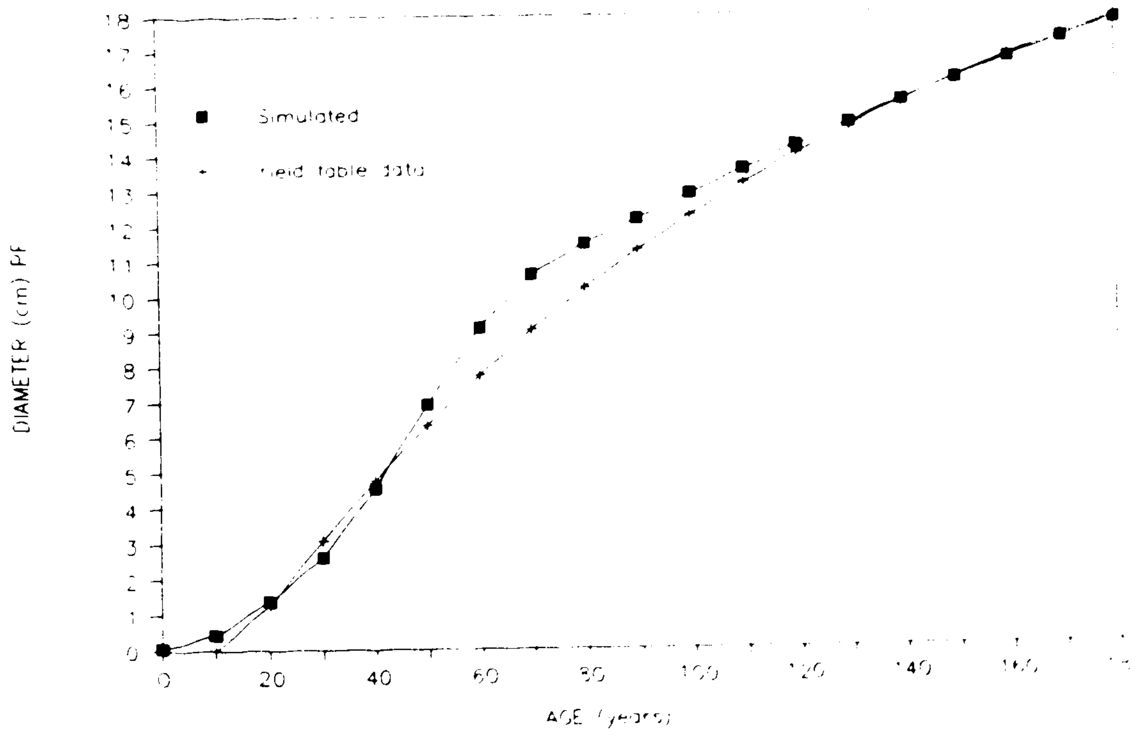


Figure 7.6 Simulated and yield table average DBHs of pine on fair sites

7.1.3 Number of trees

Figures 7.7, 7.8 and 7.9 show the data for the number of pine stems per unit area (tree density) obtained from the simulation model and obtained from the AFS Yield Tables for good, medium and fair sites respectively for a growth period of 180 years. Because the number of trees per unit area is not available on the yield tables when age is under 20 years, possible values were assumed for each of the three sites.

An indication of the goodness of fit of the model of tree density to the yield table data is obtained by examining the differences between the two sets of data as shown in Figure 7.7, 7.8 and 7.9.

7.1.4 Basal area

The comparisons of the estimates on the basal area of pine from the simulation model and the AFS Yield Table data for the three different sites

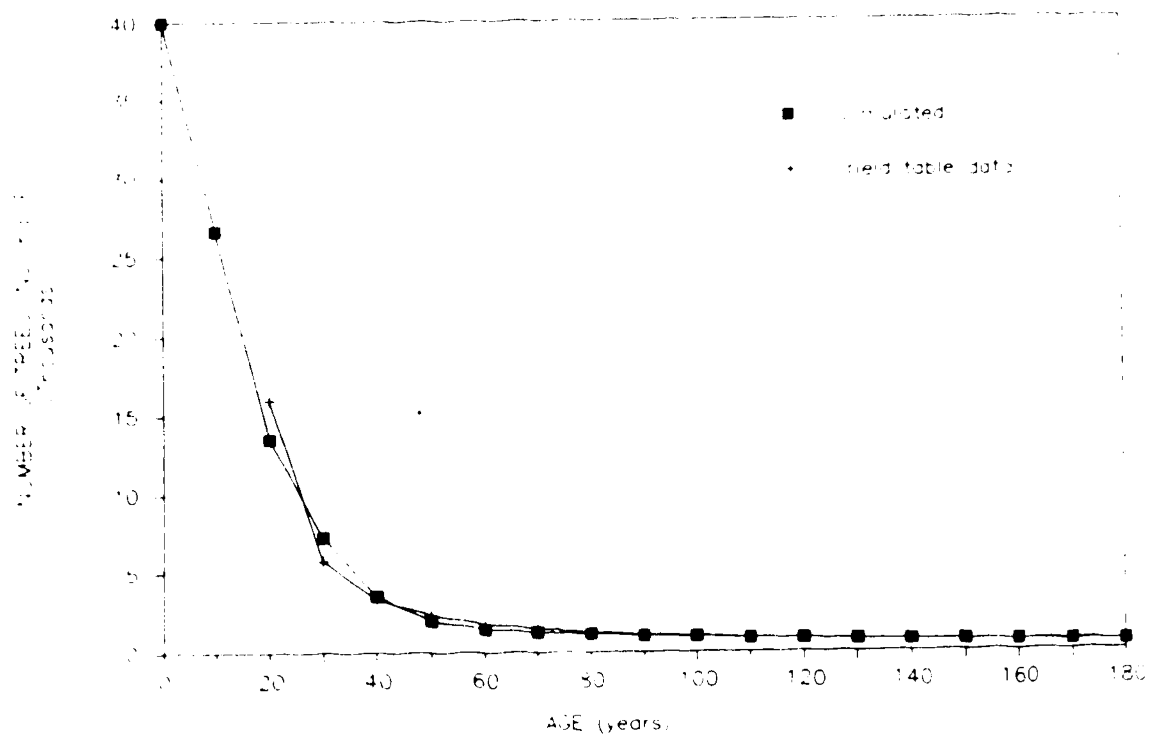


Figure 7.7 Simulated and yield table number of trees of pine on good sites

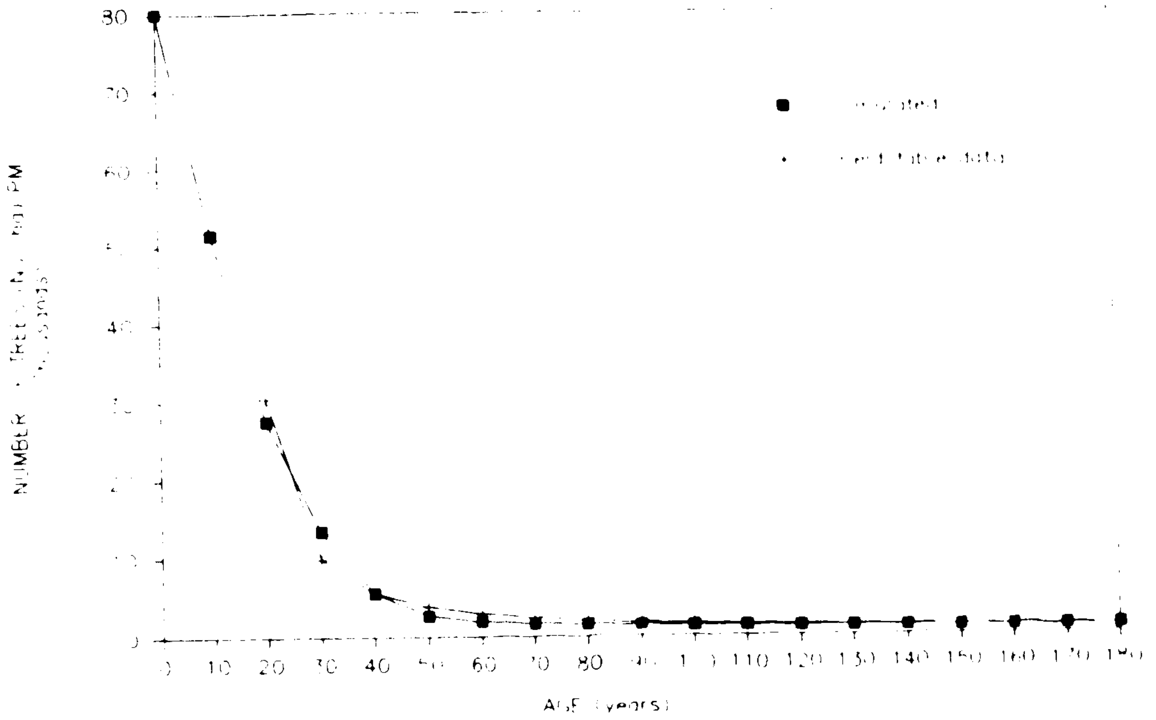


Figure 7.8 Simulated and yield table number of trees of pine on medium sites

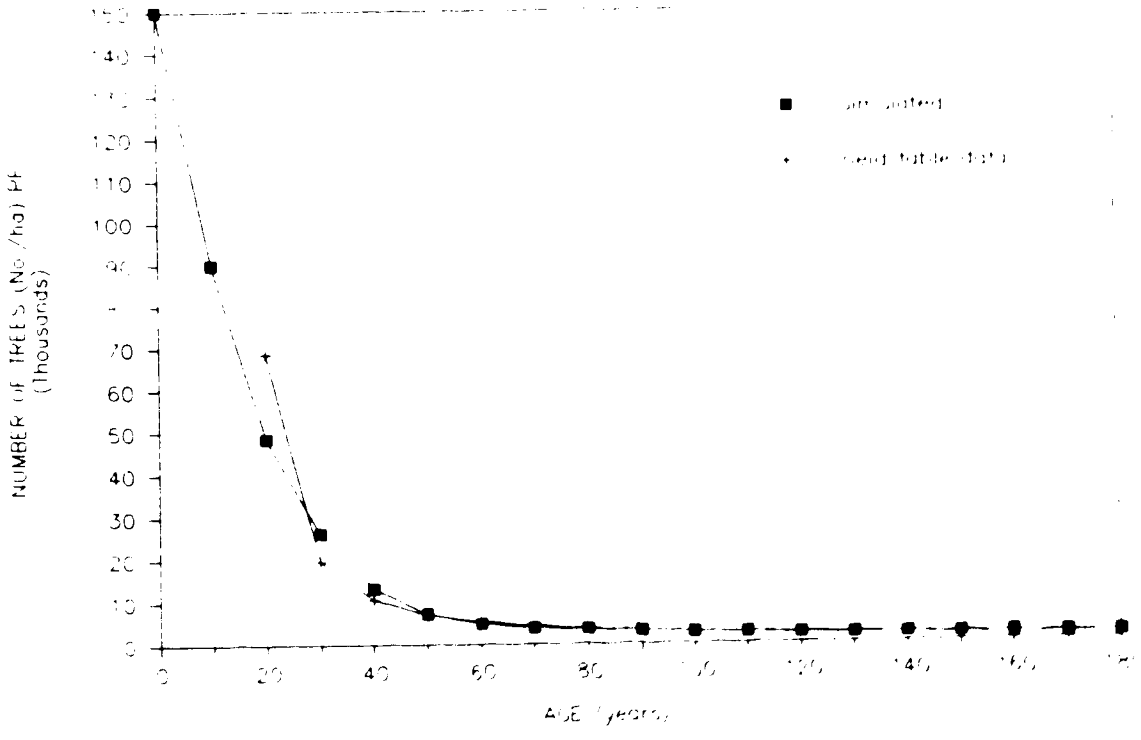


Figure 7.9 Simulated and yield table number of trees of pine on fair sites

for a period of 180 years are shown in Figure 7.10, 7.11 and 7.12, respectively. According to the basic geometrical relationships, the basal area per unit area was computed from the average tree diameter and the number of trees per unit area.

A reasonable agreement of the curves for good and fair sites and a fairly good agreement of the curves for the medium site were observed throughout the growth period. The basal areas estimated by the simulation model for good and fair sites, however, seem to be higher than the data in the Yield Tables. Those errors are mainly brought about by the overestimates of average diameter because the basal area is a function of the square of average diameter. For the medium site, there should be the similar error since a similar diameter growth curve was used(Figure 7.5). However, since the number of trees estimated by the simulation model for the medium site is below the data in the yield tables between age 40 to 120, the basal area curve appears to fit the yield table data very well.

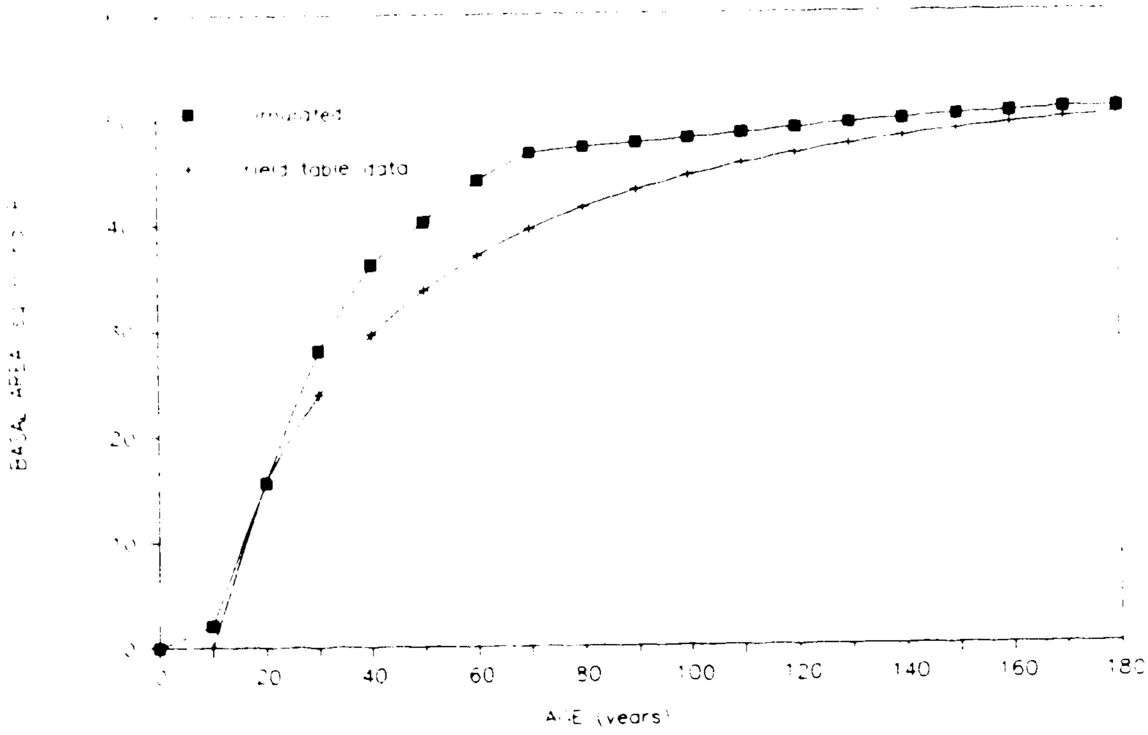


Figure 7.10 Simulated and yield table basal area of pine on good sites

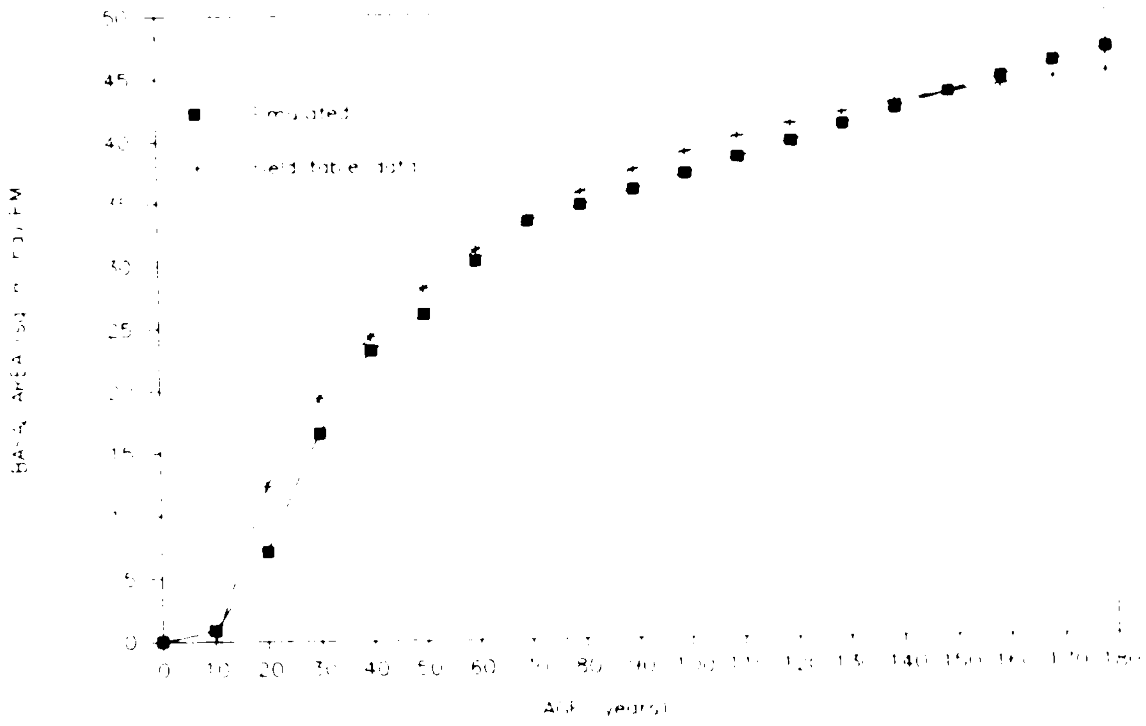


Figure 7.11 Simulated and yield table basal area of pine on medium sites

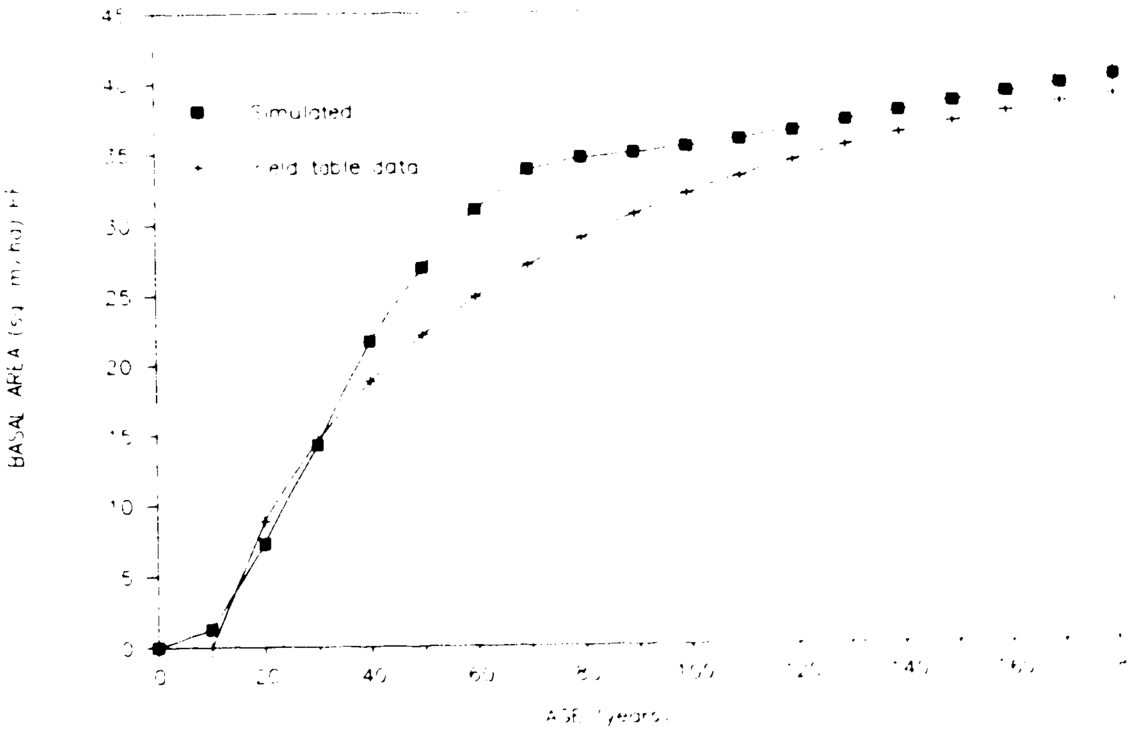


Figure 7.12 Simulated and yield table basal area of pine on fair sites

Therefore, the basal area per unit area is a combination of different variables. Any inaccurate prediction of individual variables may result in the deviation of predicted basal area from actual growth.

7.1.5 Volume

The comparisons of the estimates of gross volumes per unit area from the simulation model and the AFS Yield Tables on good, medium and fair sites for a growth period of 180 years are presented in Figure 7.13, 7.14 and 7.15. The volumes estimated using the simulation model were obtained by multiplying the volumes of a geometrical solid of the same diameter and height by the tree form factors, which were functions of tree ages derived from the yield tables. In detail, the volumes are the product of basal areas, heights, and form factors.

Similar to the basal area curves, a reasonable agreement of two curves for good and fair sites and a fairly good agreement of the curves for the medium site were seen throughout the growth period. The volume curves, however, showed larger deviations than the basal area curves. For the good site, maximum deviation is about 26% of the value given in the AFS Yield Tables. For the fair site, it is 30% of the value given in the AFS Yield Tables. The increase in deviations is caused by the overestimates of height growth after age 40. Furthermore, comparing the volume growth curves (Figure 7.13 to 7.15) with the basal area growth curves (Figure 7.10 to 7.12), the form factors might be considered as appropriate for the whole growth period.

A conclusion may be drawn that the mathematical equations employed in the tree growth models are capable of describing forest growth dynamics and depending on the accuracy of prediction for individual variables, the volumes could be calculated accurately by using the first principles. Thus, this approach is viable alternative to using regression analysis which was the procedure used to develop the equations in the AFS yield tables.

The overall simulation model prediction of tree growth was satisfactory. It closely followed the trend of the theoretical and actual performance, and, therefore, constitutes a sort of logical validation of the computer

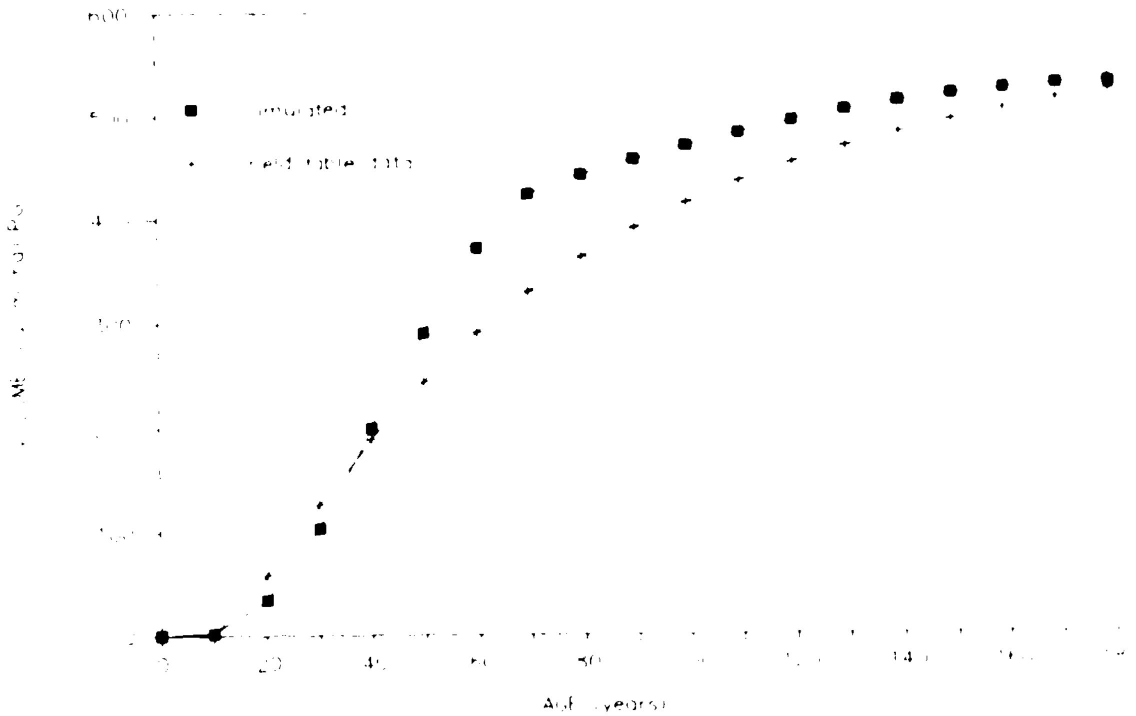


Figure 7.13 Simulated and yield table gross volume of pine on good sites

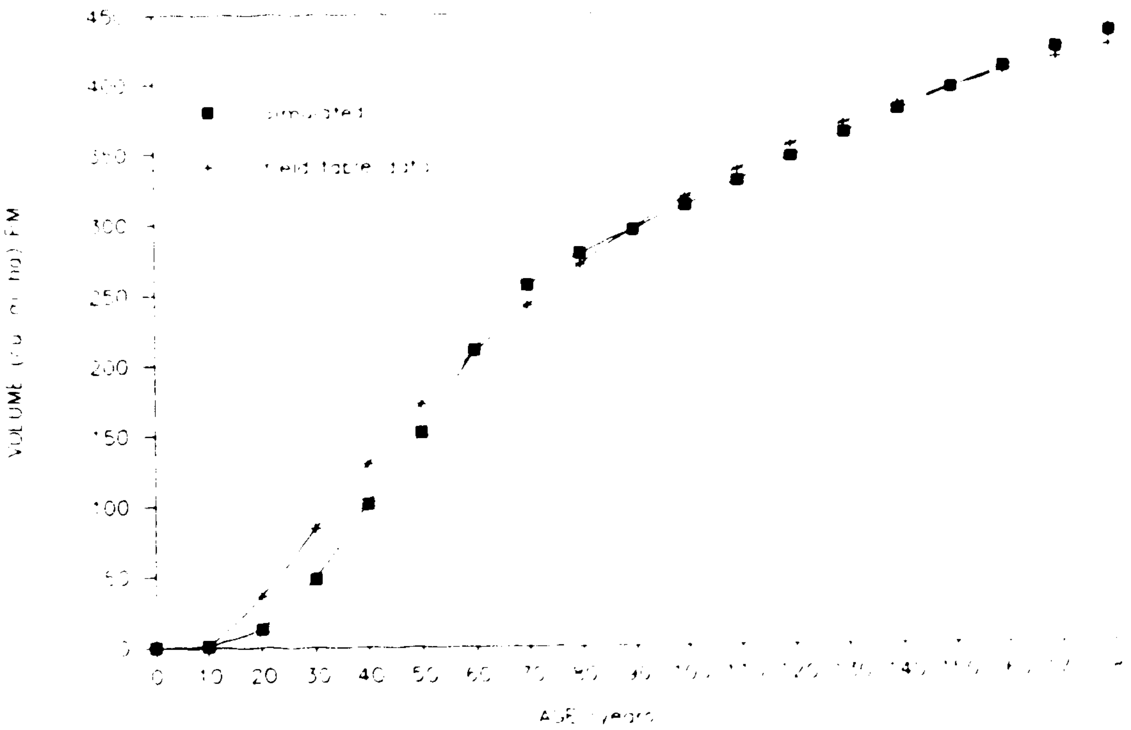


Figure 7.14 Simulated and yield table gross volume of pine on medium sites

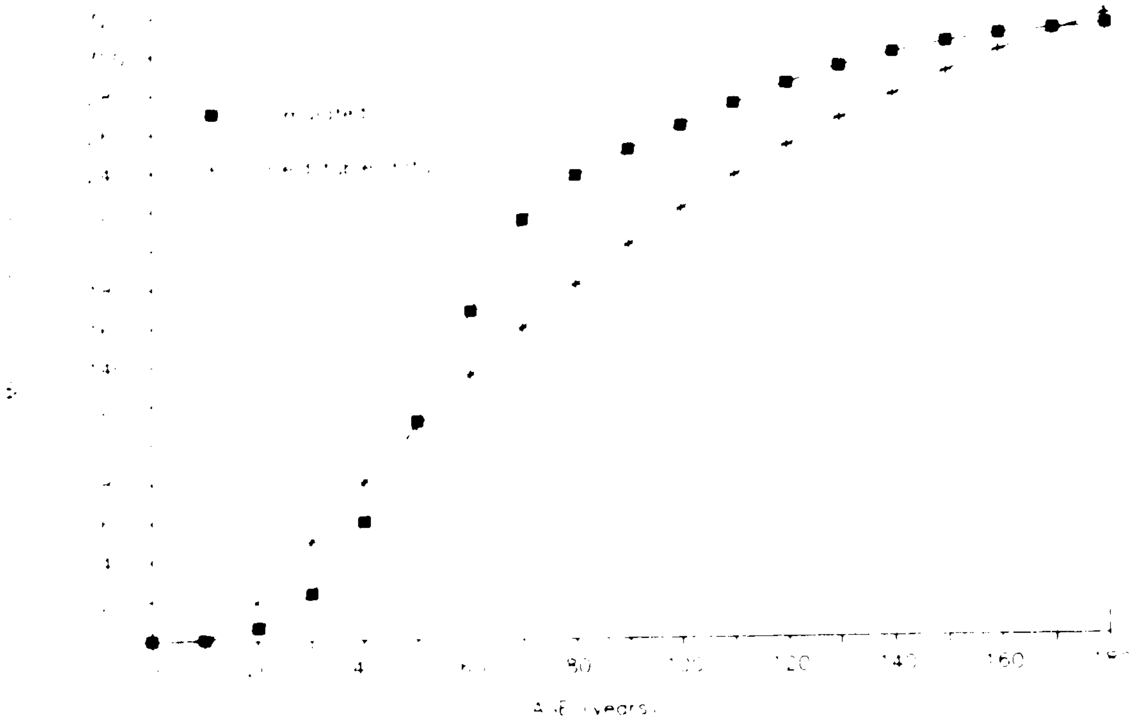


Figure 7.15 Simulated and yield table gross volume of pine on fair sites

model for simulating forest growth dynamics. For a complete validation, the field data is needed in order to make the comparisons.

7.2 Comparison of the Simulated Performance of Non-timber Resources with Other Models or Theoretical Curves

The mathematical relationships describing non-timber resource changes over time have not been well documented. The observed/measured data is either costly or impossible to collect. Therefore, it might not be meaningful at this point to validate the performance of non-timber models. However, comparing the predictions against results from other models or against the theoretical curves may provide a simple method of judging the model validity to some extents.

Figure 7.16 shows the estimates on potential grazing carrying capacity from the simulation model for good, medium and fair sites within the study area. Comparing Figure 7.16 with Figure 5.15, the estimates from the simulation model follow the same trend as the observed data, increasing or decreasing slightly and then decreasing dramatically as the tree age increases. For the good site, the carrying capacity reduces to the minimum value at a earlier age than the other two sites since trees grow faster resulting in understories being eliminated quickly.

Figure 7.17 shows the estimates on potential caribou population changes from the simulation model on the three sites. They basically follow the same curve since the caribou habitat was assumed to be a function of tree age. Food needed by caribou has been found only in the old growth forest. The estimate of the caribou population using the simulation model increases gradually to a certain level as the trees become old. Unlike caribou, moose population tends to decline quickly as trees age. This is due to moose require browse available in stands under 20 years old.

The estimates from the simulation model on moose and caribou hunting visitor-days are presented in Figure 7.19 and 7.20. Under a linear assumption, the recreational use of game hunting can only be allowed when the wildlife populations are greater than the minimum reserved populations. For moose hunting, the annual visitor-day varies from 0 to 1.4/q. sec. and for caribou hunting it only changes from 0 to 0.23/q. sec.

Figure 7.21 shows the simulation model estimates on mean ratings of forest scenic beauty change with trees growth for a period of 180 years. Based on the esthetic model developed by Benson and Ullrich (1981), the esthetic rating curve estimated using simulation model are in good agreement with Benson and Ullrich's curve (Figure 5.14).

The estimate of timber-water yield relationship on three different sites of Bolton basin was produced by the simulation model and plotted in Figure 7.22. Since the model was developed according to the WRENSS outputs provided by the Alberta Forest Service, the annual water yields would change from 194 *mm* to 120 *mm* over time following clearcutting. The water yields on a good site decline at the fastest rate since the total basal area and site height increase more quickly than medium and fair sites. Obviously, the water yield on a fair site has the lowest declining rate and stops at the highest yield (145 *mm*) by year 180.

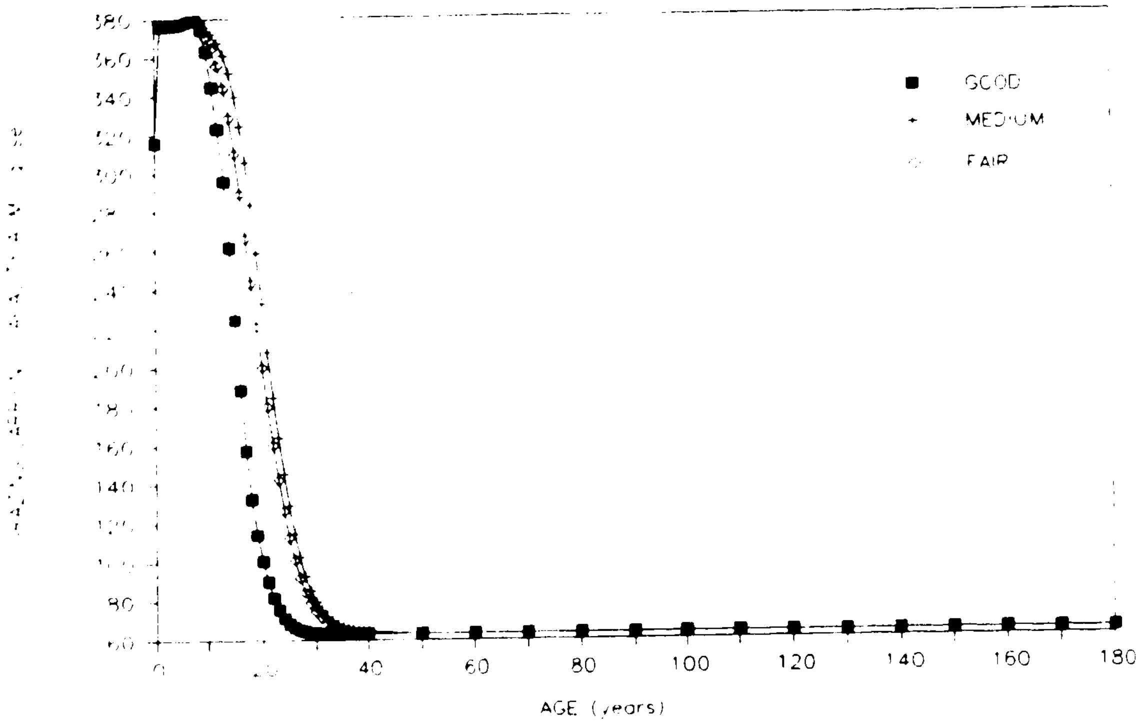


Figure 7.16 Simulated grazing carrying capacity of three different sites

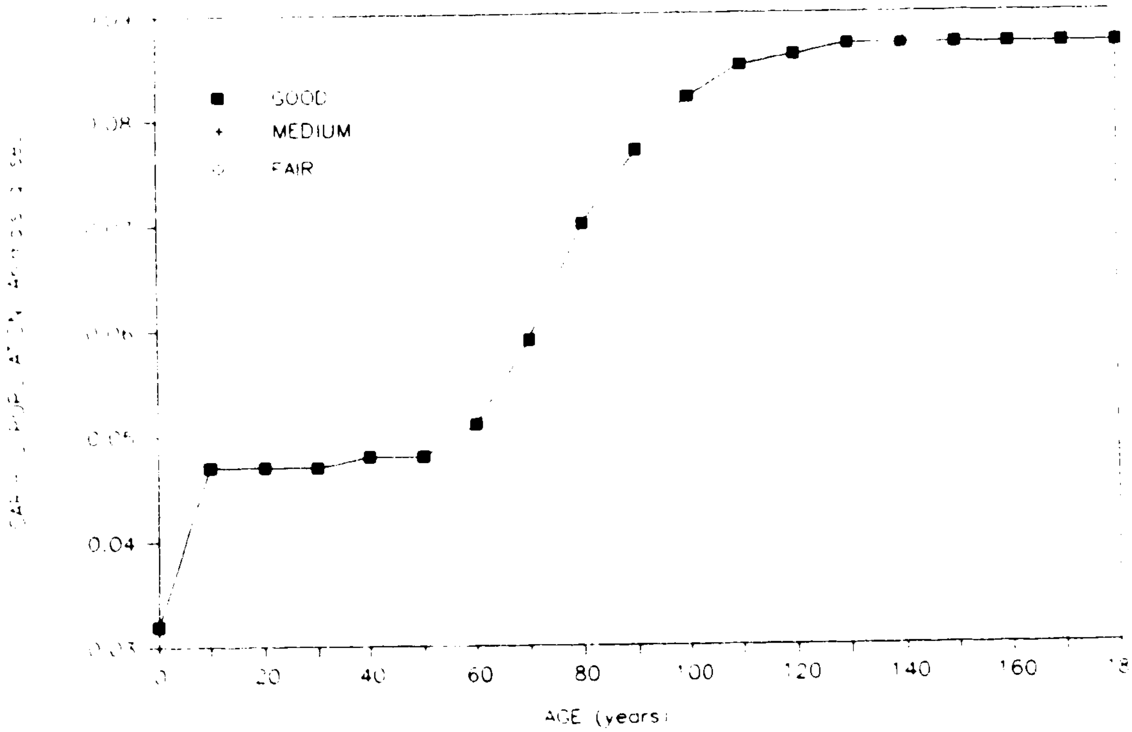


Figure 7 17 Simulated caribou population on three different sites

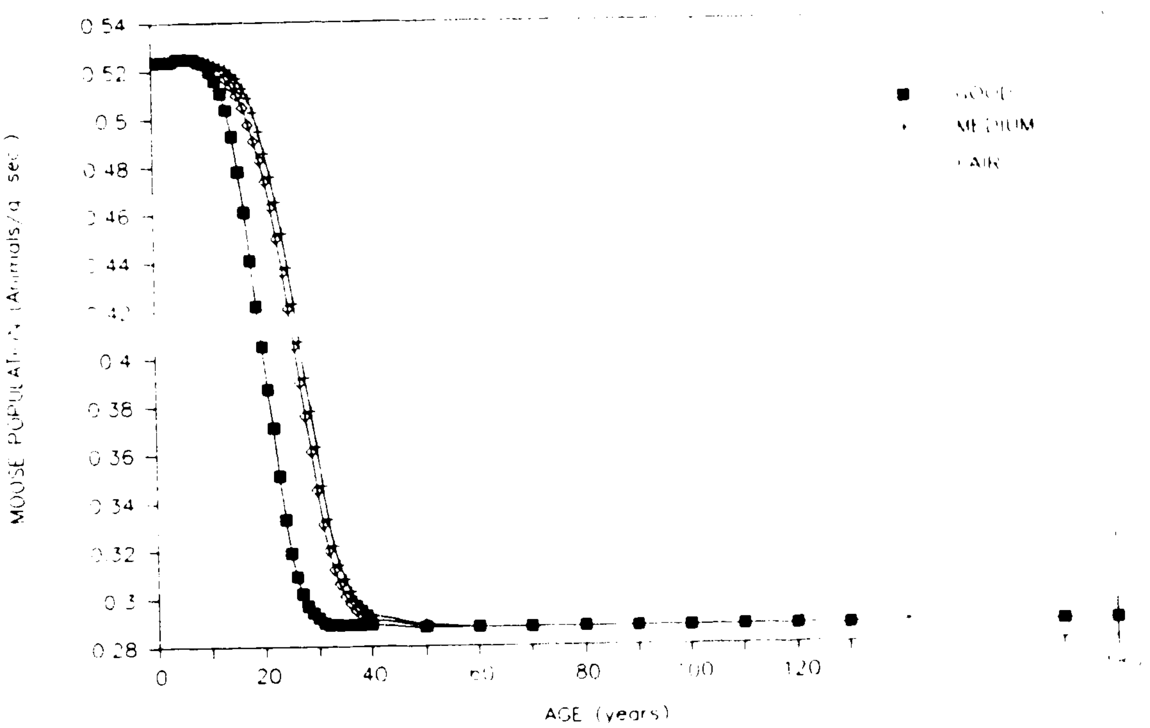


Figure 7.18 Simulated moose population on three different sites

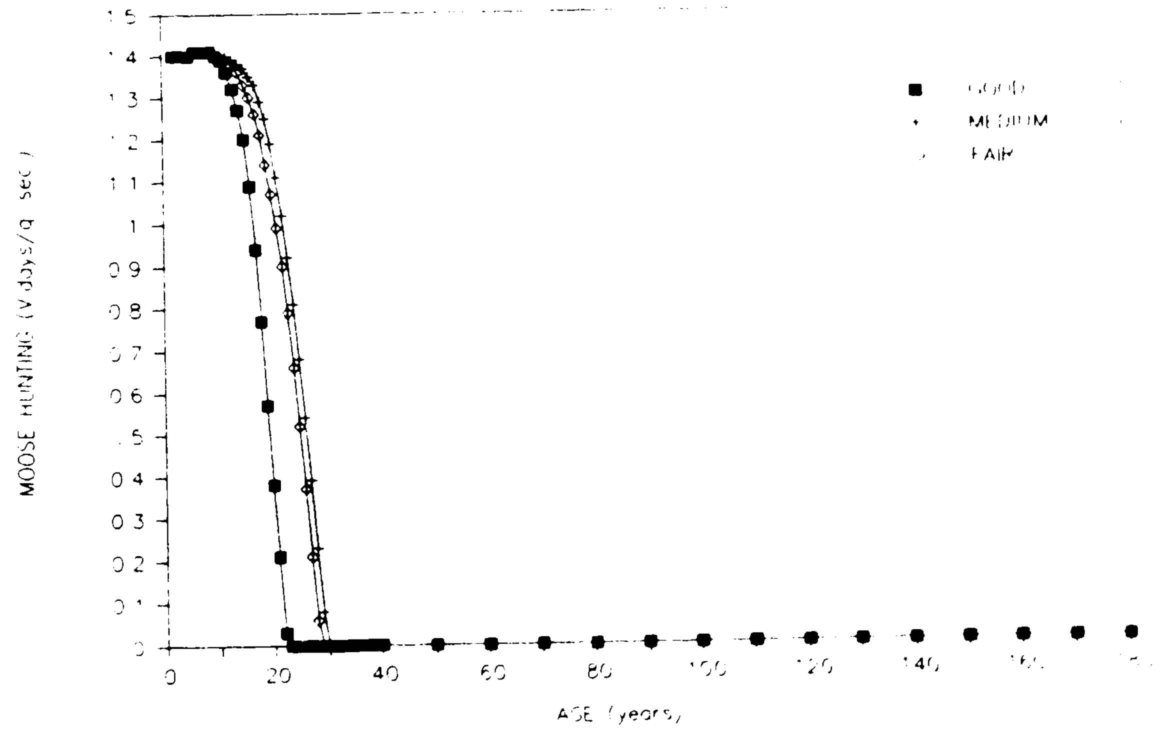


Figure 7.19 Simulated visitor-days of moose hunting on three different sites

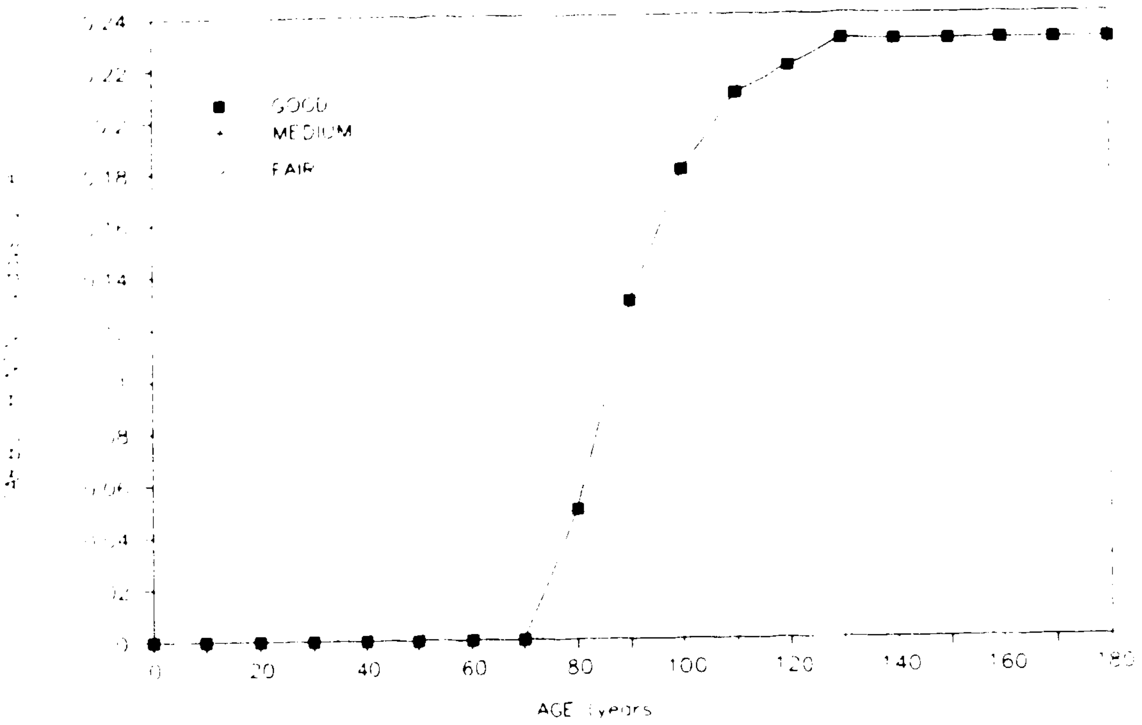


Figure 7.20 Simulated visitor-days of caribou hunting on three different sites

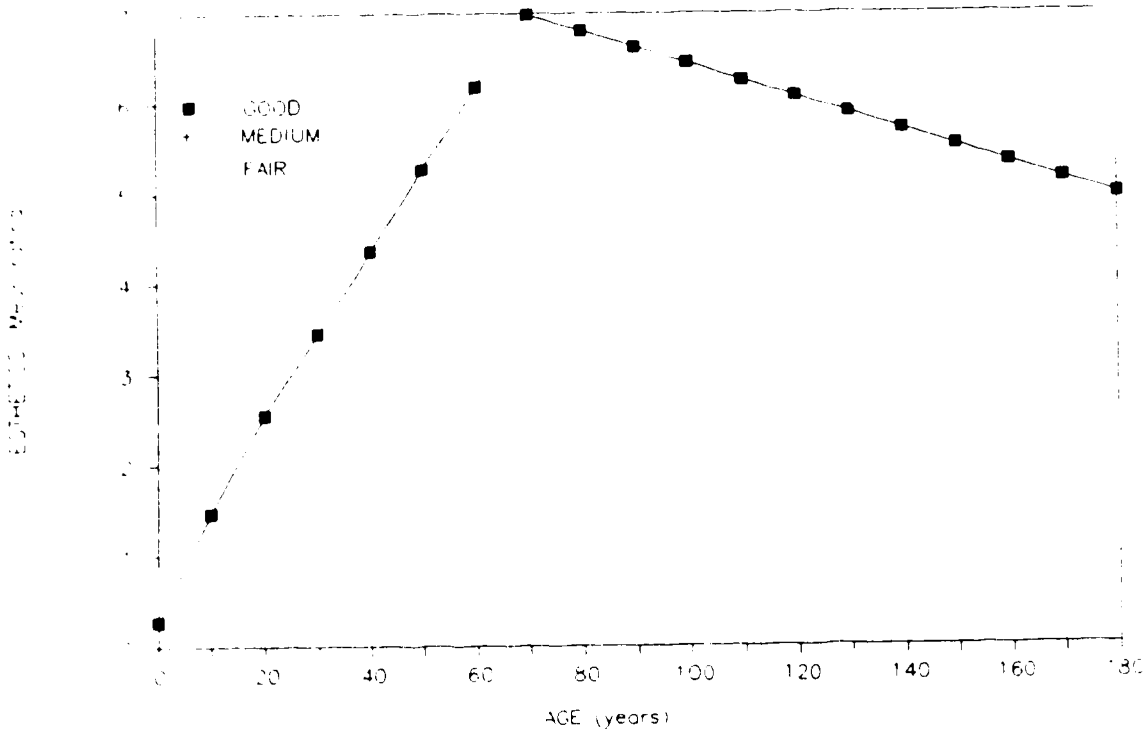


Figure 7.21 Simulated esthetics on three different sites

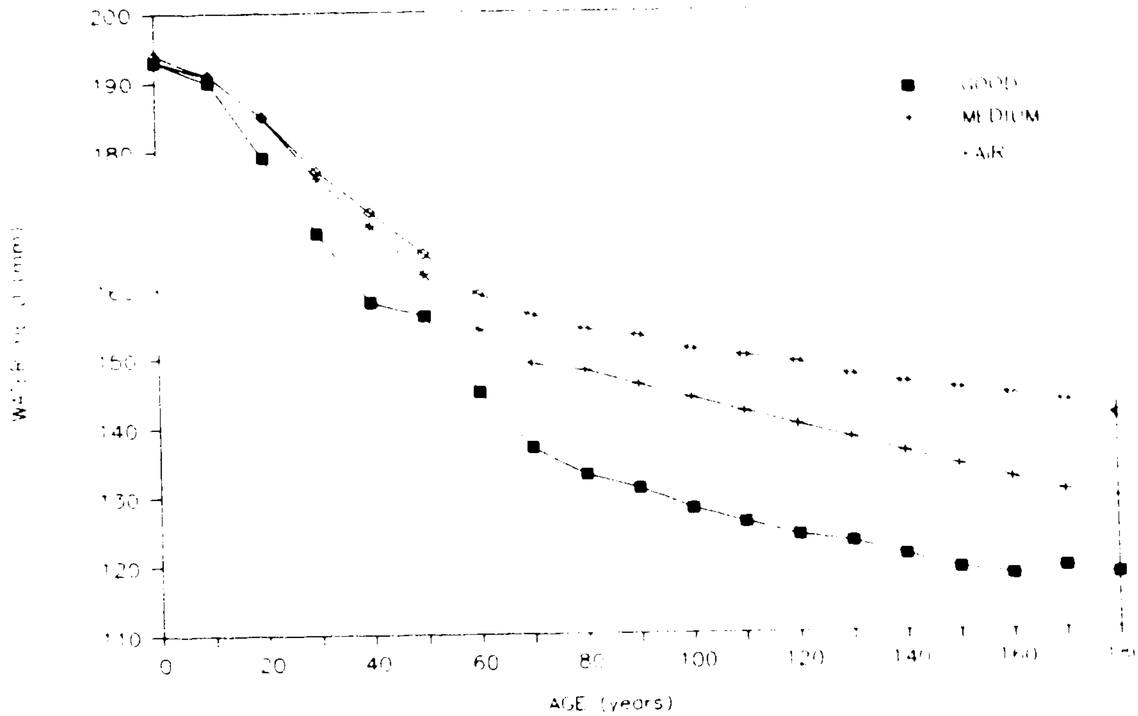


Figure 7.22 Simulated water yield on three different sites

7.3 Discussions on Model Performance

Logically, the simulation model performed properly. This can be seen from the validation parts of the model. The simulation model is capable of illustrating the dynamics of joint production of multiple resources. The submodels of the model have been validated only with the AFS Yield Tables and a few other models. The validation showed that the simulated tree growth was reasonably close to the AFS Yield Table data except for some basal area and volume results. In most cases, the simulation model provides data that overestimate basal area and gross volume to some degree. This is because the equations used to simulate the site height and the average DBH produce data that estimate the maturity of the forest to occur earlier than the data shown in the AFS Yield Tables. Although the plotted data from the tree growth model appear to be above those in the yield table, its shape agree with the theoretical growth curves. With the availability of field data, either the coefficients or the structure of the growth models can be modified so that a better prediction over the range of the yield tables can be expected for the whole growth period by using the simulation model.

Since the whole simulation model has not been validated with actual measured data, the results predicted by the model can be interpreted only as indications of what might be expected in a forest.

8 RESULTS AND DISCUSSIONS

8.1 Applications of the Simulation Model in Integrated Resource Planning

To demonstrate how the simulation model might be used to analyze management alternatives, two timber harvest strategies were simulated for on the Berland forest area. One such strategy attempted to develop the best possible harvest schedule of clearcutting and thinning in order to convert the forest to a regulated state. A second strategy attempted to increase caribou population and at the same time, to maintain other activities at a uniform level.

One process for deciding how public land should be managed involves a focus on tradeoffs. A tradeoff is a relationship between two or more effects of a change in some condition (such as the condition of the forest). The relationship signifies a difference between the initial condition and the new one which the change would bring about (Brown, 1981). For example, an alternative maximizing timber yield--A--would require a longer (80 to 120 years) rotation. It would result in fairly low livestock forage production, medium or better scenic quality depending on clearup of logging debris, low moose habitat quality and recreational hunting uses, and medium caribou habitat quality and medium water yield. An alternative based on maximizing grazing carrying capacity--B--would require a shorter (approximately 50 years) rotation period resulting in low caribou population and scenic quality, small diameter trees, high moose population, high water yield, and high hunting visitor-days. The advantage of examining tradeoffs between forest conditions is that attention is focused on actual quantities and values of those quantities rather than on abstract values.

Charts depicting tradeoffs can be very helpful in formulating viable alternatives (Brown, 1981). One common tradeoff chart uses individual graphs to show how specific resources might respond to changes in one element of the forest environment-in this case, tree age. Figures 8.1 to Figure 8.11 represent 100 planning units (quarter sections) within the study area that are managed under a 50 year rotation and with/without precommercial thinning. Figures 8.12 to Figure 8.22 show the same sites that are managed under two different rotation periods, 50 years and 100 years. Similar graphs could be developed with an element other than age as the common variable.

Figures 8.1 to Figure 8.4 show the simulation results of timber volume over a 100-year period. After the conversion period of 50 years to a normal forest the gross volumes to be harvested from all species, except the regenerated pine species, decrease to zero. The pine volumes to be harvested per year would fluctuate around an average value of 60000 cubic meters if no thinning activities occur. With precommercial thinning at age 30, however, the annual volumes to be harvested could decrease to an average level of 40000 cubic meters with a residual of 50 percent of initial basal area and 2000 stems per ha. which is less than a half of prethinning density. Thinning would result in the timber growing larger and faster.

Responding to thinning, the livestock carrying capacity of the area (Figure 8.5) would increase from age 30 and reached a steady state of about 45000 AUMs per year after the conversion period. This provides 2000 more animal-unit months annually on the thinned sites than on the unthinned sites. For wildlife (Figure 8.6 and Figure 8.7), the caribou population is unchanged, since thinning does not provide more food for them. Moose population increases about 7.4 percent due to the decrease of basal area.

Figure 8.8 and Figure 8.9 show the recreational hunting changes brought by thinning. Depending largely on the number of wildlife, those changes follow the same trends as that of the wildlife populations. The caribou hunting visitor-day reduces to zero, and the moose hunting visitor-day decreases about 7.5 percent. According to Figure 8.10, the rating scale esthetics is not influenced by thinning because it was assumed that the scenic quality changes only as the age increases since clearcutting.

The effects of thinning on annual water yields are shown in Figure 8.11. From age 30, when thinning is implemented, the annual water yields start to increase over that of unthinned sites and reach the value of 188 *mm* at the end of the conversion period. This indicates a 3 *mm* surplus yield gained from thinning.

By running the simulation with different thinning time and thinning levels, the optimal thinning regime could be determined. Those simulation predictions would provide the resource manager with some quantitative information regarding how resources will respond when different management strategies are actually implemented.

To increase caribou population to a certain level, the forest must be managed under a longer rotation (> 90 years). However, a longer rotation will reduce the annual timber production, forage production, water yield, and so on. In order to balance these problems, it is necessary to select a optimal rotation to convert the forest to a regulated state.

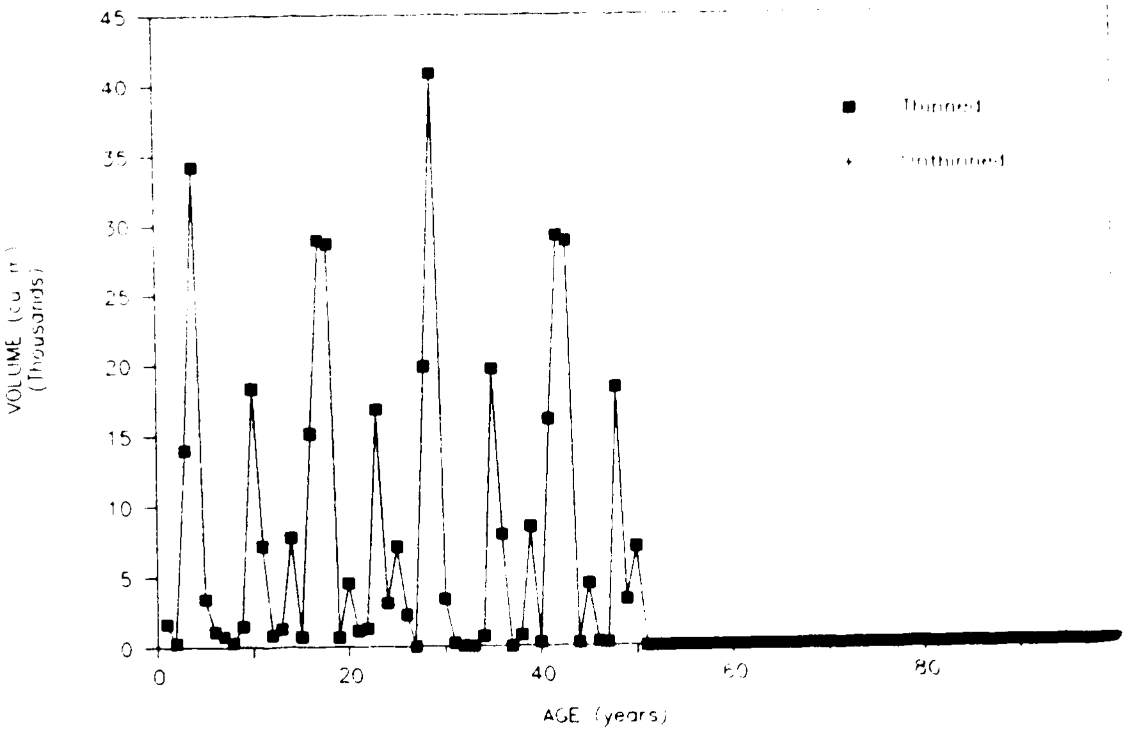


Figure 8.1 Simulated annual harvested volume of white spruce for th and unthinned stands (without regeneration)

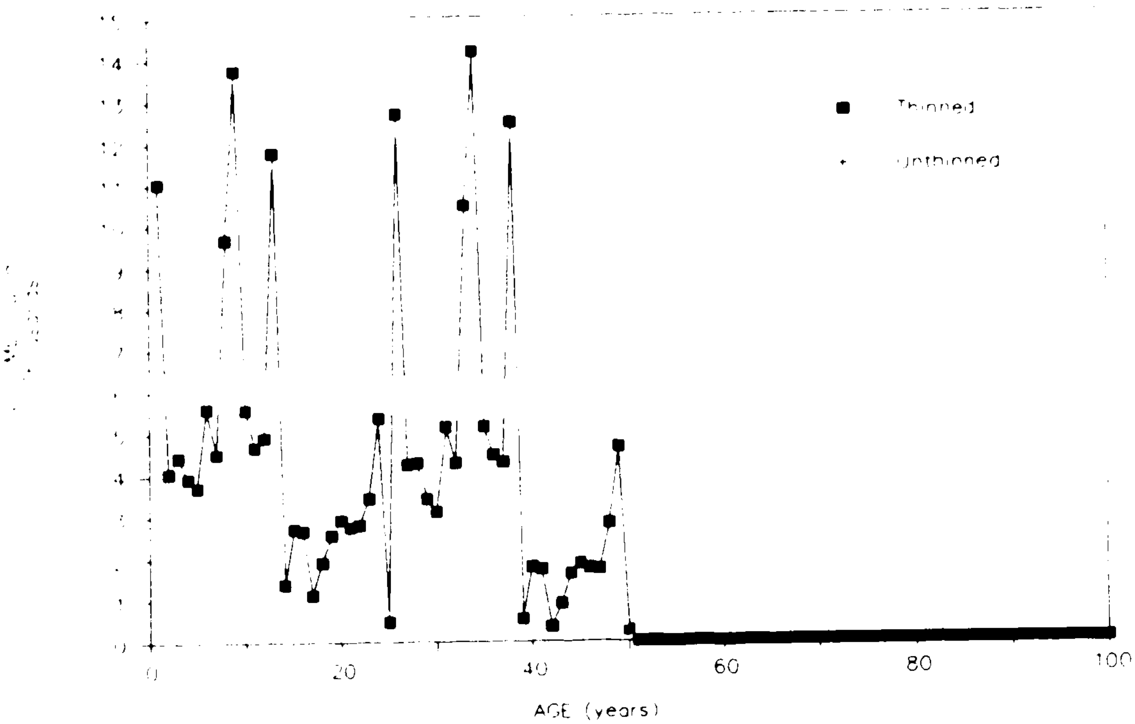


Figure 8.2 Simulated annual harvested volume of black spruce for thinned and unthinned stands (without regeneration)

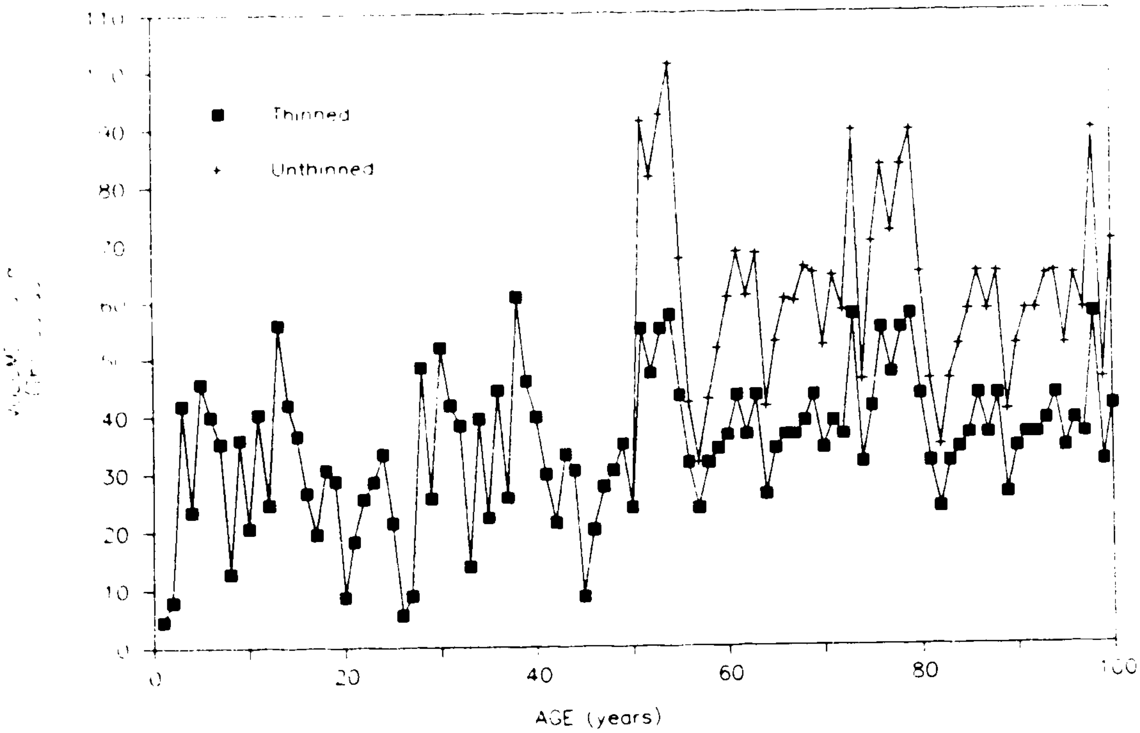


Figure 8.3 Simulated annual harvested volume of pine for thinned and unthinned stands (with regeneration)

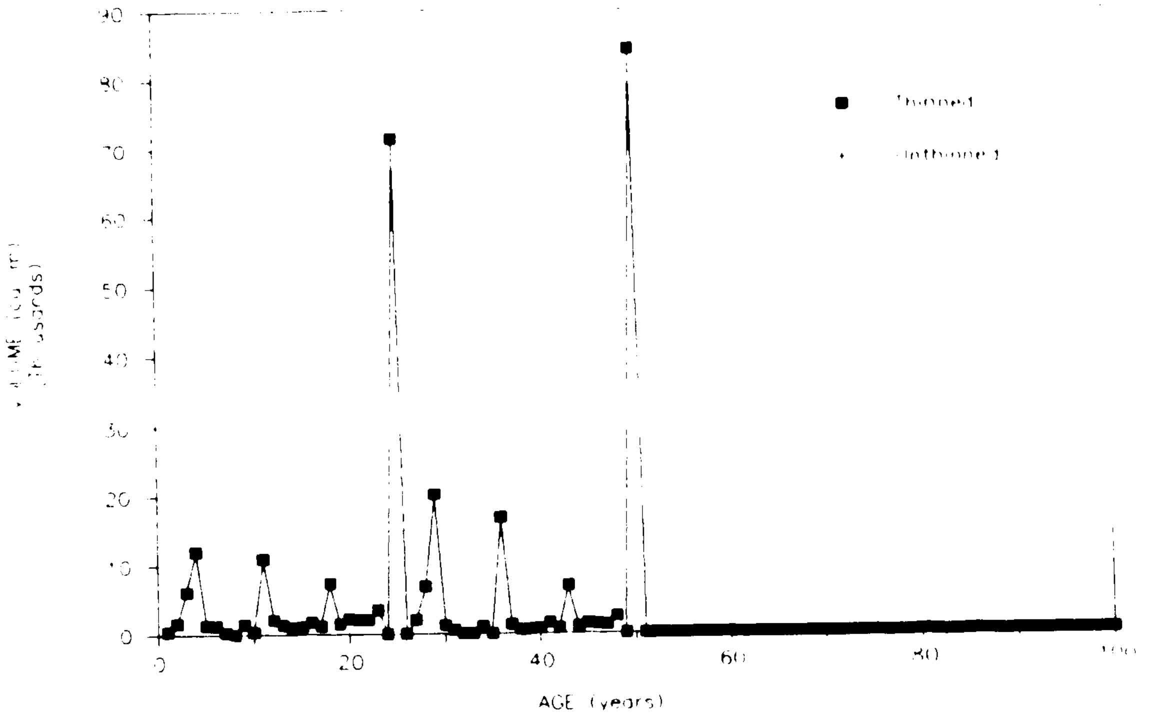


Figure 8.4 Simulated annual harvested volume of aspen for thinned and unthinned stands (without regeneration)

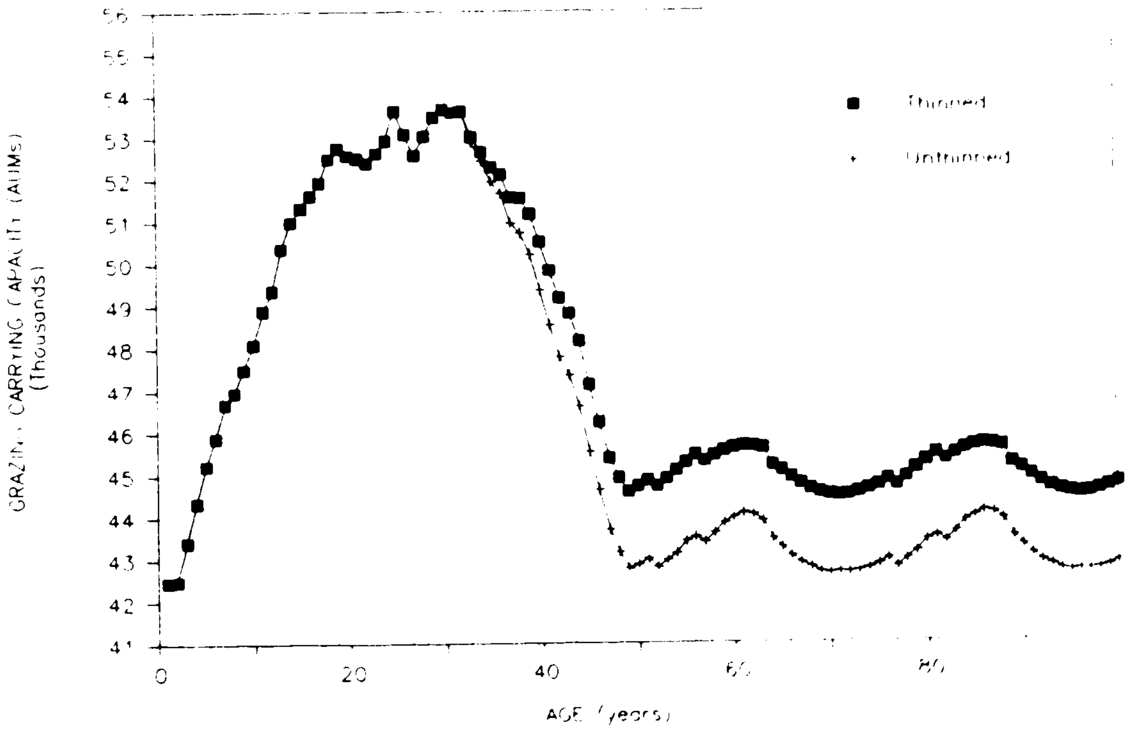


Figure 8.5 Simulated annual grazing carrying capacity changes for thinned and unthinned stands

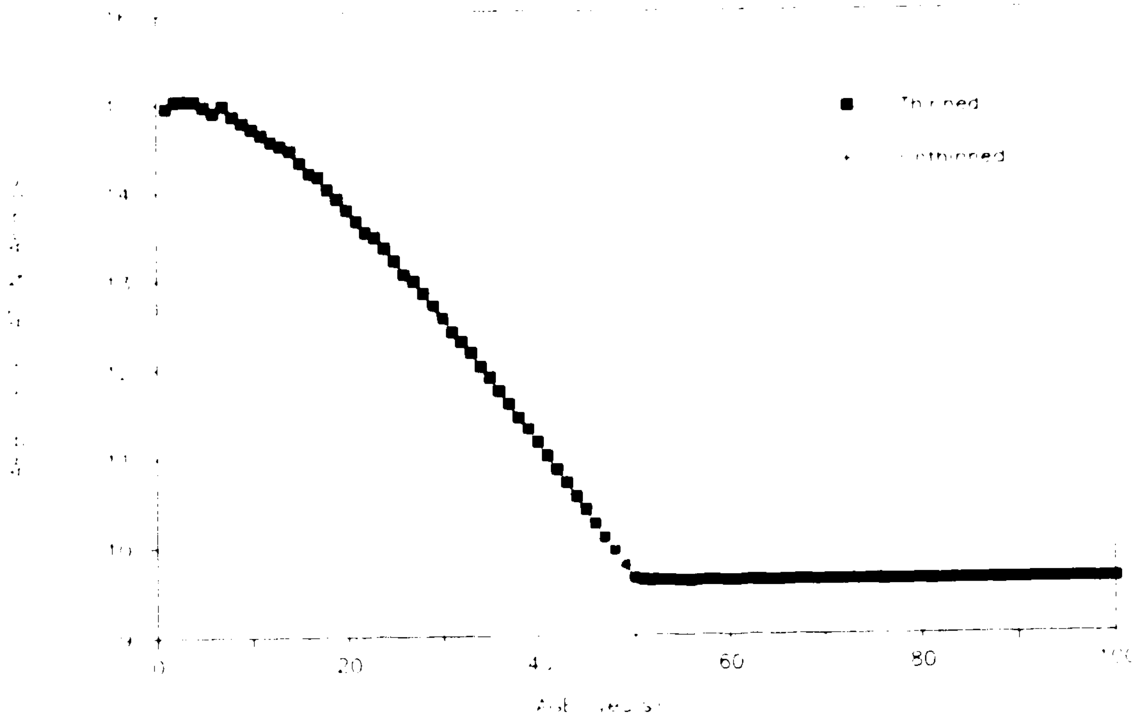


Figure 8.6 Simulated annual caribou population changes for thinned and unthinned stands

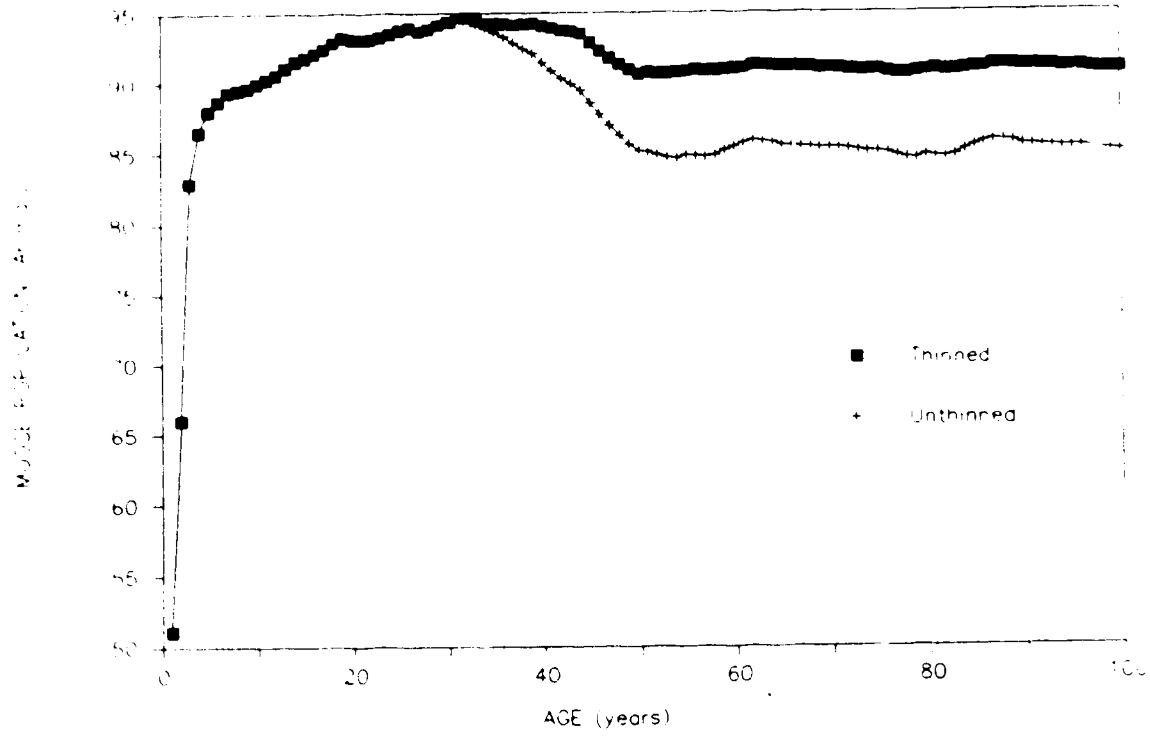


Figure 8.7 Simulated annual moose population changes for thinned and unthinned stands

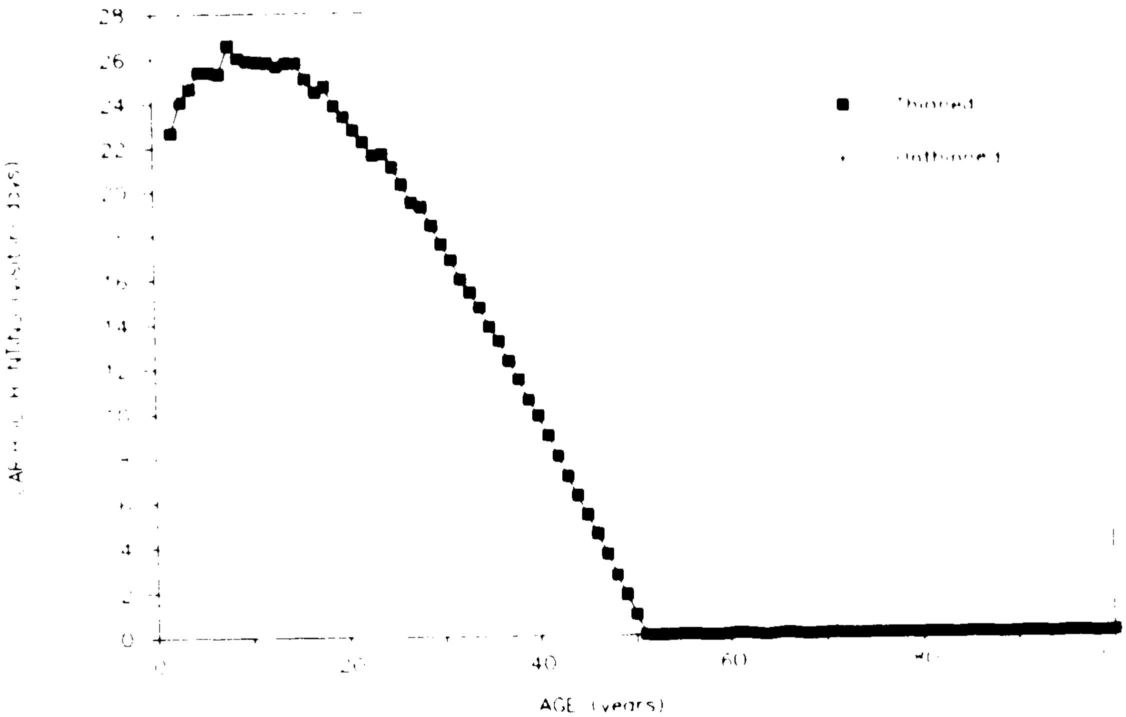


Figure 8.8 Simulated annual visitor-days changes of caribou hunting for thinned and unthinned stands

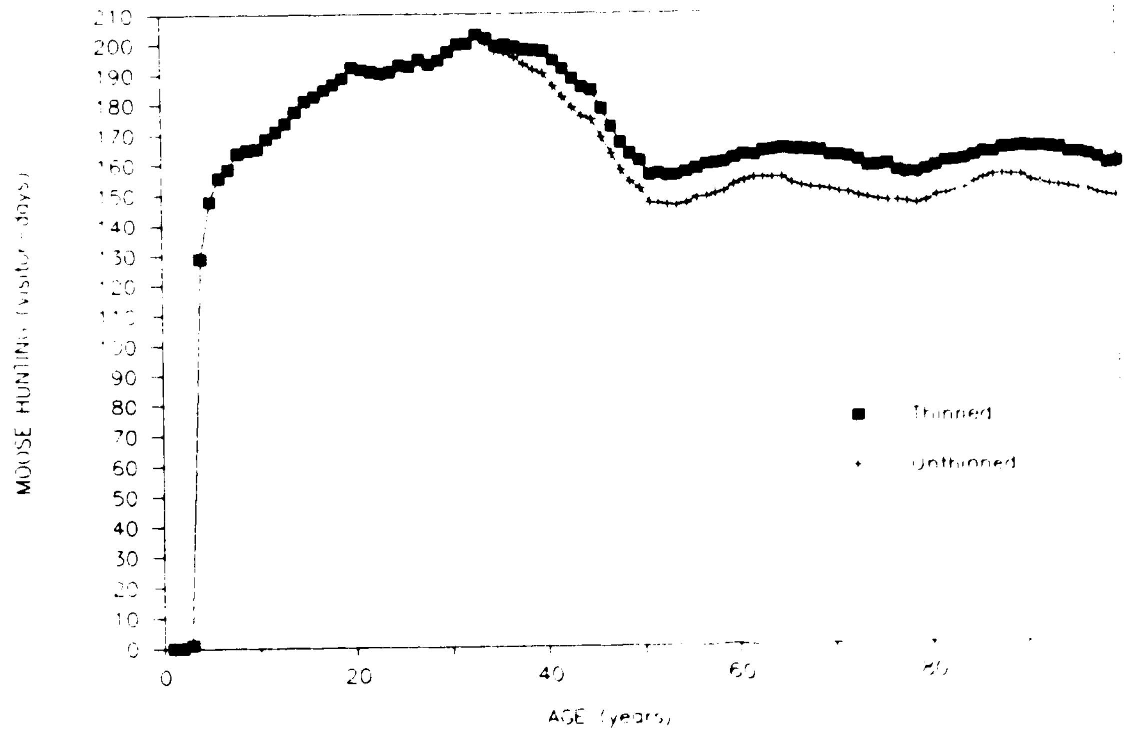


Figure 8.9 Simulated annual visitor-days changes of moose hunting for thinned and unthinned stands

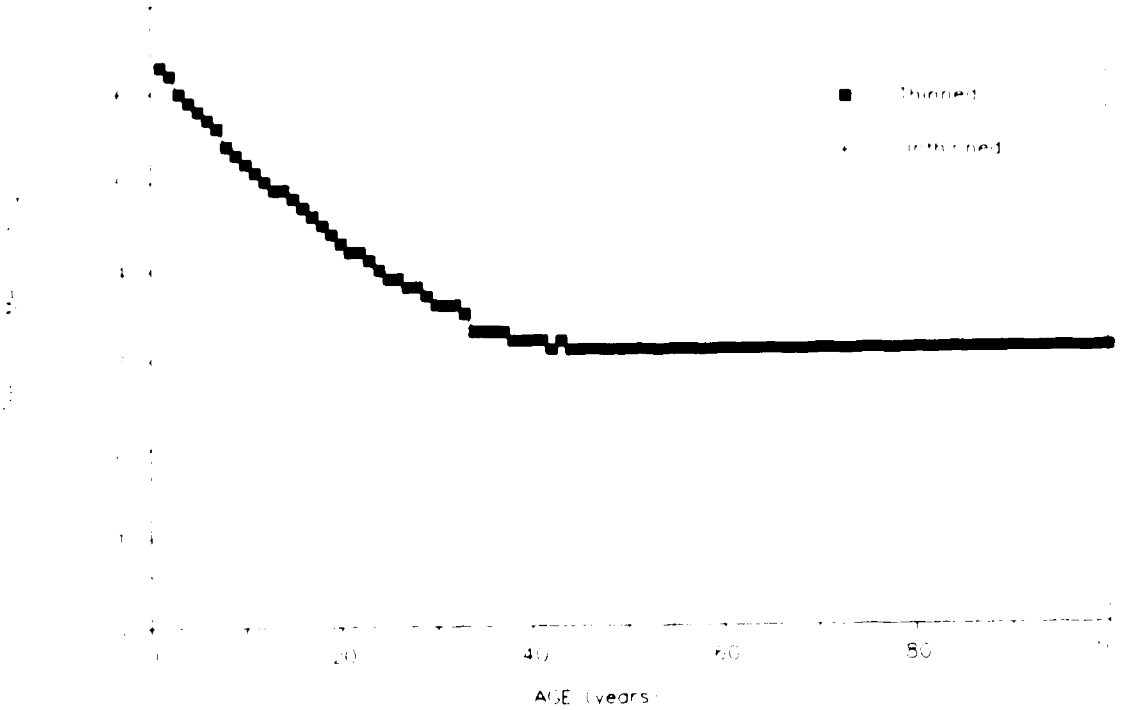


Figure 8.10 Simulated annual mean rating changes of natural scenic beauty for thinned and unthinned stands

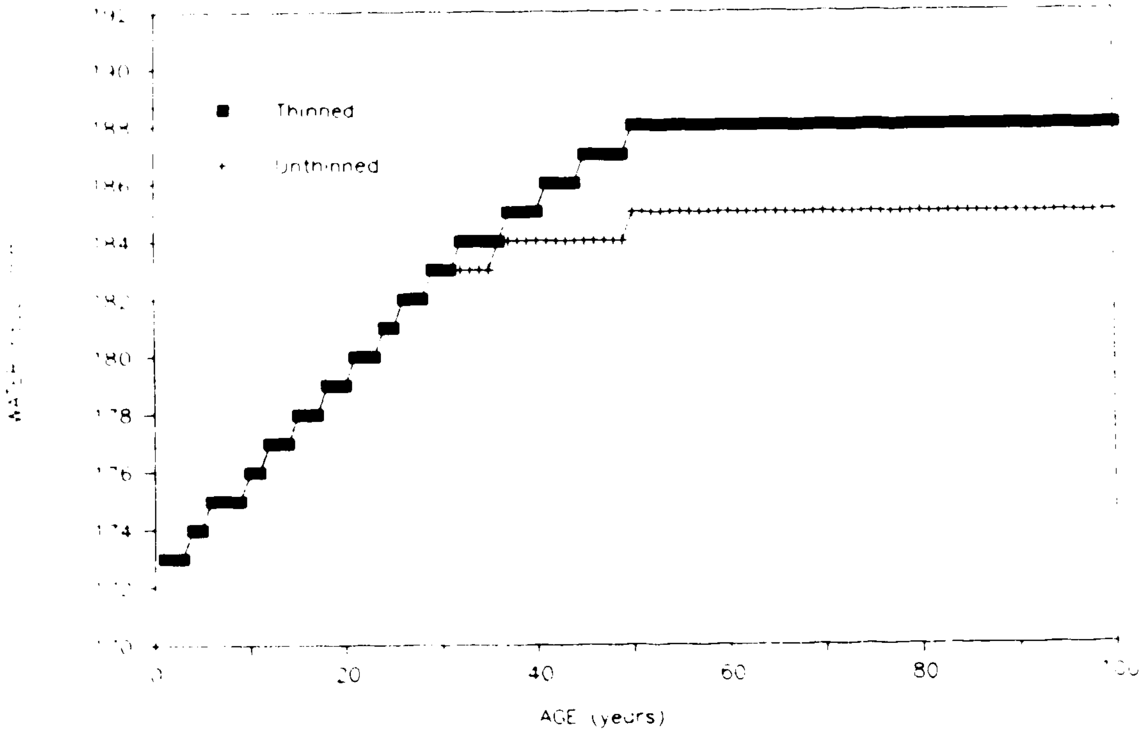


Figure 8.11 Simulated annual water yield changes for thinned and unthinned stands

If a 100-year rotation, a conversion period of 100 years, and a cutting interval of 1 year are assumed, an area regulation approach would require 64 hectares (1 quarter section) to be patch cut. If a 50-year rotation and a conversion period of 50 years are used, this patch cut will be 128 hectares annually. Figure 8.12 to Figure 8.15 indicate the simulation results over a 100-year period for the annual timber harvest under both 50-year and 100-year rotations. The average harvest rate is higher for both conversion and post-conversion periods, if the forest is managed under 50-year rotation.

The caribou population changes under two rotations are plotted in Figure 8.16. Obviously, caribou population under 100-year rotation is larger than that under 50-year rotation. For the 100-year rotation forest, after a period of approximately 80 years the caribou population begins to stabilize at 11.5 animals per year. This represents an increased population of 18% over the caribou population in the forest under 50-year rotation. Depending on young forest to provide good habitat, moose population (Figure 8.17) is smaller if using a 100-year rotation. The reduction is around 10.6%.

For other resources, the impact of rotation is also significant. Figure 8.18 shows the simulated annual grazing carrying capacity changes under 50-year and 100-year rotation. The total animal unit months of the shorter rotation forest increase as the old trees are removed, and after year 30 it starts to decline. At the end of conversion period, they appear to remain relatively stable. For the 100-year rotation forest, the carrying capacity would drop to 31000 AUMs when it moves into a regulated state. This represents a capacity loss of 28 % over that provided by the 50-year rotation forest. From Figure 8.19 and Figure 8.20, when rotation age increases from 50 to 100, the caribou hunting changes from prohibited to 6 visitor-days per year, and moose hunting reduces from 150 visitor-days to 90 visitor-days. Figure 8.21 shows the comparison of scenic quality changes. For both rotation periods, the mean rating will decrease. The difference is that with a shorter rotation, this rating decreases quickly and stops at a lower value (3.1), and with a longer rotation, the rating decreases gradually and ends up with a value of 4.6 because of more older trees present. Figure 8.22 indicates water yield fluctuations under each different rotation period. With a 50-year rotation, the water yields can be expected to increase to 185 mm, and with the 100-year rotation, it can only get to 182 mm.

These types of multiple production simulation could be applied to many potential managerial problem analyses. Studies relating to determining optional unit configurations for joint production management would certainly be a possibility. The impact of various forest regulation control measures could be examined for the same planning area. The effects of social and environmental impact constraints or logging needs could be estimated over any time period. The scope of the simulation analysis in the joint production area offers a wide array of capabilities.

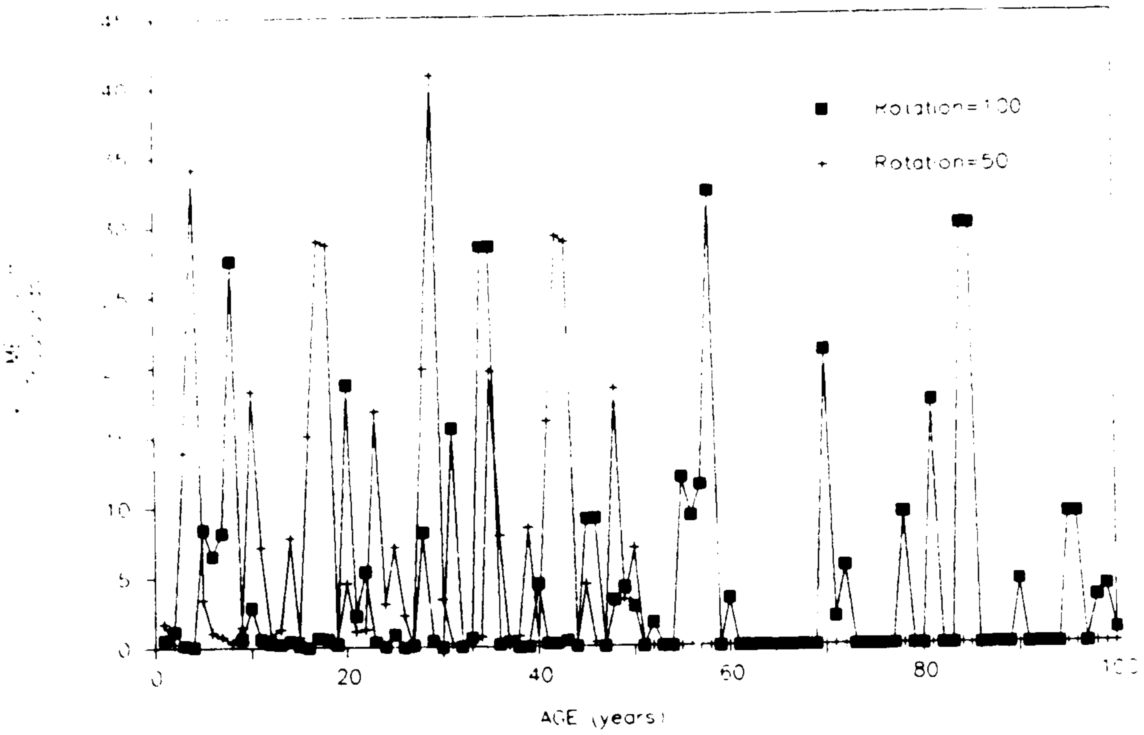


Figure 8.12 Simulated annual harvested volume of white spruce under 50 and 100 year rotations (without regeneration)

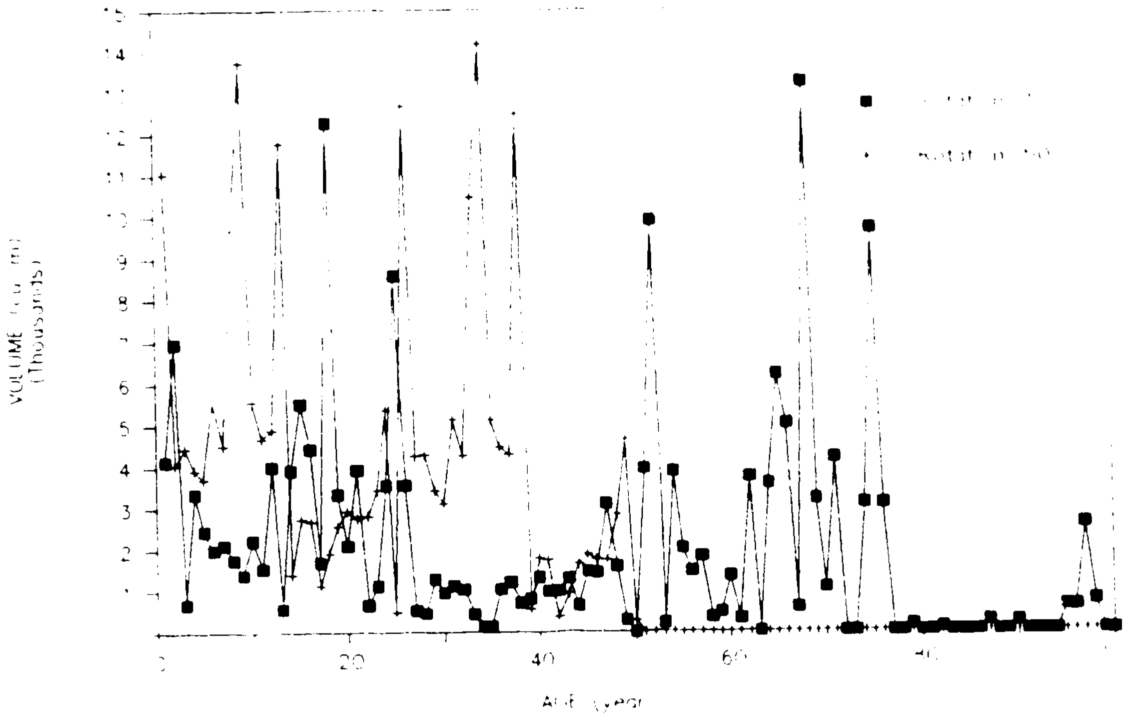


Figure 8.13 Simulated annual harvested volume of black spruce under 50 and 100 year rotations (without regeneration)

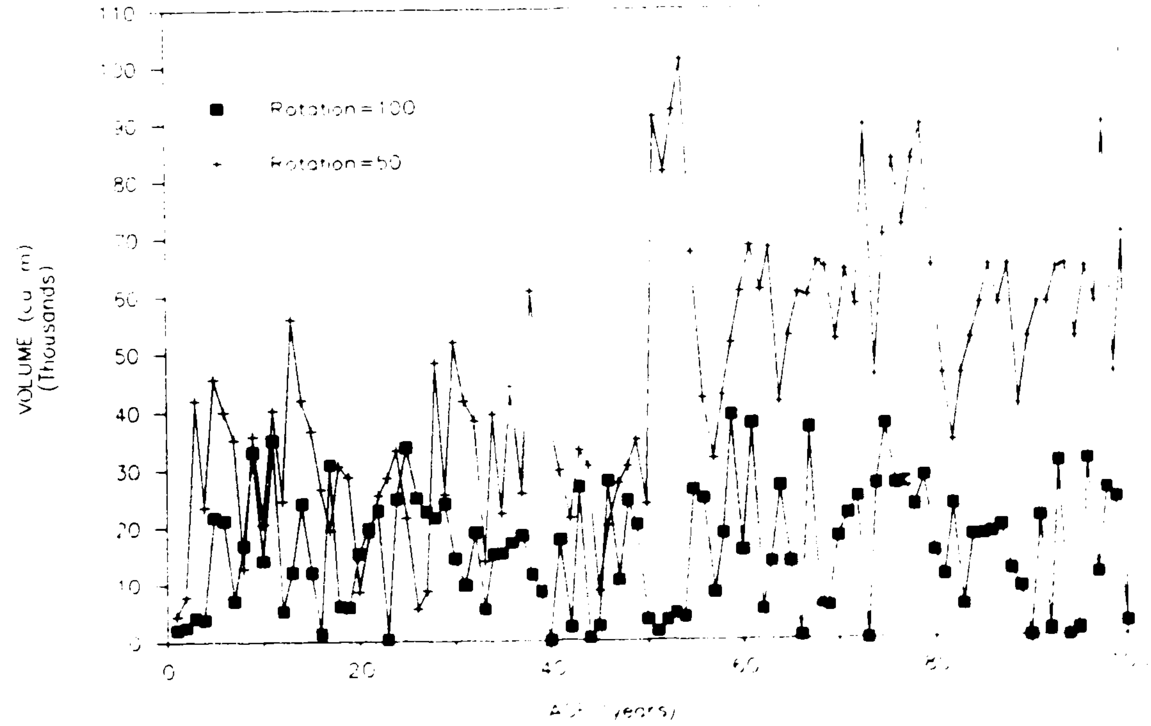


Figure 8.14 Simulated annual harvested volume of pine under 50 and 100 year rotations (with regeneration)

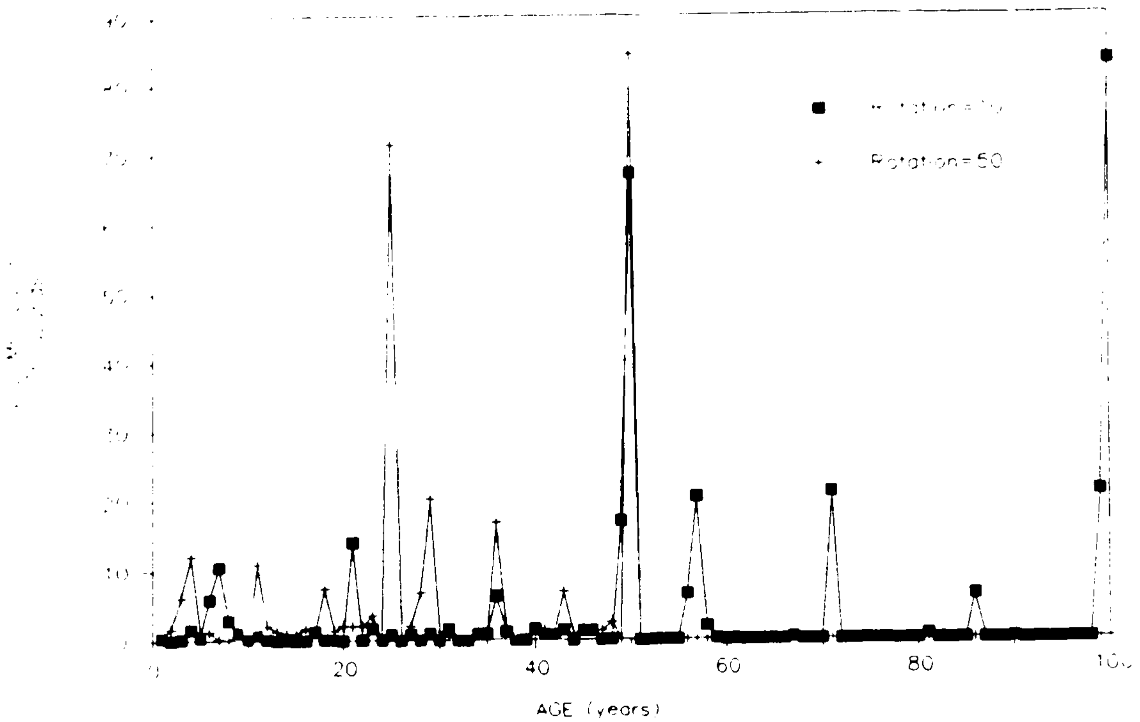


Figure 8.15 Simulated annual harvested volume of aspen under 50 and 100 year rotations (without regeneration)

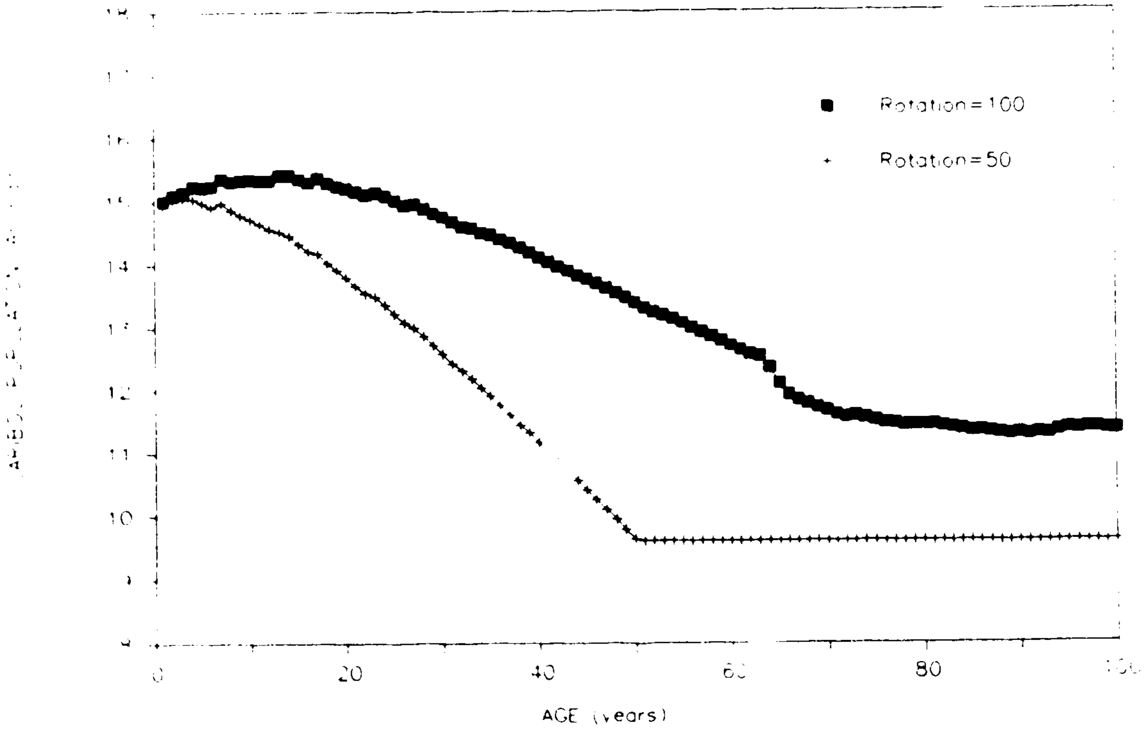


Figure 8.16 Simulated annual caribou population changes under 50 and 100 year rotations

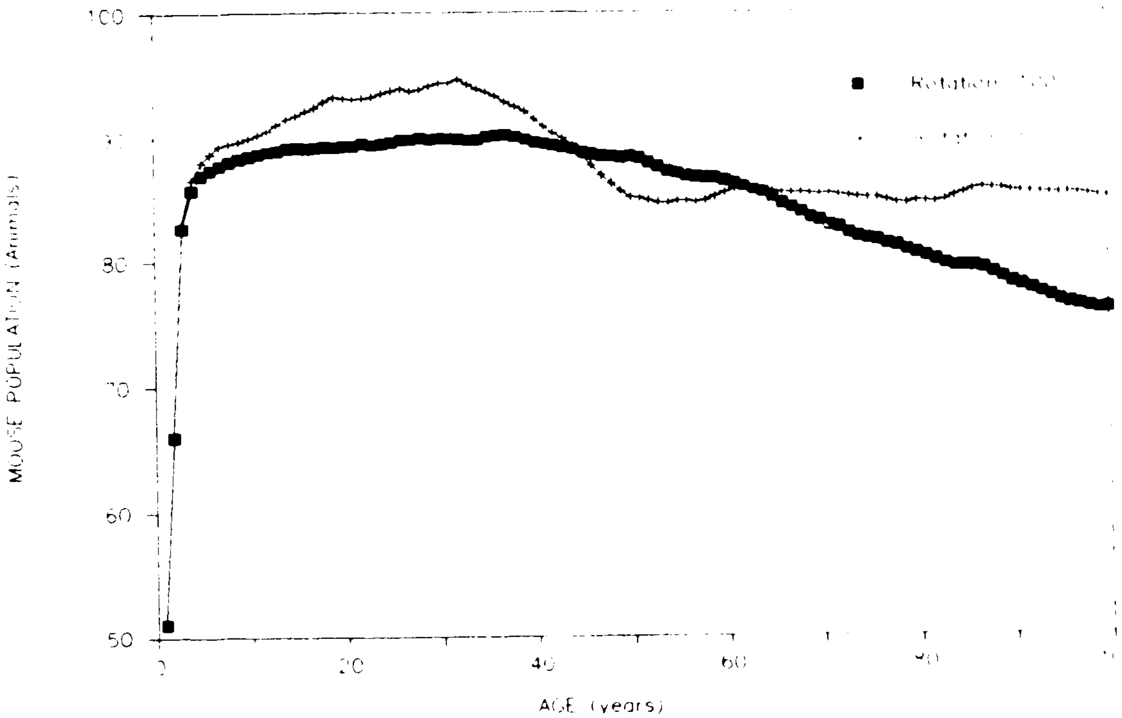


Figure 8.17 Simulated annual moose population changes under 50 and 100 year rotations

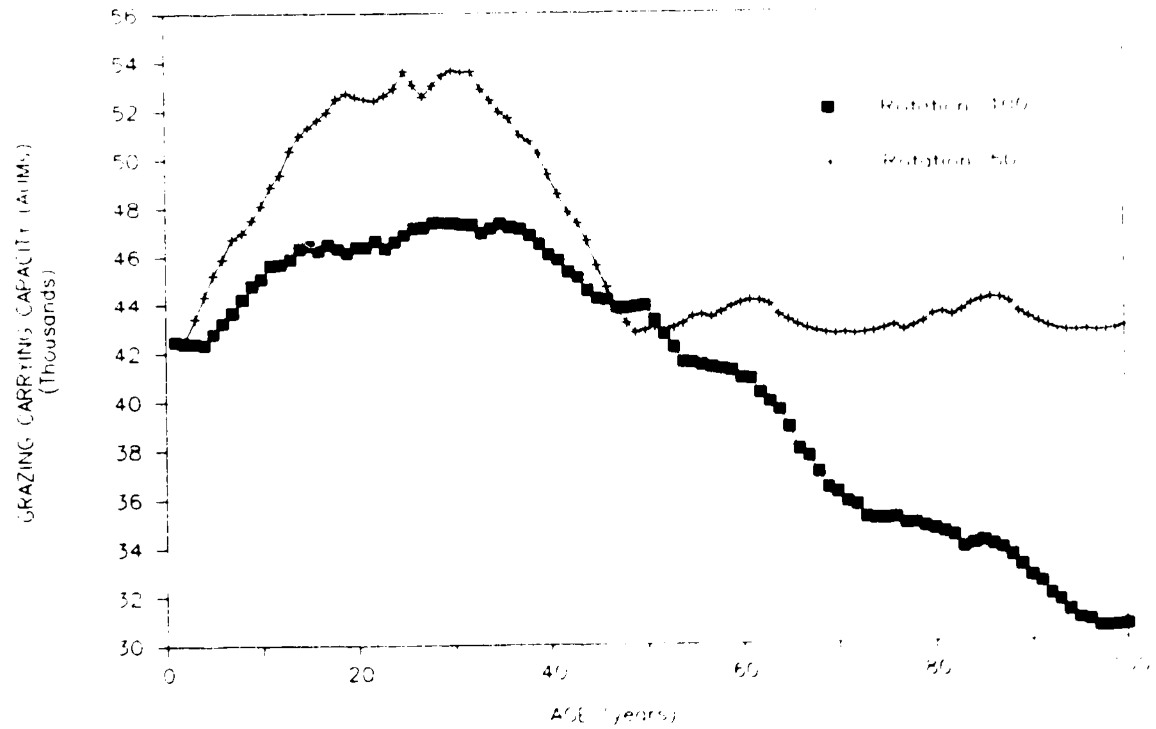


Figure 8.18 Simulated annual grazing carrying capacity changes under 50 and 100 year rotations

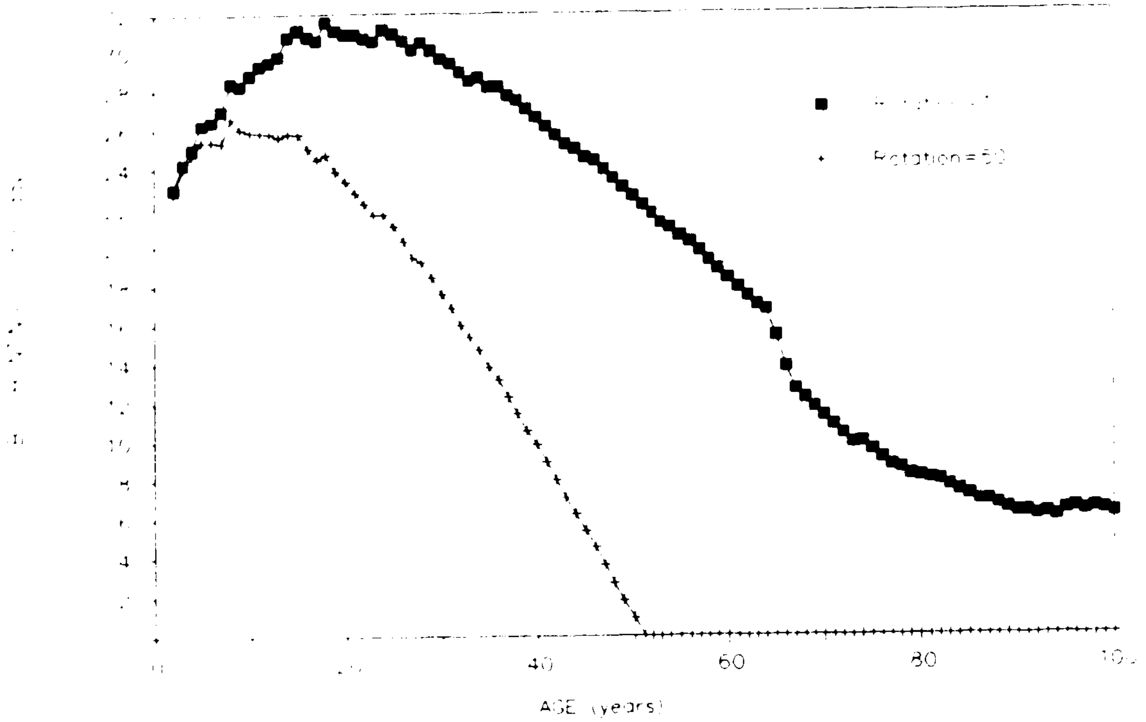


Figure 8.19 Simulated annual visitor-days changes of caribou hunting under 50 and 100 year rotations

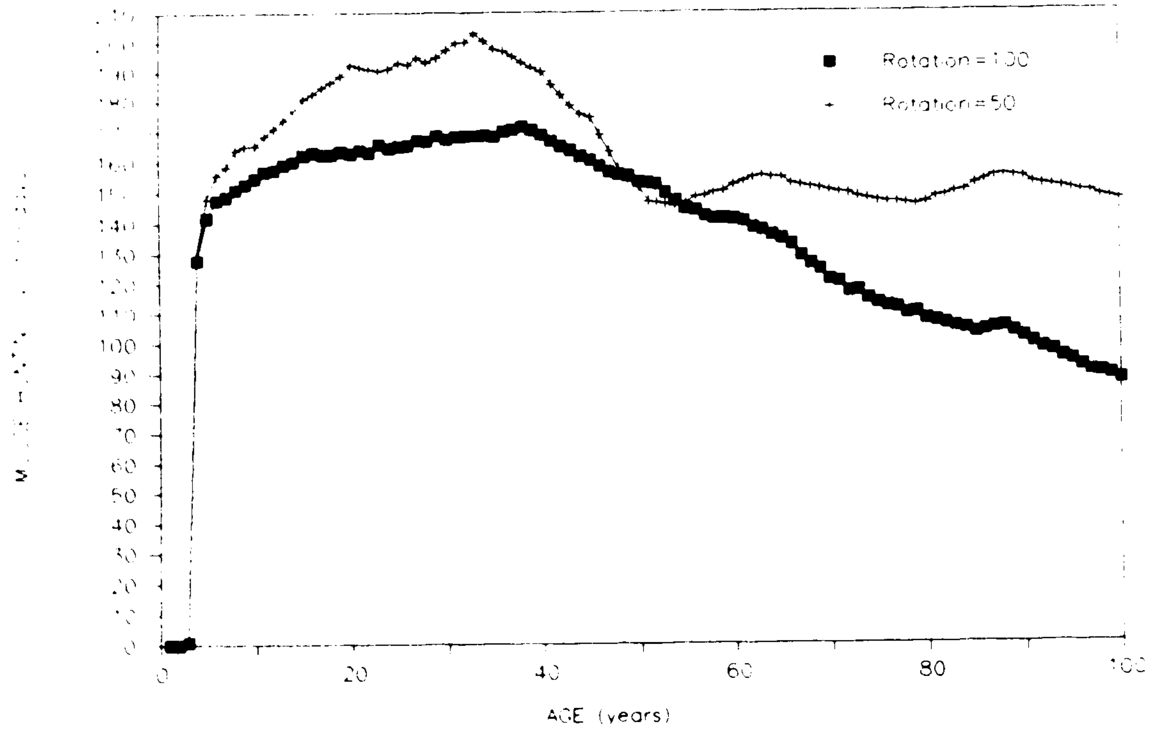


Figure 8.20 Simulated annual visitor-days changes of moose hunting under 50 and 100 year rotations

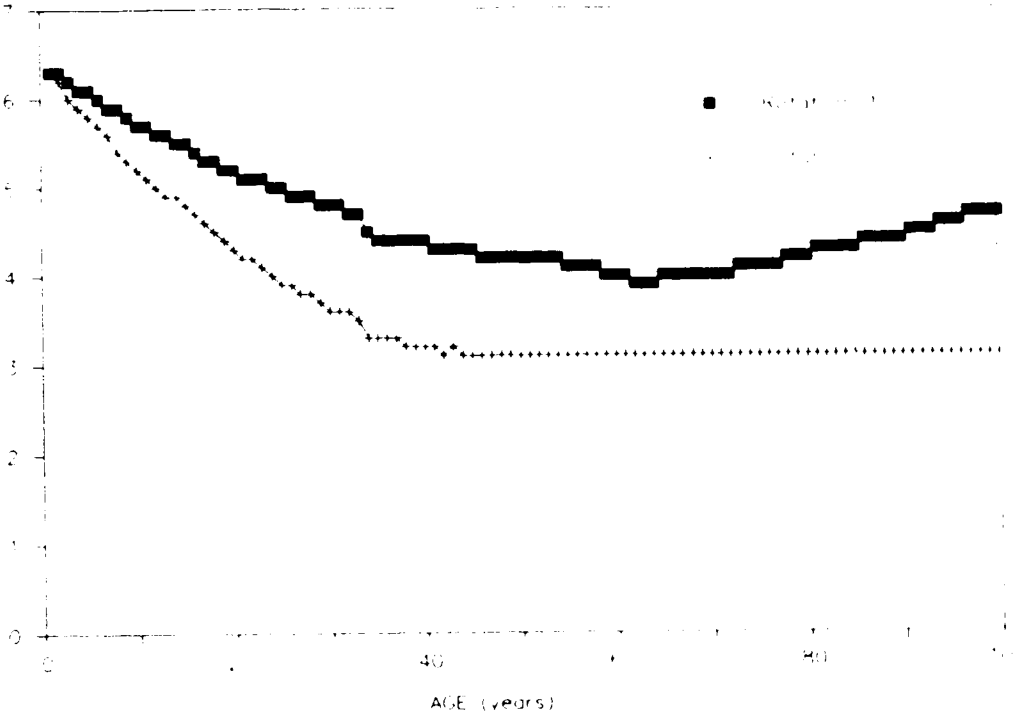


Figure 8.21 Simulated annual mean rating changes of natural scenic beauty under 50 and 100 year rotations

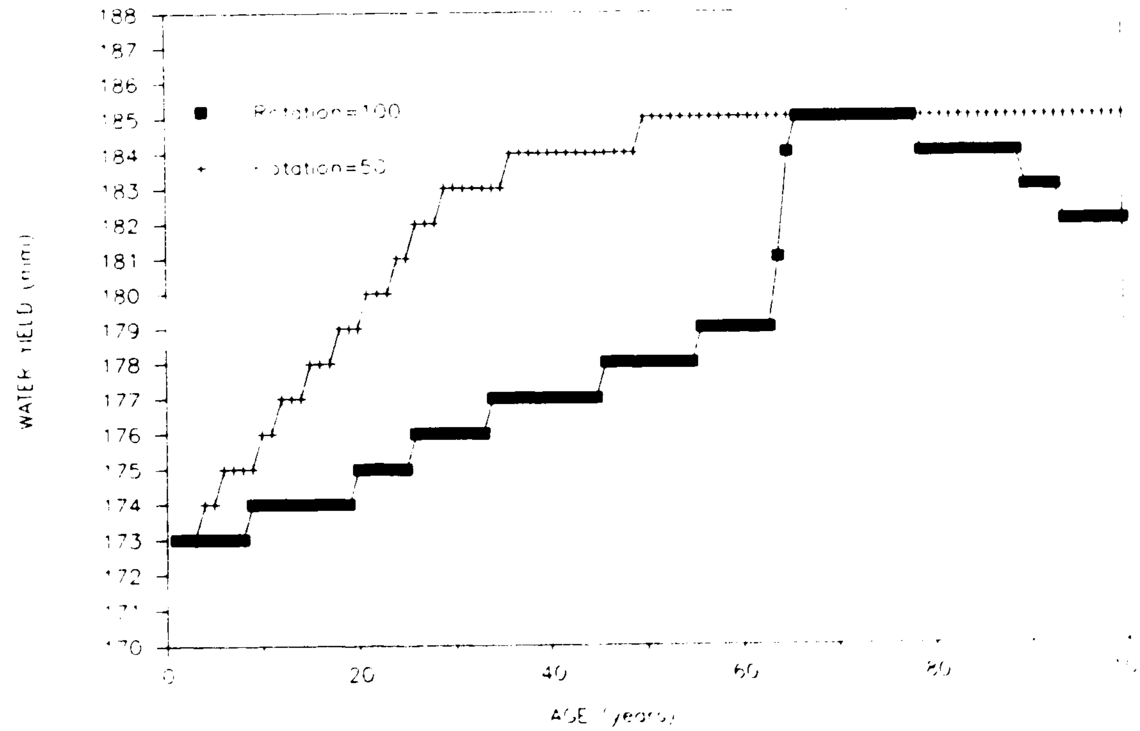


Figure 8.22 Simulated annual water yield changes under 50 and 100 year rotations

The previous application runs have been implemented only on 100 quarter sections because it required seventeen days to simulate the whole study area (E8 management unit). In order to illustrate the model's capability of applying to a large area, one simulation run under a 50 years rotation was executed. Figure 8.23 to 8.30 show the simulation results. By comparing with the annual timber harvests on a small area (Figure 8.14), the simulation results on a large area (Figure 8.23) indicate a smaller percentage variation. Otherwise, the figure showing the changes to the other wildland resources show trends similar to that shown for the smaller area.

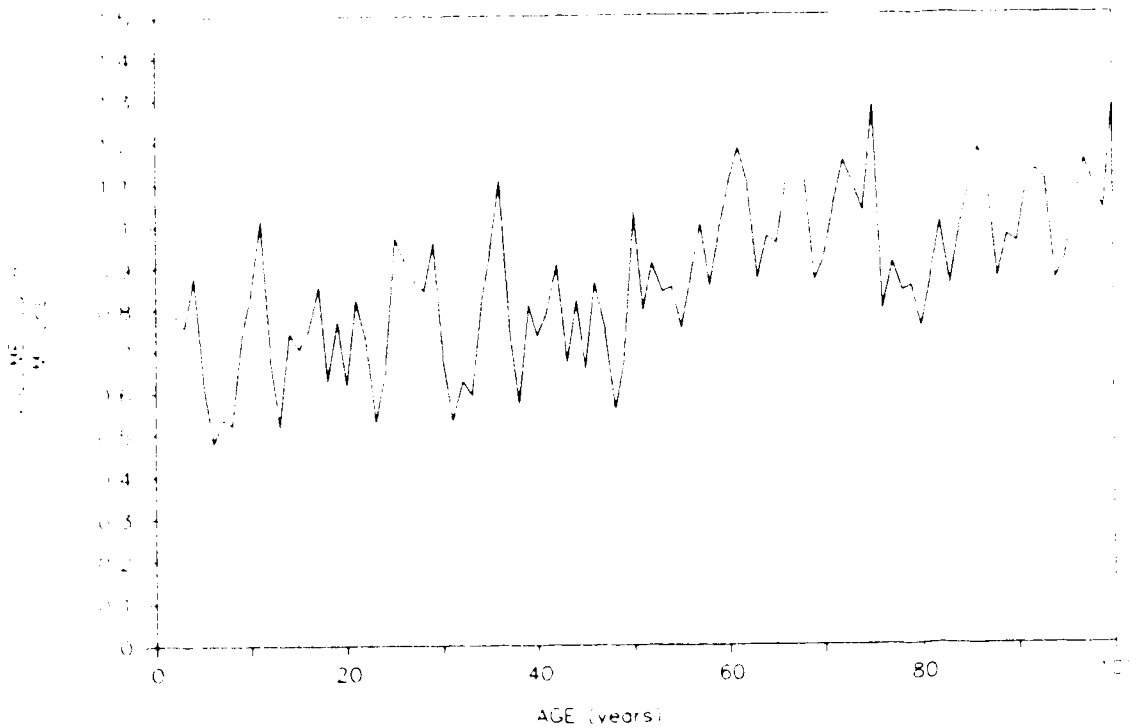


Figure 8.23 Simulated annual harvested volume of pine of E8 unit under 50 year rotation

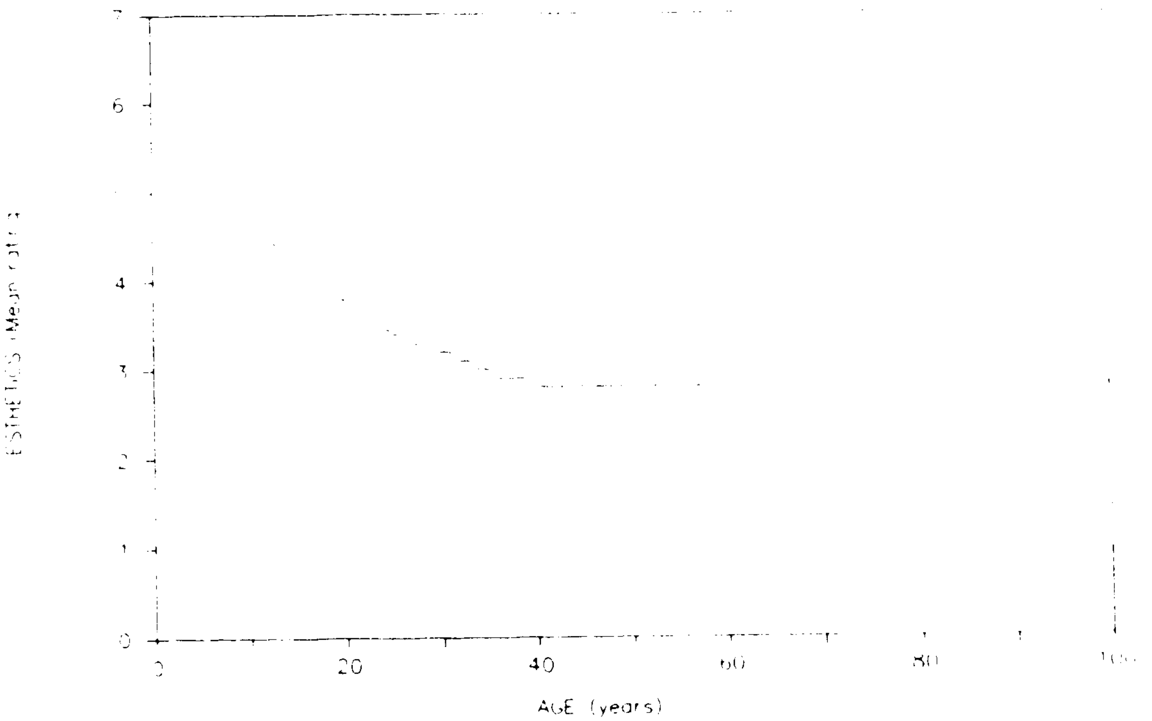


Figure 8.24 Simulated annual mean rating changes of natural scenic beauty of E8 unit under 50 year rotation

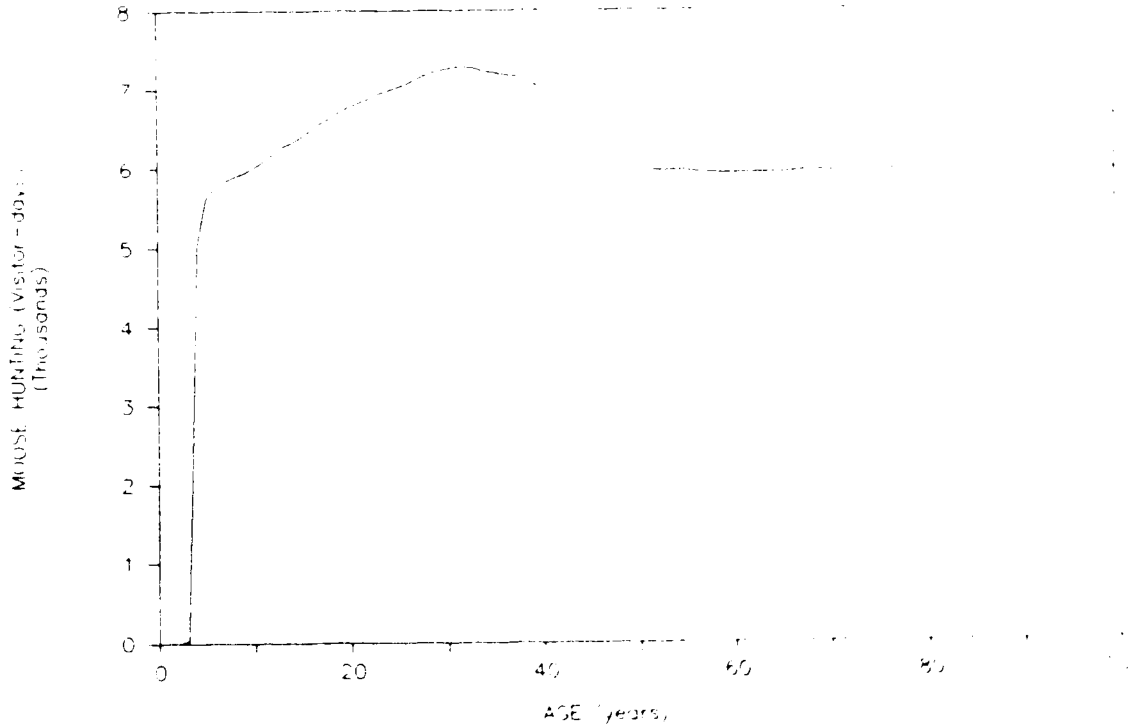


Figure 8.25 Simulated annual visitor-days changes of caribou hunting of E8 unit under 50 year rotation

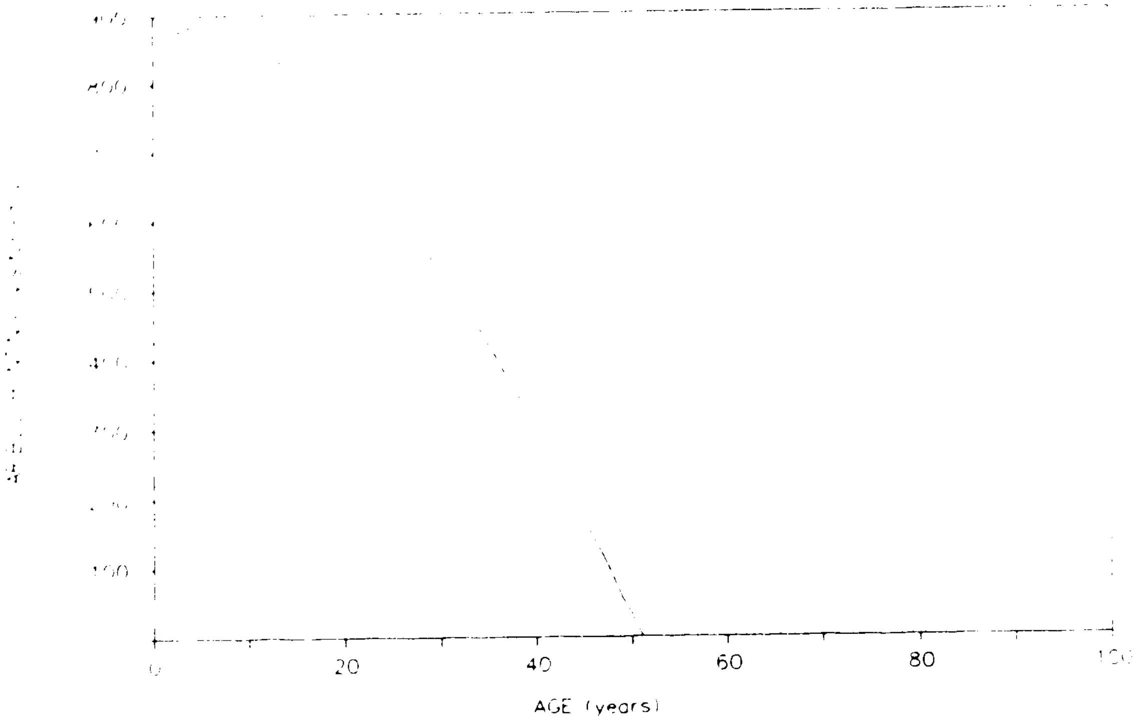


Figure 8.26 Simulated annual visitor-days changes of moose hunting of E8 unit under 50 year rotation

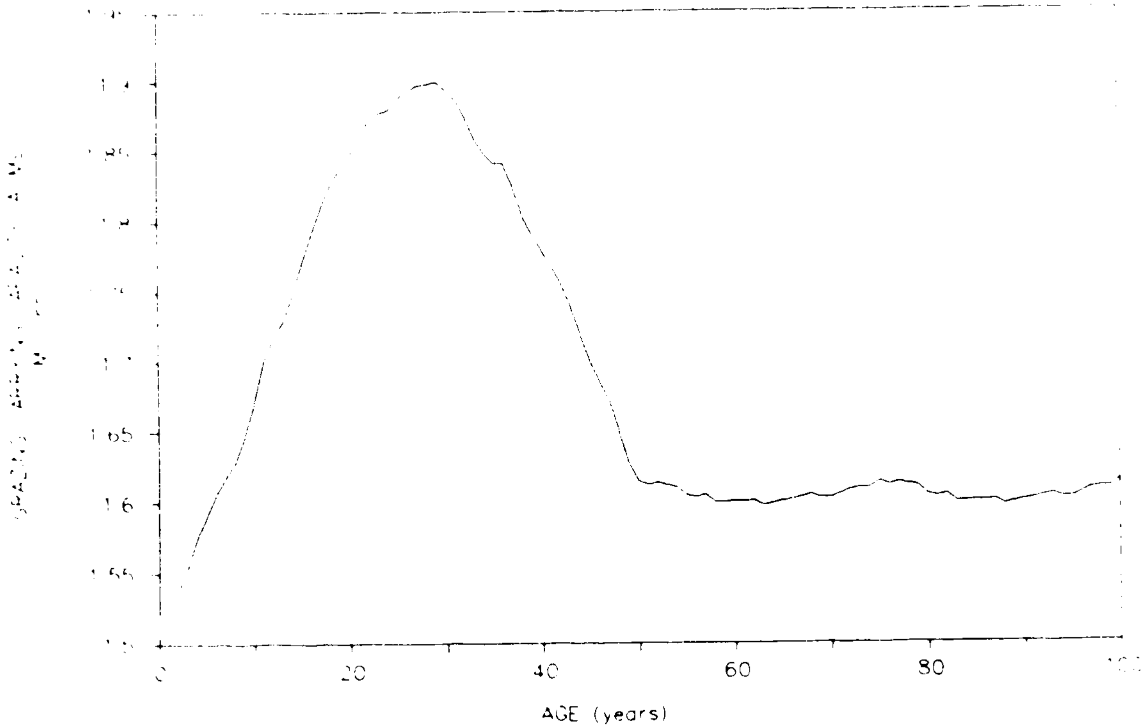


Figure 8.27 Simulated annual grazing carrying capacity changes of E8 unit under 50 year rotation

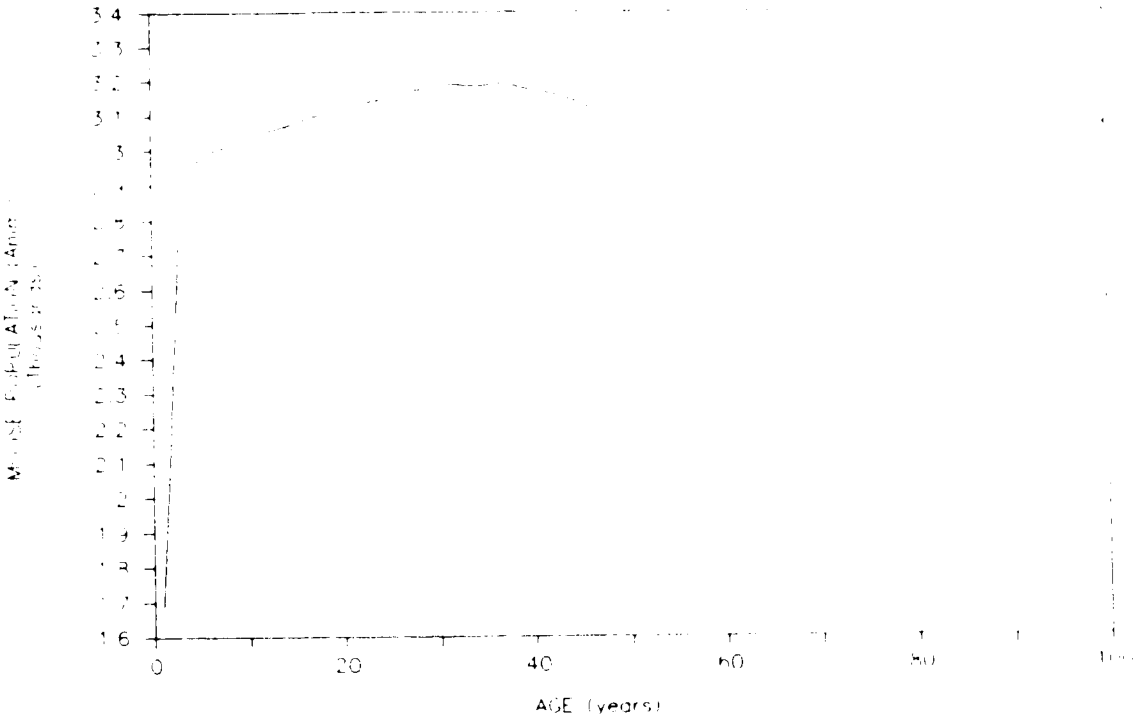


Figure 8.28 Simulated annual moose population changes of E8 unit under 50 year rotation

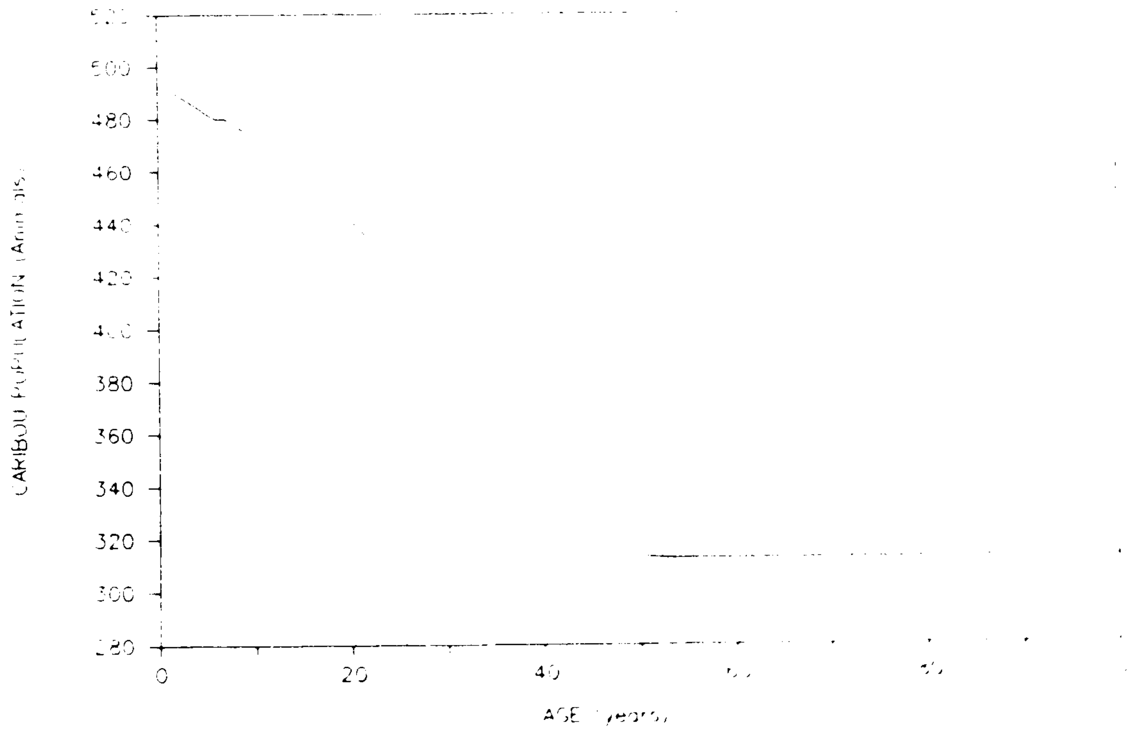


Figure 8.29 Simulated annual caribou population changes of E8 unit under 50 year rotation

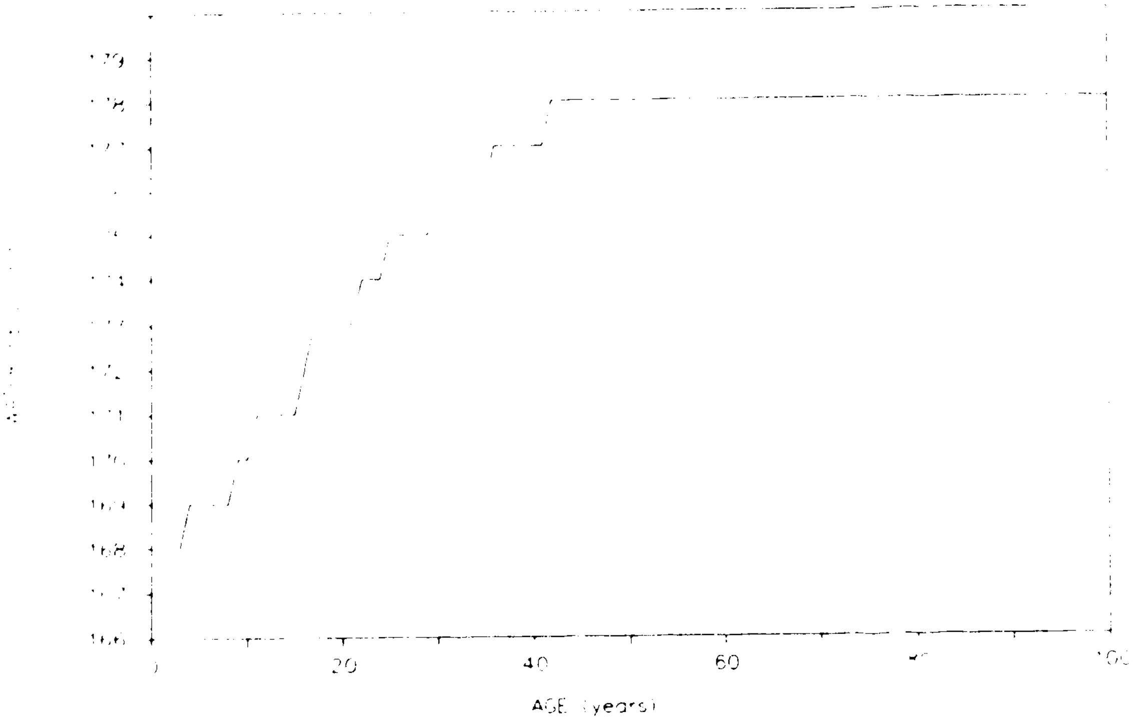


Figure 8.30 Simulated annual water yield changes of E8 unit under 50 year rotation

8.2 Evaluation of the Simulation Model Developed with dBASE IV

As a general principle, the simpler the technique employed, the more useful or generally applicable is the model (Bunnell, 1974). Many of the advantages of using a database to develop and implement an integrated resource planning simulation model revolve around this principle. The user-friendly nature of the dBASE IV, whereby successful development and implementation of a model using the dBASE IV program does not depend on an experienced computer programmer, is a major advantage.

The dBASE IV package contains a full-screen program editor (like a word processor) which facilitates rapid development of similar or exact copies of desired commands and formulas and provides outstanding flexibility. The dBASE IV language also includes almost all of features (functions, subroutines, arrays, loops, etc.) available from other general purpose programming languages like BASIC, FORTRAN, and PASCAL. But over those languages, it makes the data manipulation (input and output) easier and more understandable. The REPLACE command can update any values on a database file by their "record" number and "field" name. In addition, dBASE IV contains an Application Generator whereby one can create a program through menu choices without programming. The Application Generator is convenient and an easy way to develop a program when the user's needs are relatively straightforward. While it does not provide the flexibility of writing code, it also does not require much specialized knowledge. This feature may help foresters develop their own models.

Another advantage of using dBASE IV is that this software has a linkage that enables the user to interchange data to the widely used LOTUS 1-2-3 quite easily. This will allow the dBASE simulation model user to access the useful and convenient graphic functions in LOTUS.

This project and others (Silvert, 1983; Gray and Keith, 1986) have shown that the spreadsheet approach is a useful modeling tool. However, the amount of computer memory available can limit the modeler's ability to construct and run large models using spreadsheet packages. This is an annoying limitation because even though some planning models can be simple, they may require a large database (Gray and Keith, 1986) Different from the spreadsheet packages, dBASE IV does not load the whole database file into computer memory. Therefore, this package theoretically can handle a database as large as 255 fields and 1 billion records depending on the disk storage spaces available. For this simulation model, a database with 87 fields and 3257 records was established and it used about 2 megabytes of space on the hard disk. Thus, the package is capable of running a simulation model for a reasonably large planning area on a microcomputer.

This study also showed that dBASE IV's full-screen report design feature is very convenient and quick for the model user to summarize the simulation results and database files. The report format can be modified on the screen at will.

The type of microcomputer used will determine the simulation execution time required to run the model. The simulation program coded in dBASE language for this study required approximately seventeen days to run a

100-year simulation for all of 3257 quarter sections on an IBM PS/2 Model 60 computer. This is about 4-5 seconds for one quarter section for each time interval. From the computing efficiency point of view, this is too slow to run, and from the user point of view, the model may be impractical for the resource manager to use to obtain results for a large area. The reason for the slow processing could be caused by the amount of numerical calculation and by the data-reading and data-writing process. This limitation is not particularly formidable because dBASE package allows the use of assembler language programs which can speed up the calculations considerably. Moreover, the simulation time can be expected to decrease greatly as the capacities of microcomputer systems increase.

Alternatively, this kind of simulation can be implemented on mainframe computer by scarifying the user-friendly and popular characteristics of the microcomputer and its software. New parallel processing computers should be able to reduce the run time considerably because the same equations are applied to each record for each time step. Present computers require that the processing of each record be performed sequentially. If the calculations could be performed simultaneously for several records, computer execution time would be reduced dramatically.

9 CONCLUSIONS

This model shows that use of dBASE IV is an alternative language to a spreadsheet for simulation modeling integrated resource management problems. Its use does not require any advanced expertise in any particular programming language. This makes the tool particularly useful for managers to evaluate alternative strategies. dBASE IV is very easy to use because of its various menus. The conceptual model of the planning process is always visible. There is little abstractness in the model, which is not the case with many simulation models.

The simulation model, constructed by combining the five mathematical models, does offer a means of predicting long-term timber, recreation, grazing, wildlife, and watershed yields A comparison of the simulation results with available inventory (yield table) data indicates that the model is reasonable approximation of actual resource responses over longer time spans. Complete validation would require additional trials on other integrated resource planning areas and further refinement of the functional relationships.

The model is capable of simulating integrated resource planning process for a large area. The execution of the program on an IBM PS/2 Model 60 computer takes approximately seventeen days for 100-year-periods of a simulation run for the whole study area. By linking dBASE IV (database) with LOTUS 1-2-3 (spreadsheet), simulation results may be reported either in tabular or graphic forms.

The literature review of this project did not contain any references to an integrated numerical simulation model for multiple land use planning. There are a number of models available for predicting individual resource changes. Few models, however, combine two mathematical models to study the interactive influence of changing one resource product on another. Furthermore, these models are sensitive to fluctuations in conditions on a site to site basis. The simulation model, used in this research, is for analyzing the effects of integrated resource use on forestland, and is not site specified in terms of mathematical model structures. The structure of the simulation model does not restrict the number of areas to be evaluated but computer capacity will provide a potential limit.

The results of this research were limited by insufficient disk input-output speed and by limited memory capacity of the computer used and the limited number of validated mathematical functions used for simulating the resource changes that were available to the modeler. The model is currently limited in a number of respects. First, the current mathematical models are not validated representations of the biological behavior. Second, economic analysis is not incorporated. Finally, thinning and harvesting alternatives are the only stand-level treatments considered.

A fundamental knowledge of dynamic joint production possibilities is essential to evaluating alternatives and tradeoffs in integrated resource planning. This basic methodology, that of developing individual mathematical models and of the combining of them into a computerized simulation, offers a means of analyzing complex multi-resource relationships and studying the effects of different managerial operations. By providing these capabilities, the simulation procedure acts as a gaming tool and aid for the manager in testing and studying the effects of long-term management alternatives. It provides a useful tool in the development of a systems approach to integrated resource planning.

10 RECOMMENDATIONS FOR FUTURE WORK

The integrated resource planning simulation model developed in this project is only a prototype decision making tool that may help resource managers and foresters bridge the gap between strategic planning and implementation. It can be useful to managers in implementing integrated resource planning. However, much remains to be done to make the model more accurate and acceptable. The following are recommended for future work in the same area:

a. Economic relationships should be included in the simulation model. By employing economic values, the simulation model can be used to select rotation length based on maximizing forest use benefits.

b. The calculation parts of the programs can be programmed in assembly language to reduce the computing time. Since the program works with the computer's microprocessor, assembly language is closer to the actual hardware of the computer than dBASE. Assembly language programs can be run within dBASE IV. Another alternative is to use the computer which has a higher speed and more capacity.

c. To use the model as a simulator of integrated resource planning processes, "better" mathematical models for individual natural resource dynamics are needed to improve the validity of the simulation model. Moreover, the natural resources such as fisheries and minerals should be included in the model if suitable mathematical models are available.

d. The outputs of the simulation model, if possible, may be entered into a geographic information system (GIS) to show the spatial changes of natural resources over time under different management decisions.

e. The simulation model requires further validation with field data

REFERENCES

Alberta Forest Service. 1985. Alberta Phase 3 Forest Inventory: Yield Tables for Unmanaged Stands. Department of Energy and Natural Resources, Edmonton, Canada.

Alberta Forest Service. 1988a. Alberta's Forests. Department of Forestry, Lands and Wildlife, Edmonton, Canada.

Alberta Forest Service. 1988b. Water Yield Potential from Vegetation Manipulation in Drainages in the Berland I.R.P. Department of Forestry, Lands and Wildlife, Edmonton, Canada.

Arthur, L.M and R.S. Boster. 1976. Measuring Scenic Beauty: a Selected Annoted Bibliography. USDA For. Serv. Gen. Tech. Rep. RM-25, Rocky Mt. For. and Range Exp. Stn., Fort Collins, CO.

Ashton-Tate. 1988. dBASE IV Manual, Advanced topics. Ashton-Tate Corporation, Torrance, CA.

Avery, T.E. 1975. Natural Resources Measurements. McGraw-Hill Book Company. New York, NY.

Bare, B.B. and G.F. Schreuder. 1975. Environmental impact assessment modeling and land-use planning. *In* System Analysis and Forest Resource Management. Soc. Amer. For., Bethesda, MD.

Bare, B.B. and R.C. Field. 1987. An evaluation of FORPLAN from an operations research perspective. Proceedings of a FORPLAN Symposium, November 4-6, 1986, Denver, CO.

Bell, E.F. 1976. Goal Programming for Land Use Planning. USDA For. Serv. Gen. Tech. Rep. PNW--53, Portland, OR.

Benson, R.E. and J.R. Ullrich. 1981. Visual Impacts of Forest Management Activities: Findings on Public Preferences. USDA For. Serv. Res. Pap. INT-262 Intermt. For. and Range Exp. Stn., Ogden, Utah.

Bottoms, K.E. and E.T. Bartlett. 1975. Resource allocation through goal programming. Jour. of Range Manage. 28:442-447.

Brown, T.C. 1981. Tradeoff Analysis in Local Land Management Planning. USDA For. Serv. Gen. Tech. Rep. RM--82, Fort Collins, CO.

Bunnell, F.L. 1974. Computer simulation of forest-wildlife relations.

In H.C. Black (ed.), Wildlife and Forest Management in the Pacific Northwest. Corvallis: School of Forestry, Oregon State University, pp. 39-50.

Buongiorno, J. and J.K. Gilless. 1987. Forest Management and Economics. Macmillan Publishing Company. New York, NY.

Camp, H.W. 1973. Land-use planning: what it is and what it is not. *In* Proceedings of 1973 National Convention of Society of American Foresters. Portland, OR.

Chappelle, D. 1988. Grande Catch Forest Operations, Alberta, Canada Personal communication.

Chappelle, D.E. 1977. Linear programming for forestry planning. *In* Forestry and Long-Range Planning. F.J. Convery and C.W. Ralston Eds., Duke University, Durham, NC., pp. 129-163.

Chappelie, D.E., M. Mang, and R.C. Miley. 1976. Evaluation of Timber RAM as a forest management planning model. J. For. 74:299-293.

Charnes, A. and W.W. Cooper. 1961. Management Models and Industrial Applications of Linear Programming. Vol 1. John Wiley & Sons, New York, NY.

Cliff, E.P. 1965. Forest Patterns -- Beauty and Use. USDA Forest Service P.A. 679.

Clutter, J.L., J.C. Fortson, L.V. Pienaar, G.H. Brister, R.L. Bailey. 1983. Timber Management: a Quantitative Approach. John Wiley & Sons, Inc., New York, NY.

Dane, C.W., N.C., Meador, and J.B. White. 1977. Goal programming in land-use planning. J. For. 75:325-329.

Daniel, T. and R.S. Boster. 1976. Measuring Landscape Esthetics: the Scenic Beauty Estimation Method. USDA For. Serv. Res. Pap. RM-167, Rocky Mt. For. and Range Exp. Stn., Fort Collins, CO.

Davis, K.P. 1976. Land Use. McGraw-Hill Book Company, New York, NY.

Department of Forestry, Land and Wildlife. 1987. Berland Sub-Regional Integrated Resource Plan (Draft). Edmonton, Alberta.

Douglass, R.W. 1969. Forest Recreation. Pergamon Press Ltd., Oxford, London.

Duerr, W.A., D.E. Teeguarden, N.B. Christiansen, and S. Guttenberg. 1982. Forest Resource Management. O.S.U. Book Stores, INC., Corvallis, OR.

Fardoe, J.S. 1984. Integrated Resource Planning in Alberta. Alberta Energy and Natural Resources, Edmonton, Canada.

Field, D.B 1973. Goal programming for forest management. For. Sci. 19:125-135.

France, J. and J.H M. Thornley. 1984. Mathematical Models in Agriculture. Butterworth & Co (Publishers) Ltd. England.

Frenkiel, F N. and D.W. Goodall. 1978. Simulation Modeling of Environmental Problems. John Wiley and Sons, Chichester, England.

Fries, J. (Ed.). 1974. Growth models for tree and stand simulation. Proceedings of IUFRO Meetings, S4.01-4. Royal College of Forestry, Stockholm.

Fries, J., H.E. Burkhart, and T A. Max (Eds.). 1978. Growth models for long term forecasting of timber yields. Proceedings of IUFRO Meeting, Div.4., Virginia Polytechnic Inst. and State Univ., Blacksburg, VA.

Garfinkel, D. 1962. Digital computer simulation of ecological system. Nature 194:856-57.

Garner, C J. 1978. Simulation: a Decision Making Aid for Managers. Technical Report No. TR-30, Forest Engineering Research Institute of Canada.

Gause, G.F. 1934. The Struggle for Existence. Hafner, New York, NY.

Golding, D.L. and R.H. Swanson. 1978. Snow accumulation and melt in small forest openings in Alberta. Canadian Journal of Forest Research, Vol. 8, No. 4, pp. 380-388.

Government of Alberta. 1984. A Policy for Resource Management of the Eastern Slopes. Department of Public Lands & Wildlife, Edmonton, Canada.

Grant, W.E. 1986. System Analysis and Simulation in Wildlife and Fisheries Sciences. John Wiley & Sons, Inc. New York, NY.

Gray, P.A. 1986. Microcomputer and minicomputer technology for wildlife habitat management in the Northwest Territories, Canada Committee on

Ecological Land Classification Newsletter, 14, pp. 1-2.

Gray, P.A. and J. Keith. 1986. The application of a spreadsheet formatting procedure in the development and implementation of a habitat evaluation model for the Northwest Territories. *In* Perspectives on Land Modeling. Polyscience Publications Inc. Montreal, Canada.

Grelen, H.E. 1978. Forest grazing in the south, Journal of Range Management, 31:244-249.

Hammitt, W.E. and D.N. Cole. 1987. Wildland Recreation. John Wiley & Sons, Inc. New York, NY.

Hanson, W.R. 1975. Current Range Management on Forest Lands in Alberta. Alberta Forest Service Report, Edmonton, Canada.

Hewlett, J.D. 1982. Principles of Forest Hydrology. The University of Georgia Press, Athens, GA.

Hoganson, H.M. and D.W. Rose. 1984. A simulation approach for optimal timber management scheduling. For. Sci. 30:220-238.

Holling, C.S. 1978. Adaptive Environmental Assessment and Management. International Institute for Applied Systems Analysis. John Wiley and Sons, Inc., Chichester, England.

Husch, B., C.I. Miller, and T.W. Beers. 1982. Forest Mensuration. John Wiley & Sons, Inc., New York, NY.

Ingles, D.M. 1985. What Every Engineer Should Know (15): Computer Modeling and simulation. Marcel Dekker, Inc., New York, NY.

Johnson, K.N., D.B. Jones, and B.M. Kent. 1980. Forest Planning Model (FORPLAN). User's Guide and Operations Manual. 251p.

Johnstone, W.D. 1981. Precommercial Thinning Speeds Growth and Development of Lodgepole Pine: 25-year Results. Can. Dept. Environ., Can. For. Serv., North. For. Res. Cent., Edmonton, Alta. Inf. Rep. NOR-X-237.

Johnstone, W.D. 1982. Heavy Thinning Accelerates Growth of 77-year-old Lodgepole Pine. Can. Dept. Environ., Can. For. Serv., North. For. Res. Cent., Edmonton, Alta. For Manag. Note 16.

Joyce, L.A. 1986. Regional forage model. *In* Perspectives on Land Modeling. Polyscience Publications Inc., Montreal, Canada.

Joyce, L.A., B. Mckinnon, J.G. Hof, and T.W. Hoekstra. 1983. Analysis of Multiresource Production for National Assessments and Appraisals. USDA Forest Service General Technical Report RM 101. Rocky Mountain Forest and Range Experiment Station, Fort Collins, CO.

Joyce, L.A. and R.L. Baker. 1987. Forest overstory-understory relationships in Alabama forests, Forest Ecology and Management, 17:49-59.

Joyce, P.M. 1981. Forest management and land use planning. Proceedings of the XVII IUFRO world congress. Tokyo, Japan.

Leaf, C.F. and B.E. Brink. 1973. Hydrologic Simulation Model of Colorado Subalpine Forest. USDA Forest Service, Pap. RM 107, Rocky Mtn. For. and Rg. Experiment Stn., Ft. Collins, CO.

Leak, W.B. 1969a. Sapling stand development: a compound exponential process. Forest Science 16:177-180.

Leak, W.B. 1969b. Stocking of northern hardwood regeneration based on exponential dropout r . For. Chron. 45(5):1-4.

Leary, R.A. 1970. System identification principles in studies of forest dynamics. USDA Forest Service. Research Paper NC-45.

Leary, R.A. 1973. Nonlinear functional equation models of forest dynamics. *In* Forest Modeling and Inventory. A.R. Ek, J.W. Balsiger and L.C. Promnitz Eds., Department of Forestry, University of Wisconsin-Madison, WI., pp. 20-36.

Lotka, R.J. 1924. Elements of Mathematical Biology. (Reprinted. Dover, New York (1956)).

Marczyk, J.S. 1985. Integrated Resource Planning in Alberta. Alberta Energy and Natural Resource. Edmonton, Canada.

Marlow, J.A. 1985. Land use planning for forest harvesting and environmental concerns. Forest Industry Lecture Series No. 16. Edmonton, Canada.

Myers, C.A. 1973. Simulating Changes in Even-aged Timber Stands. USDA Forest Service. Research Paper

Odum, E.P. 1976. International biol program:Was it worth the cost and effort? Science 193:866-868.

Odum, H.T. 1960. Ecological potential and analogue circuits for the ecosystem. American Scientist 48:1-8.

Odum, H.T. 1971. Environment, Power, and Society. Wiley-Interscience, New York, NY.

Olson, J.S. 1963. Analog computer model for movement of nuclides. *In* Radioecology. Reinhold, New York, NY.

Phillips, D.T., A. Rowindran, and J.J. Solberg. 1976. Operations Research: Principles and Practice. John Wiley & Sons, Inc., New York, NY.

Pritsker, A.B. 1984. Introduction to Simulation and SLAM II. John Wiley & Sons, Inc., New York, NY.

Reed, K.L. and S.G. Clark. 1979. Succession Simulator: A Coniferous Forest Simulator. Model Documentation. Coniferous Forest Biome, University of Washington, Seattle. Bulletin 11.

Rose, D.W. 1984. A critical review of mathematical programming in forest resource planning. Proceedings of IUFRO Symposium on Forest Management Planning and Managerial Economics. University of Tokyo, Tokyo.

Rustagi, K.P. 1976. Forest Management Planning for Timber Production: a Goal Programming Approach. Yale Sch For and Environ Stu Bull 89. Yale Univ., New Haven., Conn.

Schuler, A.T., H.H. Webster, and J.C Meadows. 1977. Goal programming in forest management. Jour. of For. 75:320-324.

Shechter, M. and R.C. Lucas. 1978. Simulation of Recreational Use for Park and Wilderness Management. Resources for the Future, Inc., Washington, D.C.

Silvert, W. 1983. Teaching ecological modeling with electronic spreadsheets. Applied Mathematical Notes of the Canadian Mathematical Society, 8, pp. 8-13.

Smith, D.M 1986. The Practice of Silviculture. John Wiley & Sons. New York, NY.

Spofford, W.O. 1975. Ecological modeling in a resource management framework: An introduction. p. 13-48. *In* Ecological modeling in a resource management framework. C.S. Russell, editor. Resources for the Future, Washington, D.C.

Stoddard, C.H. and G.M. Stoddard. 1987. Essentials of Forestry Practice. John Wiley & Sons, Inc. New York, NY.

Sweda, T. 1984. Growth curves as generatrix for improved forest mensuration and management. Proceedings of IUFRO Symposium on Forest Management Planning and Managerial Economics. University of Tokyo, Tokyo.

Taha, H.A. 1976. Operations Research. Macmillan Publishing CO., INC. New York, NY.

Troendle, C.A. and C.F. Leaf. 1980. Hydrology. *In* An Approach to Water Resources Evaluation of No-Point Silvicultural Sources. EPA 60018-80-012, Environmental Resources Laboratory, Athens, GA.

USDA, Forest Service. 1980. Forestland grazing. Atlanta: Forestry Report SA-FR 10. Forest Service State and Private Forestry Southeastern Area.

Van Dyne, G.M. and Z. Abramsky. 1975. Agricultural systems models and modeling: an overview. p. 23-106. *In* The study of agricultural systems. G.E. Dalton, editor. Applied Science Pub., Ltd. Berking, Essex, England.

Walker, B.A. 1982. The Role of Quantitative Modeling in Forest Management Planning. Unpublished thesis. University of Minnesota.

Watts, S.B. 1983. Forestry Handbook. D.W. Friesen & Sons Ltd. Cloverdale, B.C.

Weisz, R. 1984. Land management planning research and development. Proceedings of IUFRO Symposium on Forest Management Planning and Managerial Economics. University of Tokyo, Tokyo.

Weize, R. 1988. Multilevel Planning in a Spreadsheet Environment. USDA General Technical Report RM-161.

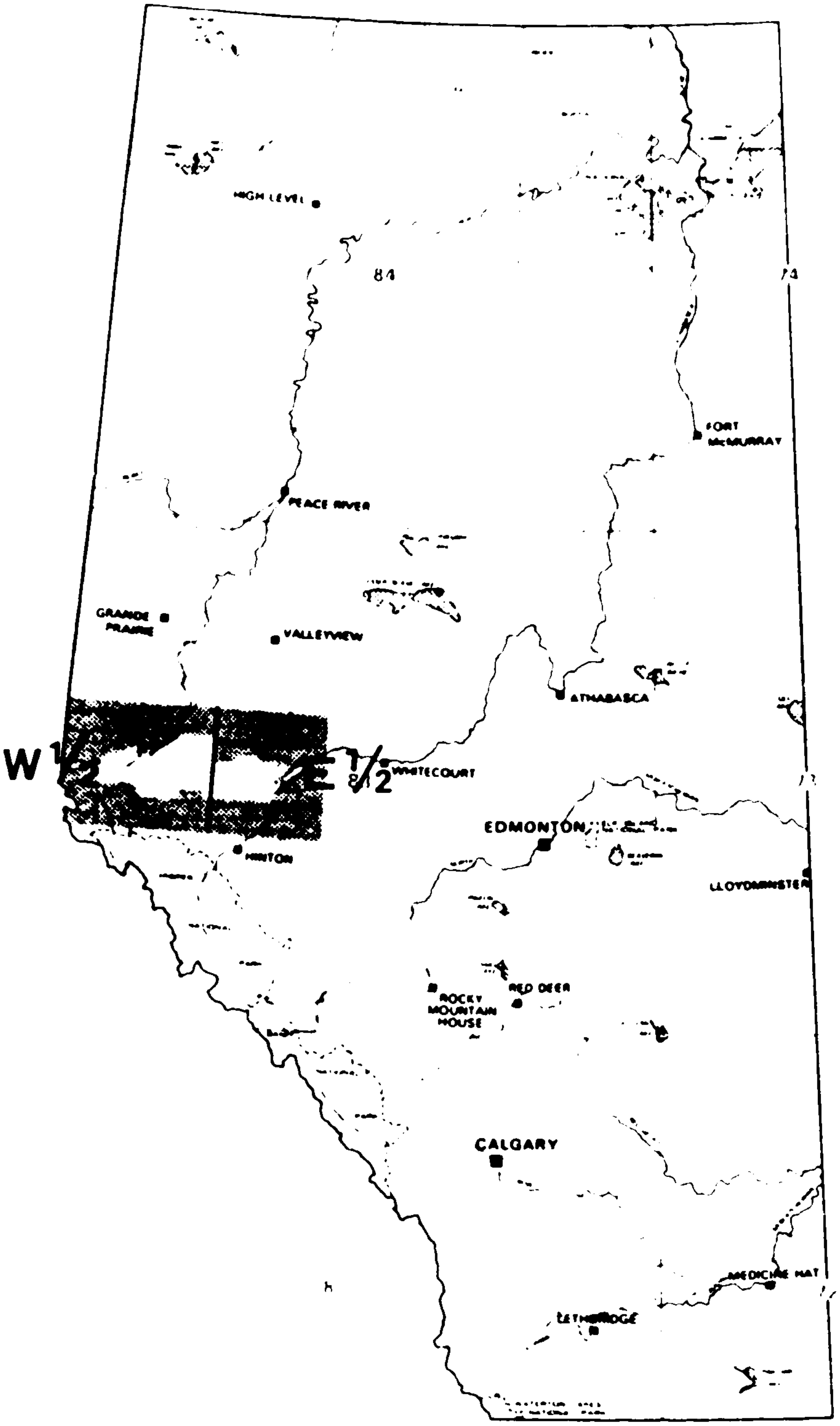
West, P.W. 1983. Comparison of stand density measures in even-aged regrowth eucalyptus forest of southern Tasmania. Can. J. For. Res. 13:22-31.

Wiegert, R.G. 1975. Simulation models of ecosystems. Annual Review of Ecology and Systematics 6:311-338.

Wolters, G.L., A. Martin and H.A. Pearson. 1982. Forage response to overstory reduction on loblolly-shortleaf pine-hardwood forest range. Journal of Range Management, 35:443-446.

Yang, Y.C. 1988. Generalized logistic mortality model for natural forests and plantations in Taiwan. Proceedings of IUFRO Forest Simulation Systems Conference, Berkeley, CA.

APPENDIX A: THE LOCATION MAP OF STUDY AREA



APPENDIX B.1: FLOWCHART OF THE SIMULATION MODEL

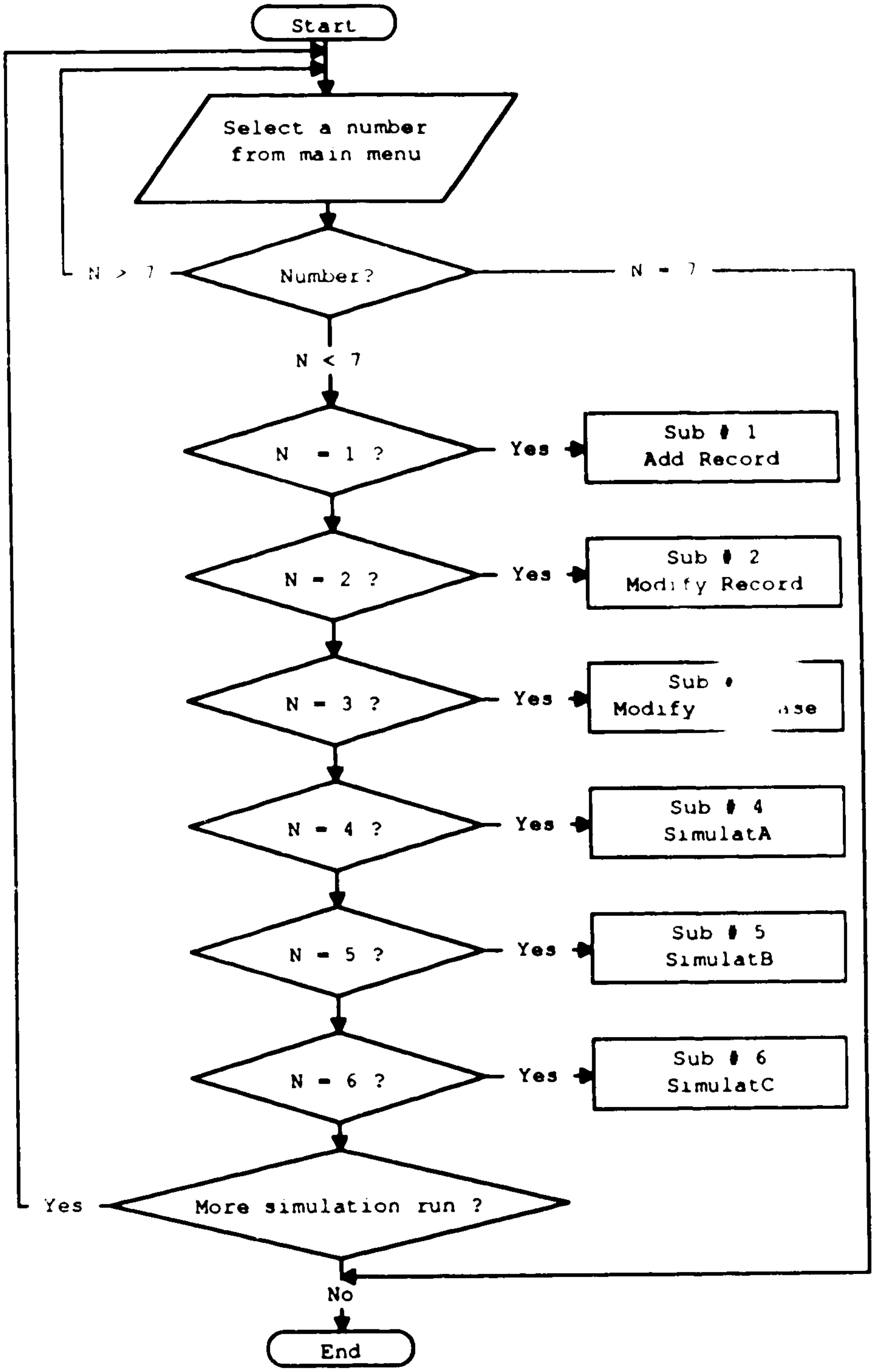


Figure B.1 Flowchart of the main program.

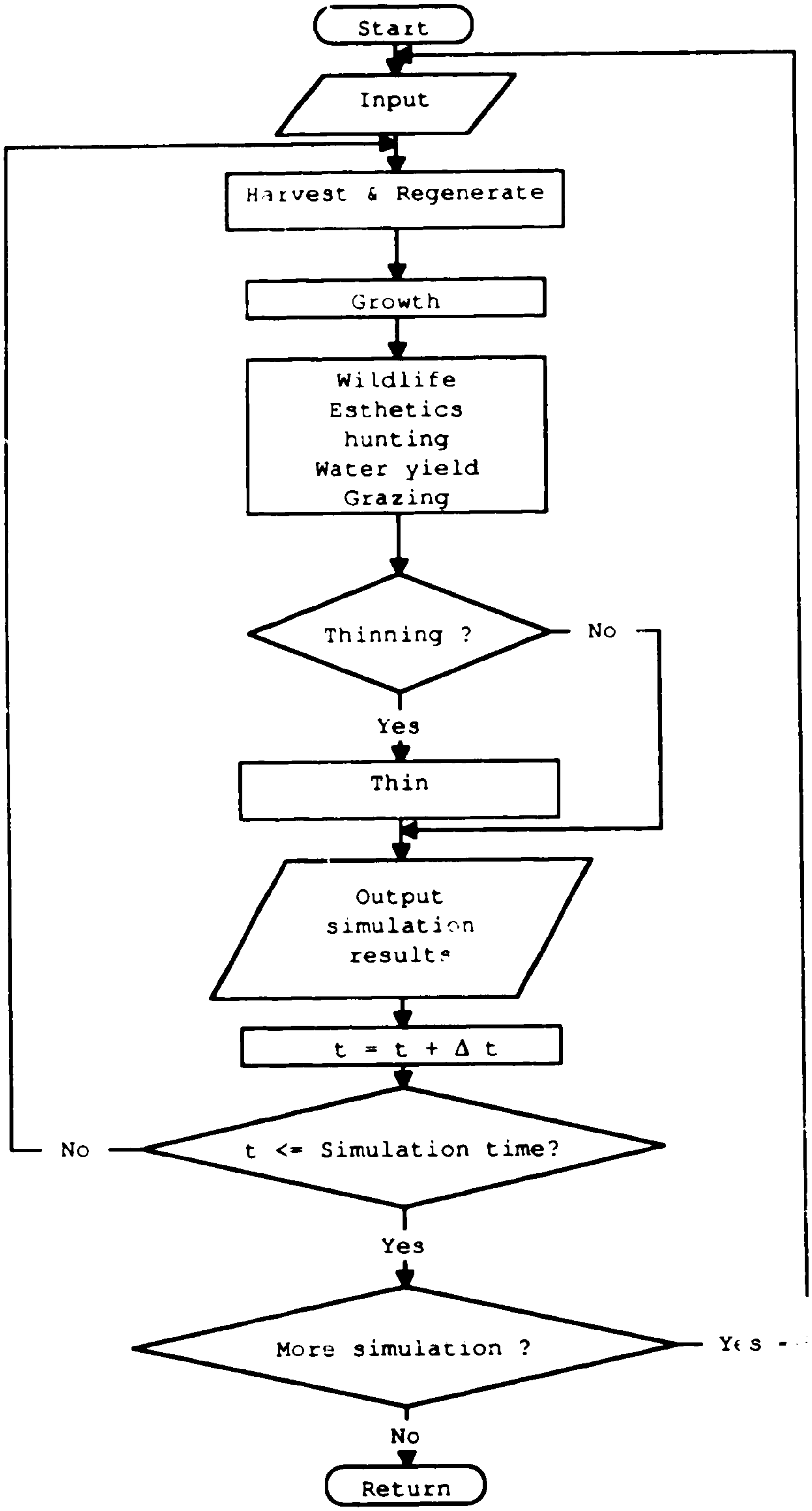


Figure B.2 Flowchart of the subroutine SIMULATA.

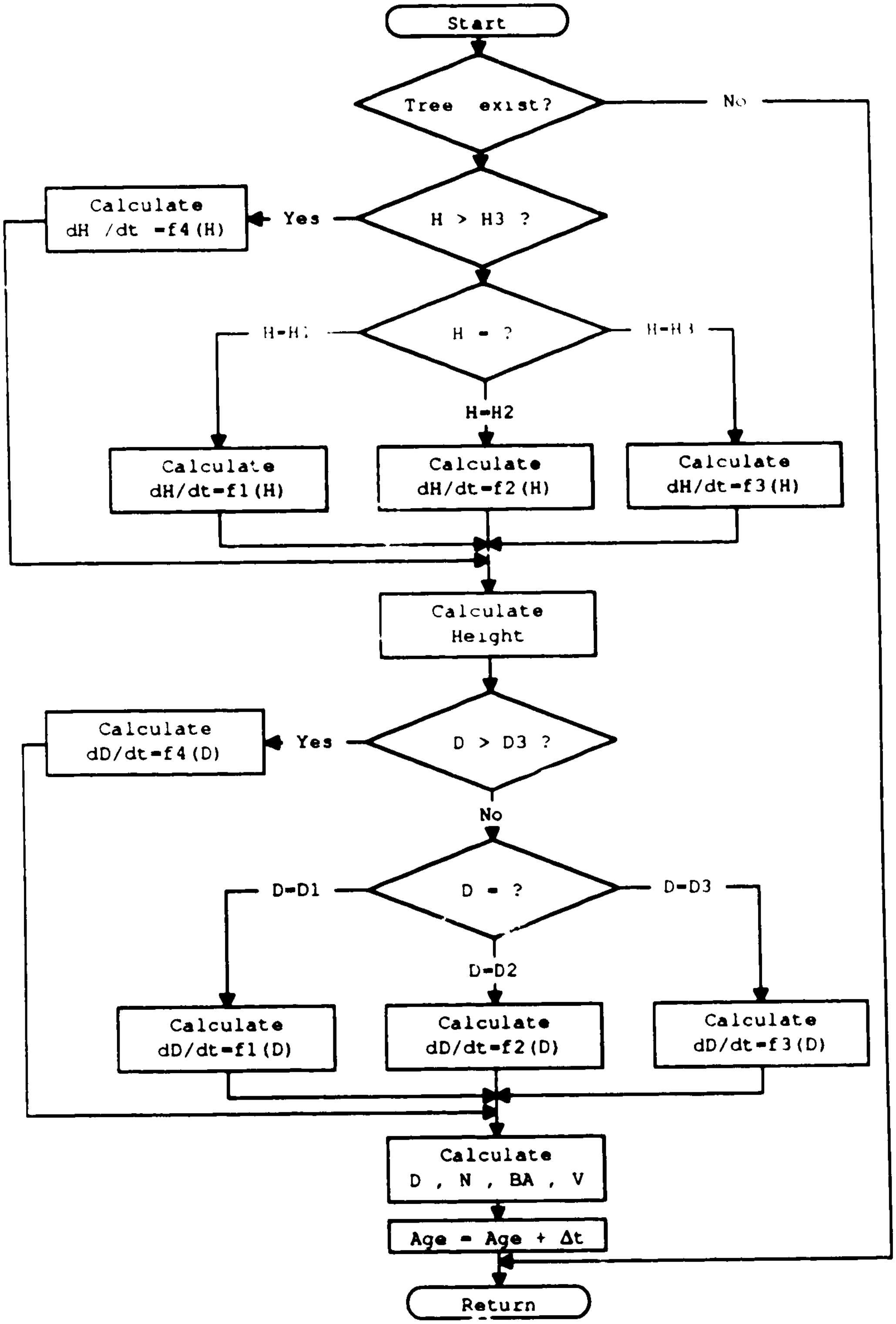


Figure B.3 Flowchart of the subroutine GROWTH.

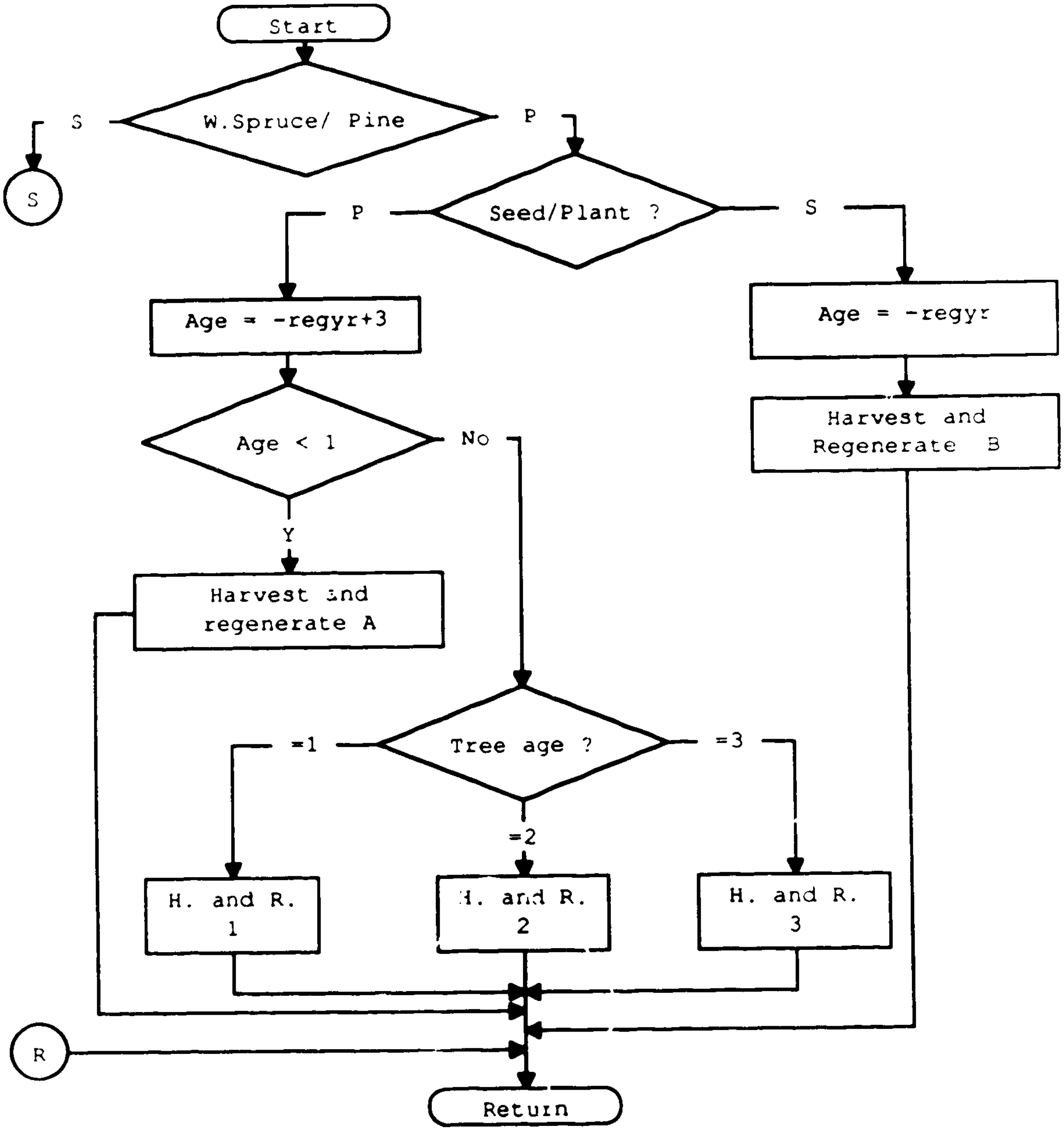


Figure B.4 Flowchart of the subroutine HARREG.

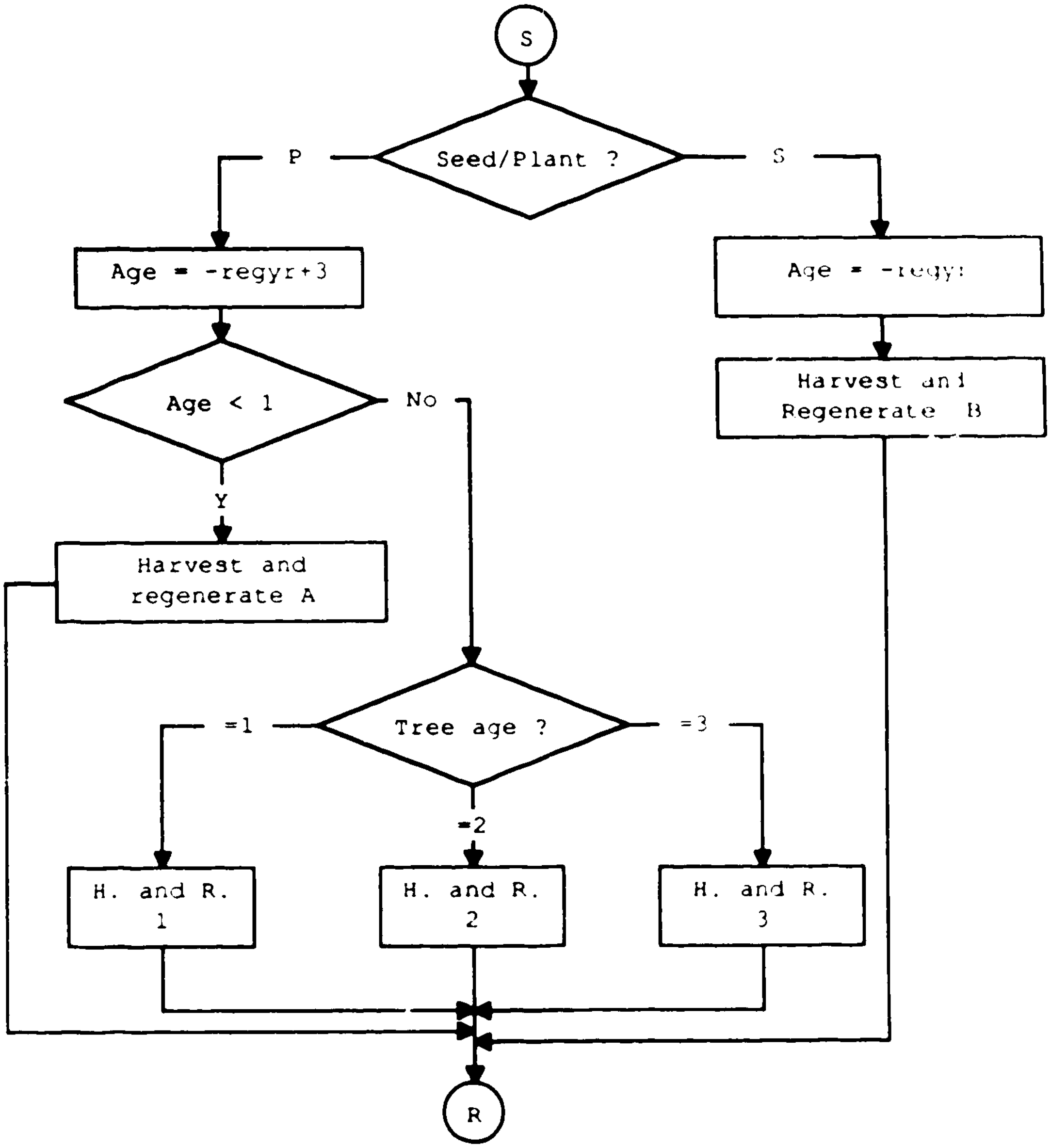


Figure B.4 Continued.

APPENDIX B.2: PROGRAM LISTING OF THE SIMULATION MODEL

```
* ############################################################################
*
*                              MAIN PROGRAM IRPM
*                   FOR THESIS PROJECT - SIMULATION MODELING
*                                     BY
*                                 DALI ZHANG
*                    DEPARTMENT OF AGRICULTURAL ENGINEERING
*                           UNIVERSITY OF ALBERTA
* ############################################################################
* Date of start: October 1, 1988
* Supervised by Prof. A. W. Anderson
* -----------------------------------------------------------------
* This is a computer simulation model which mimics the decision
* making process of the Berland Sub-regional integrated resource
* planning in Alberta, Canada. The model consists of seven
* programs -- main, add record, modify record, modify structure,
*               simulatA, simulatB, and simulatC.
* The programs were written in dBASE IV programming language.
* -----------------------------------------------------------------
SET TALK OFF
SET BELL OFF
SET DECIMALS TO 6
SET PROCEDURE TO Procedf
SET PROCEDURE TO Procedfl
* initialize memory variables
CLEAR
COPY FILE IRPF.DBF TO IRPFT.DBF
SELECT 1
USE IRPT
townsh = TOWNSHIP
range = RANGE
sect = SECTION
qsect = QSECTION
DO WHILE .T.
   CLEAR
   @ 4,15 TO 18,65 DOUBLE
   @ 7,24 SAY "             MAIN MENU             "
   @ 8,24 SAY "            ============           "
   @ 9,20 SAY "                                   "
   @ 10,24 SAY "       1. Add Record               "
   @ 11,24 SAY "       2. Modify Record            "
   @ 12,24 SAY '       3. Modify structure         "
   @ 13,24 SAY "       4. Simulate A               "
   @ 14,24 SAY "       5. Simulate B
   @ 15,24 SAY "       6. Simulate C               "
   @ 16,24 SAY "       7. Exit                     "
   @ 17,2 SAY "    "
   ?
   INPUT "      Please enter a proper number for action " TO Num
   IF Num>7
      ?
      ? "         Wrong entry, please try again!!! "
      SET ESCAPE OFF
      WAIT SPACE(19)+"Press any key to continue."
   ENDIF
   SET ESCAPE ON
   DO CASE
      CASE Num=1
         APPEND
      CASE Num=2
         CLEAR
         @ 6,25 SAY "Enter township number: "
         @ 6,60 GET townsh
```

```
         @ 8,25 SAY "Enter range number: "
         @ 8,60 GET range
         @ 10,25 SAY "Enter section number: "
         @ 10,60 GET sect
         @ 12,25 SAY "Enter quarter section number: "
         @ 12,60 GET qsect
         READ
         LOCATE FOR TOWNSHIP = townsh .AND. range = range .AND. ;
            SECTION = sect .AND. QSECTION = qsect
         EDIT
      CASE Num=3
         DO Modify
      CASE Num=4
         DO SimulatA
      CASE Num=5
         DO SimulatB
      CASE Num=6
         DO SimulatC
      CASE Num=7
         CLEAR
         @ 9,3 SAY "  "
         ? "                        WELCOME USING <IRPM>, GOODBYE!"
         CLOSE ALL
         RETURN
   ENDCASE
ENDDO
PROGRAM SIMULATA
SET TALK OFF
* initialize memory variables
rep7 = SPACE(1)
rep8 = SPACE(1)
rep9 = SPACE(1)
rep20 = SPACE(1)
deltat = 0
years = 0
rot = 0
regyr = 0
pl = 0
sd = 0
per = 0.0
thinper = 0.01
STORE 0 TO v_sw, v_sb, v_p, v_ap
STORE 0 TO thinyr, thintr
CLEAR
@ 10,15 SAY "***********************************************"
@ 11,15 SAY "                    SIMULATA                   "
@ 12,15 SAY "***********************************************"
?
ACCEPT "Do you want to do simulation now? (Y/N)" TO rep7
IF UPPER(rep7) = "Y"
   DO WHILE .T.
      SELECT 6
      USE IRPRPTB
      DELETE ALL
      PACK
      SELECT 1
      CLEAR
      @ 3,5 SAY "Enter the number of years to simulate:"
      @ 3,50 GET years
      @ 5,5 SAY "Enter the time interval of simulation:"
      @ 5,50 GET deltat
      @ 7,5 SAY "Enter the rotation length:"
      @ 7,50 GET rot
      @ 9,5 SAY "Enter the number of years to regenerate "
      @ 10,5 SAY "                forest after harvesting:"
```

```
@ 10,50 GET regyr
@ 12,5 SAY "Enter the regeneration species, "
@ 13,5 SAY "    white spruce or pine? (S/P):"
@ 13,50 GET rep8
@ 15,5 SAY "Enter the method to regenerate forest, "
@ 16,5 SAY "              planting or seeding? (P/S):"
@ 16,50 GET rep9
READ
CLEAR
@ 7,5 SAY "Do you want to do precommercial thinning? (Y/N):"
@ 7,60 GET rep20
READ
IF UPPER(rep20) = "Y"
   @ 9,5 SAY "Enter the precommercial thinning age:"
   @ 9,60 GET thinyr
   @ 11,5 SAY "Enter the percentage of basal area to "
   @ 12,5 SAY "                    be left after thinning:"
   @ 12,60 GET thinper
   @ 14,5 SAY "Enter the stand density after thinning:"
   @ 14,60 GET thintr
   READ
ENDIF
CLEAR
hr = 0.10*years/deltat
@ 9,25 SAY "Simulating..., please wait!"
@ 13,20 SAY "Simulation will take about "+str(hr,4,2)+" hour(s). "
STORE 1 TO t
STORE 0 TO tl
yc = 2*INT(3257/rot)
ycv = yc
DO WHILE t <= years
   GO TOP
   DO CASE
      CASE t > rot  AND  .NOT. MOD(t,rot) = 0
         t0 = MOD(t,rot)
      CASE t < rot
         t0 = t
      CASE t >= rot .AND. MOD(t,rot) =
         t0 = rot
   ENDCASE
   IF t0 <= INT(rot/2)
      GOTO tl + 1
      DO CASE
         CASE t0 < INT(rot/2)
            DO WHILE tl < ycv
               v_sw = v_sw + VOLUME_SW
               v_sb = v_sb + VOLUME_SB
               v_p = v_p + VOLUME_P
               v_ap = v_ap + VOLUME_AP
               DO CASE
                  CASE SITE = "G"
                     DO Harreggl
                  CASE SITE = "M"
                     DO Harregml
                  CASE SITE = "F"
                     DO Harregfl
               ENDCASE
               SKIP
               tl = tl + 1
            ENDDO
            ycv = ycv + yc
         CASE t0 = INT(rot/2)
            DO WHILE .NOT. EOF()
               v_sw = v_sw + VOLUME_SW
               v_sb = v_sb + VOLUME_SB
```

```
            v_p = v_p + VOLUME_P
            v_ap = v_ap + VOLUME_AP
            DO CASE
               CASE SITE = "G"
                  DO Harregg1
               CASE SITE = "M"
                  DO Harregm1
               CASE SITE = "F"
                  DO Harregf1
            ENDCASE
            SKIP
         ENDDO
         ycv = yc
         t1 = 0
   ENDCASE
ELSE
   GOTO t1 + 1
   DO CASE
      CASE t0 < rot
         DO WHILE t1 < ycv
            v_sw = v_sw + VOLUME_SW2
            v_sb = v_sb + VOLUME_SB2
            v_p = v_p + VOLUME_P2
            v_ap = v_ap + VOLUME_AP2
            DO CASE
               CASE SITE = "G"
                  DO Harregg2
               CASE SITE = "M"
                  DO Harregm2
               CASE SITE = "F"
                  DO Harregf2
            ENDCASE
            SKIP
            t1 = t1 + 1
         ENDDO
         ycv = ycv + yc
      CASE t0 = rot
         DO WHILE .NOT. EOF()
            v_sw = v_sw + VOLUME_SW2
            v_sb = v_sb + VOLUME_SB2
            v_p = v_p + VOLUME_P2
            v_ap = v_ap + VOLUME_AP2
            DO CASE
               CASE SITE = "G"
                  DO Harregg2
               CASE SITE = "M"
                  DO Harregm2
               CASE SITE = "F"
                  DO Harregf2
            ENDCASE
            SKIP
         ENDDO
         ycv = yc
         t1 = 0
   ENDCASE
ENDIF
DECLARE Vol[1,12]
Vol[1,2] = v_sw*64
Vol[1,3] = v_sb*64
Vol[1,4] = v_p*64
Vol[1,5] = v_ap*64
GO TOP
DO WHILE .NOT. EOF()
   DO CASE
      CASE SITE = "G"
```

```
                  DO Growgsw
                  DO Growgsw2
                  DO Growgsb
                  DO Growgsb2
                  DO Growgp
                  DO Growgp2
                  DO Growgap
                  DO Growgap2
               CASE SITE = "M"
                  DO Growmsw
                  DO Growmsw2
                  DO Growmsb
                  DO Growmsb2
                  DO Growmp
                  DO Growmp2
                  DO Growmap
                  DO Growmap2
               CASE SITE = "F"
                  DO Growfsw
                  DO Growfsw2
                  DO Growfsb
                  DO Growfsb2
                  DO Growfp
                  DO Growfp2
                  DO Growfap
                  DO Growfap2
            ENDCASE
            DO Beauty
            DO Hunting
            DO Water
            DO Grawld
            IF UPPER(rep20) = "Y"
               DO Thinning
            ENDIF
            SKIP
         ENDDO
         DO ReportB
         t = t + deltat
         STORE 0 TO v_sw, v_sb, v_p, v_ap
      ENDDO
      SELECT 6
      COPY TO IRPRPTB TYPE WKS
      REPORT FORM IRPRPTB1 TO PRINTER
      REPORT FORM IRPRPTB2 TO PRINTER
      USE
      CLEAR
      rep10 = " "
      DO WHILE .NOT. rep10$"YyNn"
         @ 10,15 SAY "Do you want to do more simulation? (Y/N)" GET;
                      rep10
         READ
      ENDDO
      IF UPPER(rep10) = "N"
         RELEASE t
         EXIT
      ENDIF
   ENDDO
ENDIF
SET TALK ON
RETURN
Program SIMULATB
SET TALK OFF
SET ESCAPE OFF
CLEAR
DO WHILE .T.
```

```
   CLEAR
   @ 5,15 TO 16,65 DOUBLE
   @ 8,24 SAY  "         SIMULATION MENU       "
   @ 9,24 SAY  "       ------------------      "
   @ 10,24 SAY " "
   @ 11,24 SAY "          1. Harvest           "
   @ 12,24 SAY "          2. Growth            "
   @ 13,24 SAY "          3. Regeneration      "
   @ 14,24 SAY "          4. Exit              "
   @ 18,20 SAY " "
   INPUT "    Please enter a proper number for action " to SNum
   IF SNum>4
      ?
      ?     "    Incorrect entry, please try again! "
      WAIT SPACE(15)+"Press any key to continue..."
   ENDIF
   IF SNum=1
      DO Harvest
   ENDIF
   IF SNum=2
      DO Grow
   ENDIF
   IF SNum=3
      DO Regenerate
   ENDIF
   IF SNum=4
      CLEAR
      @ 5,20 SAY "WELCOME USING <SIMULATE>!"
      SET ESCAPE ON
      SET TALK ON
      RETURN
   ENDIF
ENDDO
```

Program SIMULATC

```
*
SET TALK OFF
* initialize memory variables
rep7 = SPACE(1)
rep8 = SPACE(1)
rep9 = SPACE(1)
rep20 = SPACE(1)
deltat = 0.5
years = 0
STORE 0 TO t, tl
CLEAR
@ 10,15 SAY "************************************************"
@ 11,15 SAY "*                     SIMULATC                 *"
@ 12,15 SAY "************************************************"
?
ACCEPT "Do you want to do simulation now? (Y/N)" TO rep7
IF UPPER(rep7) = "Y"
   DO WHILE .T.
      SELECT 3
      USE IRPRPT1
      DELETE ALL FOR YEAR > 0
      PACK
      SELECT 4
      USE IRPRPT2
      DELETE ALL FOR YEAR > 0
      PACK
      SELECT 5
      USE IRPRPT3
      DELETE ALL FOR YEAR > 0
      PACK
      SELECT 1
```

```
CLEAR
@ 3,5 SAY "Enter the number of years to simulate:"
@ 3,50 GET years
@ 5,5 SAY "Enter the time interval of simulation:"
@ 5,50 GET deltat
READ
DO WHILE t < years
   deltT - deltat
   GO TOP
   DO WHILE .NOT. EOF()
      DO CASE
         CASE SITE - "G"
            DO Growgsw
            DO Growgsw2
            DO Growgsb
            DO Growgsb2
            DO Growgp
            DO Growgp2
            DO Growgap
            DO Growgap2
         CASE SITE - "M"
            DO Growmsw
            DO Growmsw2
            DO Growmsb
            DO Growmsb2
            DO Growmp
            DO Growmp2
            DO Growmap
            DO Growmap2
         CASE SITE - "F"
            DO Growfsw
            DO Growfsw2
            DO Growfsb
            DO Growfsb2
            DO Growfp
            DO Growfp2
            DO Growfap
            DO Growfap2
      ENDCASE
      DO Beauty
      DO Hunting
      DO Water
      DO Grawld
      DO Thinning
      SKIP
   ENDDO
   t - t + deltat
   DO Reportc
ENDDO
SELECT 3
COPY TO IRPRPT1 TYPE WKS
USE
SELECT 4
COPY TO IRPRPT2 TYPE WKS
USE
SELECT 5
COPY TO IRPRPT3 TYPE WKS
USE
CLEAR
repl0 - " "
DO WHILE .NOT. repl0$"YyNn"
   @ 10,15 SAY "Do you want to do more simulation? (Y/N)" GET;
               repl0
   READ
ENDDO
```

```
      IF UPPER(rep10) = "N"
         RELEASE t
         EXIT
      ENDIF
   ENDDO
ENDIF
SET TALK ON
RETURN
Subroutine HARREG
* HARREGG1.PRG
*
DO CASE
   CASE UPPER(rep8) = "S"
      IF UPPER(rep9) = "P"
         IF .NOT. AGE_SW = 0
            REPLACE AGE_SW WITH - regyr + 3
            DO CASE
               CASE -regyr+3 < 1
                  REPLACE BA_SW WITH 0.0
                  REPLACE HEIGHT_SW WITH 0.15
                  REPLACE DIA_SW WITH 0.035
                  REPLACE NO_TREE_SW WITH 16000
                  REPLACE VOLUME_SW WITH 0.0
               CASE -regyr+3 = 1
                  REPLACE BA_SW WITH 0.0
                  REPLACE HEIGHT_SW WITH 0.15
                  REPLACE DIA_SW WITH 0.035
                  REPLACE NO_TREE_SW WITH 16000
                  REPLACE VOLUME_SW WITH 0.0
               CASE -regyr+3 = 2
                  REPLACE BA_SW WITH 0.0
                  REPLACE HEIGHT_SW WITH 0.17
                  REPLACE DIA_SW WITH 0.05
                  REPLACE NO_TREE_SW WITH 15406
                  REPLACE VOLUME_SW WITH 0.0
               CASE -regyr+3 = 3
                  REPLACE BA_SW WITH 0.0
                  REPLACE HEIGHT_SW WITH 0.2
                  REPLACE DIA_SW WITH 0.06
                  REPLACE NO_TREE_SW WITH 14831
                  REPLACE VOLUME_SW WITH 0.0
            ENDCASE
         ENDIF
*
         IF .NOT. AGE_SB = 0
            REPLACE AGE_SB WITH 0
            REPLACE BA_SB WITH 0.0
            REPLACE HEIGHT_SB WITH 0.15
            REPLACE DIA_SB WITH 0.005
            REPLACE NO_TREE_SB WITH 0
            REPLACE VOLUME_SB WITH 0.0
         ENDIF
*
         IF .NOT. AGE_P = 0
            REPLACE AGE_P WITH 0
            REPLACE BA_P WITH 0.0
            REPLACE HEIGHT_P WITH 0.7
            REPLACE DIA_P WITH 0.1
            REPLACE NO_TREE_P WITH 0
            REPLACE VOLUME_P WITH 0.0
         ENDIF
*
         IF .NOT. AGE_AP = 0
            REPLACE AGE_AP WITH 0
            REPLACE BA_AP WITH 0.0
```

```
            REPLACE HEIGHT_AP WITH 0.015
            REPLACE DIA_AP WITH 0.015
            REPLACE NO_TREE_AP WITH 0
            REPLACE VOLUME_AP WITH 0.0
         ENDIF
      ENDIF
      IF UPPER(rep9) = "S"
         IF .NOT. AGE_SW = 0
            REPLACE AGE_SW WITH -regyr
            REPLACE BA_SW WITH 0.0
            REPLACE HEIGHT_SW WITH 0.15
            REPLACE DIA_SW WITH 0.035
            REPLACE NO_TREE_SW WITH 16000
            REPLACE VOLUME_SW WITH 0.0
         ENDIF
*
         IF .NOT. AGE_SB = 0
            REPLACE AGE_SB WITH 0
            REPLACE BA_SB WITH 0.0
            REPLACE HEIGHT_SB WITH 0.15
            REPLACE DIA_SB WITH 0.005
            REPLACE NO_TREE_SB WITH 0
            REPLACE VOLUME_SB WITH 0.0
         ENDIF
*
         IF .NOT. AGE_P = 0
            REPLACE AGE_P WITH 0
            REPLACE BA_P WITH 0.0
            REPLACE HEIGHT_P WITH 0.7
            REPLACE DIA_P WITH 0.1
            REPLACE NO_TREE_P WITH 0
            REPLACE VOLUME_P WITH 0.0
         ENDIF
*
         IF .NOT. AGE_AP = 0
            REPLACE AGE_AP WITH 0
            REPLACE BA_AP WITH 0.0
            REPLACE HEIGHT_AP WITH 0.015
            REPLACE DIA_AP WITH 0.015
            REPLACE NO_TREE_AP WITH 0
            REPLACE VOLUME_AP WITH 0.0
         ENDIF
      ENDIF
   CASE UPPER(rep8) = "P"
      IF UPPER(rep9) = "P"
         IF .NOT. AGE_SW = 0
            REPLACE AGE_SW WITH 0
            REPLACE BA_SW WITH 0.0
            REPLACE HEIGHT_SW WITH 0.15
            REPLACE DIA_SW WITH 0.035
            REPLACE NO_TREE_SW WITH 0
            REPLACE VOLUME_SW WITH 0.0
         ENDIF
*
         IF .NOT. AGE_SB = 0
            REPLACE AGE_SB WITH 0
            REPLACE BA_SB WITH 0.0
            REPLACE HEIGHT_SB WITH 0.15
            REPLACE DIA_SB WITH 0.005
            REPLACE NO_TREE_SB WITH 0
            REPLACE VOLUME_SB WITH 0.0
         ENDIF
*
         IF .NOT. AGE_P = 0
            REPLACE AGE_P WITH -regyr + 3
```

```
            DO CASE
               CASE -regyr+3 < 1
                  REPLACE BA_P WITH 0.0
                  REPLACE HEIGHT_P WITH 0.7
                  REPLACE DIA_P WITH 0.1
                  REPLACE NO_TREE_P WITH 40000
                  REPLACE VOLUME_P WITH 0.0
               CASE -regyr+3 = 1
                  REPLACE BA_P WITH 0.0
                  REPLACE HEIGHT_P WITH 0.7
                  REPLACE DIA_P WITH 0.1
                  REPLACE NO_TREE_P WITH 40000
                  REPLACE VOLUME_P WITH 0.0
               CASE -regyr+3 = 2
                  REPLACE BA_P WITH 0.0
                  REPLACE HEIGHT_P WITH 0.79
                  REPLACE DIA_P WITH 0.13
                  REPLACE NO_TREE_P WITH 38699
                  REPLACE VOLUME_P WITH 0.0
               CASE -regyr+3 = 3
                  REPLACE BA_P WITH 0.0
                  REPLACE HEIGHT_P WITH 0.89
                  REPLACE DIA_P WITH 0.16
                  REPLACE NO_TREE_P WITH 37412
                  REPLACE VOLUME_P WITH 0.0
            ENDCASE
         ENDIF
*
         IF .NOT. AGE_AP = 0
            REPLACE AGE_AP WITH 0
            REPLACE BA_AP WITH 0.0
            REPLACE HEIGHT_AP WITH 0.015
            REPLACE DIA_AP WITH 0.015
            REPLACE NO_TREE_AP WITH 0
            REPLACE VOLUME_AP WITH 0.0
         ENDIF
      ENDIF
      IF UPPER(rep9) = "S"
         IF .NOT. AGE_SW = 0
            REPLACE AGE_SW WITH 0
            REPLACE BA_SW WITH 0.0
            REPLACE HEIGHT_SW WITH 0.15
            REPLACE DIA_SW WITH 0.035
            REPLACE NO_TREE_SW WITH 0
            REPLACE VOLUME_SW WITH 0.0
         ENDIF
*
         IF .NOT. AGE_SB = 0
            REPLACE AGE_SB WITH 0
            REPLACE BA_SB WITH 0.0
            REPLACE HEIGHT_SB WITH 0.15
            REPLACE DIA_SB WITH 0.005
            REPLACE NO_TREE_SB WITH 0
            REPLACE VOLUME_SB WITH 0.0
         ENDIF
*
         IF .NOT. AGE_P = 0
            REPLACE AGE_P WITH -regyr
            REPLACE BA_P WITH 0.0
            REPLACE HEIGHT_P WITH 0.7
            REPLACE DIA_P WITH 0.1
            REPLACE NO_TREE_P WITH 40000
            REPLACE VOLUME_P WITH 0.0
         ENDIF
*
```

```
         IF .NOT. AGE_AP = 0
            REPLACE AGE_AP WITH 0
            REPLACE BA_AP WITH 0.0
            REPLACE HEIGHT_AP WITH 0.015
            REPLACE DIA_AP WITH 0.015
            REPLACE NO_TREE_AP WITH 0
            REPLACE VOLUME_AP WITH 0.0
         ENDIF
      ENDIF
ENDCASE
* HARREGG2.PRG
*
DO CASE
   CASE UPPER(rep8) = "S"
      IF UPPER(rep9) = "P"
         IF .NOT. AGE_SW2 = 0
            REPLACE AGE_SW2 WITH - regyr + 3
            DO CASE
               CASE -regyr+3 < 1
                  REPLACE BA_SW2 WITH 0.0
                  REPLACE HEIGHT_SW2 WITH 0.15
                  REPLACE DIA_SW2 WITH 0.035
                  REPLACE NO_TRE_SW2 WITH 16000
                  REPLACE VOLUME_SW2 WITH 0.0
               CASE -regyr+3 = 1
                  REPLACE BA_SW2 WITH 0.0
                  REPLACE HEIGHT_SW2 WITH 0.15
                  REPLACE DIA_SW2 WITH 0.035
                  REPLACE NO_TRE_SW2 WITH 16000
                  REPLACE VOLUME_SW2 WITH 0.0
               CASE -regyr+3 = 2
                  REPLACE BA_SW2 WITH 0.0
                  REPLACE HEIGHT_SW2 WITH 0.17
                  REPLACE DIA_SW2 WITH 0.05
                  REPLACE NO_TRE_SW2 WITH 15406
                  REPLACE VOLUME_SW2 WITH 0.0
               CASE -regyr+3 = 3
                  REPLACE BA_SW2 WITH 0.0
                  REPLACE HEIGHT_SW2 WITH 0.2
                  REPLACE DIA_SW2 WITH 0.06
                  REPLACE NO_TRE_SW2 WITH 14831
                  REPLACE VOLUME_SW2 WITH 0.0
            ENDCASE
         ENDIF
*
         IF .NOT. AGE_SB2 = 0
            REPLACE AGE_SB2 WITH 0
            REPLACE BA_SB2 WITH 0.0
            REPLACE HEIGHT_SB2 WITH 0.15
            REPLACE DIA_SB2 WITH 0.005
            REPLACE NO_TRE_SB2 WITH 0
            REPLACE VOLUME_SB2 WITH 0.0
         ENDIF
*
         IF .NOT. AGE_P2 = 0
            REPLACE AGE_P2 WITH 0
            REPLACE BA_P2 WITH 0.0
            REPLACE HEIGHT_P2 WITH 0.7
            REPLACE DIA_P2 WITH 0.1
            REPLACE NO_TREE_P2 WITH 0
            REPLACE VOLUME_P2 WITH 0.0
         ENDIF
*
         IF .NOT. AGE_AP2 = 0
            REPLACE AGE_AP2 WITH 0
```

```
            REPLACE BA_AP2 WITH 0.0
            REPLACE HEIGHT_AP2 WITH 0.015
            REPLACE DIA_AP2 WITH 0.015
            REPLACE NO_TRE_AP2 WITH 0
            REPLACE VOLUME_AP2 WITH 0.0
         ENDIF
      ENDIF
      IF UPPER(rep9) = "S"
         IF .NOT. AGE_SW2 = 0
            REPLACE AGE_SW2 WITH -regyr
            REPLACE BA_SW2 WITH 0.0
            REPLACE HEIGHT_SW2 WITH 0.15
            REPLACE DIA_SW2 WITH 0.035
            REPLACE NO_TRE_SW2 WITH 16000
            REPLACE VOLUME_SW2 WITH 0.0
         ENDIF
*
         IF .NOT. AGE_SB2 = 0
            REPLACE AGE_SB2 WITH 0
            REPLACE BA_SB2 WITH 0.0
            REPLACE HEIGHT_SB2 WITH 0.15
            REPLACE DIA_SB2 WITH 0.005
            REPLACE NO_TRE_SB2 WITH 0
            REPLACE VOLUME_SB2 WITH 0.0
         ENDIF
*
         IF .NOT. AGE_P2 = 0
            REPLACE AGE_P2 WITH 0
            REPLACE BA_P2 WITH 0.0
            REPLACE HEIGHT_P2 WITH 0.7
            REPLACE DIA_P2 WITH 0.1
            REPLACE NO_TREE_P2 WITH 0
            REPLACE VOLUME_P2 WITH 0.0
         ENDIF
*
         IF .NOT. AGE_AP2 = 0
            REPLACE AGE_AP2 WITH 0
            REPLACE BA_AP2 WITH 0.0
            REPLACE HEIGHT_AP2 WITH 0.015
            REPLACE DIA_AP2 WITH 0.015
            REPLACE NO_TRE_AP2 WITH 0
            REPLACE VOLUME_AP2 WITH 0.0
         ENDIF
      ENDIF
   CASE UPPER(rep8) = "P"
      IF UPPER(rep9) = "P"
         IF .NOT. AGE_SW2 = 0
            REPLACE AGE_SW2 WITH 0
            REPLACE BA_SW2 WITH 0.0
            REPLACE HEIGHT_SW2 WITH 0.15
            REPLACE DIA_SW2 WITH 0.035
            REPLACE NO_TRE_SW2 WITH 0
            REPLACE VOLUME_SW2 WITH 0.0
         ENDIF
*
         IF .NOT. AGE_SB2 = 0
            REPLACE AGE_SB2 WITH 0
            REPLACE BA_SB2 WITH 0.0
            REPLACE HEIGHT_SB2 WITH 0.15
            REPLACE DIA_SB2 WITH 0.005
            REPLACE NO_TRE_SB2 WITH 0
            REPLACE VOLUME_SB2 WITH 0.0
         ENDIF
*
         IF .NOT. AGE_P2 = 0
```

```
         REPLACE AGE_P2 WITH -regyr + 3
         DO CASE
            CASE -regyr+3 < 1
               REPLACE BA_P2 WITH 0.0
               REPLACE HEIGHT_P2 WITH 0.7
               REPLACE DIA_P2 WITH 0.1
               REPLACE NO_TREE_P2 WITH 40000
               REPLACE VOLUME_P2 WITH 0.0
            CASE -regyr+3 = 1
               REPLACE BA_P2 WITH 0.0
               REPLACE HEIGHT_P2 WITH 0.7
               REPLACE DIA_P2 WITH 0.1
               REPLACE NO_TREE_P2 WITH 40000
               REPLACE VOLUME_P2 WITH 0.0
            CASE -regyr+3 = 2
               REPLACE BA_P2 WITH 0.0
               REPLACE HEIGHT_P2 WITH 0.79
               REPLACE DIA_P2 WITH 0.13
               REPLACE NO_TREE_P2 WITH 38699
               REPLACE VOLUME_P2 WITH 0.0
            CASE -regyr+3 = 3
               REPLACE BA_P2 WITH 0.0
               REPLACE HEIGHT_P2 WITH 0.89
               REPLACE DIA_P2 WITH 0.16
               REPLACE NO_TREE_P2 WITH 37412
               REPLACE VOLUME_P2 WITH 0.0
         ENDCASE
      ENDIF
*
      IF .NOT. AGE_AP2 = 0
         REPLACE AGE_AP2 WITH 0
         REPLACE BA_AP2 WITH 0.0
         REPLACE HEIGHT_AP2 WITH 0.015
         REPLACE DIA_AP2 WITH 0.015
         REPLACE NO_TRE_AP2 WITH 0
         REPLACE VOLUME_AP2 WITH 0.0
      ENDIF
   ENDIF
   IF UPPER(rep9) = "S"
      IF .NOT. AGE_SW2 = 0
         REPLACE AGE_SW2 WITH 0
         REPLACE BA_SW2 WITH 0.0
         REPLACE HEIGHT_SW2 WITH 0.15
         REPLACE DIA_SW2 WITH 0.035
         REPLACE NO_TRE_SW2 WITH 0
         REPLACE VOLUME_SW2 WITH 0.0
      ENDIF
*
      IF .NOT. AGE_SB2 = 0
         REPLACE AGE_SB2 WITH 0
         REPLACE BA_SB2 WITH 0.0
         REPLACE HEIGHT_SB2 WITH 0.15
         REPLACE DIA_SB2 WITH 0.005
         REPLACE NO_TRE_SB2 WITH 0
         REPLACE VOLUME_SB2 WITH 0.0
      ENDIF
*
      IF .NOT. AGE_P2 = 0
         REPLACE AGE_P2 WITH -regyr
         REPLACE BA_P2 WITH 0.0
         REPLACE HEIGHT_P2 WITH 0.7
         REPLACE DIA_P2 WITH 0.1
         REPLACE NO_TREE_P2 WITH 40000
         REPLACE VOLUME_P2 WITH 0.0
      ENDIF
```

```
         IF .NOT. AGE_AP2 = 0
            REPLACE AGE_AP2 WITH 0
            REPLACE BA_AP2 WITH 0.0
            REPLACE HEIGHT_AP2 WITH 0.015
            REPLACE DIA_AP2 WITH 0.015
            REPLACE NO_TRE_AP2 WITH 0
            REPLACE VOLUME_AP2 WITH 0.0
         ENDIF
      ENDIF
ENDCASE
* HARREGM1.PRG
*
NOTE This program modify database file because of harvesting
*
DO CASE
   CASE UPPER(rep8) = "S"
      IF UPPER(rep9) = "P"
         IF .NOT. AGE_SW = 0
            REPLACE AGE_SW WITH - regyr + 3
            DO CASE
               CASE -regyr+3 < 1
                  REPLACE BA_SW WITH 0.0
                  REPLACE HEIGHT_SW WITH 0.15
                  REPLACE DIA_SW WITH 0.005
                  REPLACE NO_TREE_SW WITH 43000
                  REPLACE VOLUME_SW WITH 0.0
               CASE -regyr+3 = 1
                  REPLACE BA_SW WITH 0.0
                  REPLACE HEIGHT_SW WITH 0.15
                  REPLACE DIA_SW WITH 0.005
                  REPLACE NO_TREE_SW WITH 43000
                  REPLACE VOLUME_SW WITH 0.0
               CASE -regyr+3 = 2
                  REPLACE BA_SW WITH 0.0
                  REPLACE HEIGHT_SW WITH 0.17
                  REPLACE DIA_SW WITH 0.01
                  REPLACE NO_TREE_SW WITH 41498
                  REPLACE VOLUME_SW WITH 0.0
               CASE -regyr+3 = 3
                  REPLACE BA_SW WITH 0.0
                  REPLACE HEIGHT_SW WITH 0.19
                  REPLACE DIA_SW WITH 0.01
                  REPLACE NO_TREE_SW WITH 40045
                  REPLACE VOLUME_SW WITH 0.0
            ENDCASE
         ENDIF
 *
         IF .NOT. AGE_SB = 0
            REPLACE AGE_SB WITH 0
            REPLACE BA_SB WITH 0.0
            REPLACE HEIGHT_SB WITH 0.15
            REPLACE DIA_SB WITH 0.004
            REPLACE NO_TREE_SB WITH 0
            REPLACE VOLUME_SB WITH 0.0
         ENDIF
 *
         IF .NOT. AGE_P = 0
            REPLACE AGE_P WITH 0
            REPLACE BA_P WITH 0.0
            REPLACE HEIGHT_P WITH 0.5
            REPLACE DIA_P WITH 0.08
            REPLACE NO_TREE_P WITH 0
            REPLACE VOLUME_P WITH 0.0
         ENDIF
 *
```

```
         IF .NOT. AGE_AP = 0
            REPLACE AGE_AP WITH 0
            REPLACE BA_AP WITH 0.0
            REPLACE HEIGHT_AP WITH 0.015
            REPLACE DIA_AP WITH 0.015
            REPLACE NO_TREE_AP WITH 0
            REPLACE VOLUME_AP WITH 0.0
         ENDIF
      ENDIF

      IF UPPER(rep9) = "S"
         IF .NOT. AGE_SW = 0
            REPLACE AGE_SW WITH -regyr
            REPLACE BA_SW WITH 0.0
            REPLACE HEIGHT_SW WITH 0.15
            REPLACE DIA_SW WITH 0.005
            REPLACE NO_TREE_SW WITH 43000
            REPLACE VOLUME_SW WITH 0.0
         ENDIF
*
         IF .NOT. AGE_SB = 0
            REPLACE AGE_SB WITH 0
            REPLACE BA_SB WITH 0.0
            REPLACE HEIGHT_SB WITH 0.15
            REPLACE DIA_SB WITH 0.004
            REPLACE NO_TREE_SB WITH 0
            REPLACE VOLUME_SB WITH 0.0
         ENDIF
*
         IF .NOT. AGE_P = 0
            REPLACE AGE_P WITH 0
            REPLACE BA_P WITH 0.0
            REPLACE HEIGHT_P WITH 0.5
            REPLACE DIA_P WITH 0.08
            REPLACE NO_TREE_P WITH 0
            REPLACE VOLUME_P WITH 0.0
         ENDIF
*
         IF .NOT. AGE_AP = 0
            REPLACE AGE_AP WITH 0
            REPLACE BA_AP WITH 0.0
            REPLACE HEIGHT_AP WITH 0.015
            REPLACE DIA_AP WITH 0.015
            REPLACE NO_TREE_AP WITH 0
            REPLACE VOLUME_AP WITH 0.0
         ENDIF
      ENDIF
   CASE UPPER(rep8) = "P"
      IF UPPER(rep9) = "P"
         IF .NOT. AGE_SW = 0
            REPLACE AGE_SW WITH 0
            REPLACE BA_SW WITH 0.0
            REPLACE HEIGHT_SW WITH 0.15
            REPLACE DIA_SW WITH 0.005
            REPLACE NO_TREE_SW WITH 0
            REPLACE VOLUME_SW WITH 0.0
         ENDIF
*
         IF .NOT. AGE_SB = 0
            REPLACE AGE_SB WITH 0
            REPLACE BA_SB WITH 0.0
            REPLACE HEIGHT_SB WITH 0.15
            REPLACE DIA_SB WITH 0.004
            REPLACE NO_TREE_SB WITH 0
```

```
             REPLACE VOLUME_SB WITH 0.0
          ENDIF
*
          IF .NOT. AGE_P = 0
             REPLACE AGE_P WITH -regyr + 3
             DO CASE
                CASE -regyr+3 < 1
                   REPLACE BA_P WITH 0.0
                   REPLACE HEIGHT_P WITH 0.5
                   REPLACE DIA_P WITH 0.08
                   REPLACE NO_TREE_P WITH 80000
                   REPLACE VOLUME_P WITH 0.0
                CASE -regyr+3 = 1
                   REPLACE BA_P WITH 0.0
                   REPLACE HEIGHT_P WITH 0.5
                   REPLACE DIA_P WITH 0.08
                   REPLACE NO_TREE_P WITH 80000
                   REPLACE VOLUME_P WITH 0.0
                CASE -regyr+3 = 2
                   REPLACE BA_P WITH 0.0
                   REPLACE HEIGHT_P WITH 0.57
                   REPLACE DIA_P WITH 0.1
                   REPLACE NO_TREE_P WITH 76829
                   REPLACE VOLUME_P WITH 0.0
                CASE -regyr+3 = 3
                   REPLACE BA_P WITH 0.0
                   REPLACE HEIGHT_P WITH 0.65
                   REPLACE DIA_P WITH 0.11
                   REPLACE NO_TREE_P WITH 73749
                   REPLACE VOLUME_P WITH 0.0
             ENDCASE
          ENDIF
*
          IF .NOT. AGE_AP = 0
             REPLACE AGE_AP WITH 0
             REPLACE BA_AP WITH 0.0
             REPLACE HEIGHT_AP WITH 0.015
             REPLACE DIA_AP WITH 0.015
             REPLACE NO_TREE_AP WITH 0
             REPLACE VOLUME_AP WITH 0.0
          ENDIF
       ENDIF
       IF UPPER(rep9) = "S"
          IF .NOT. AGE_SW = 0
             REPLACE AGE_SW WITH 0
             REPLACE BA_SW WITH 0.0
             REPLACE HEIGHT_SW WITH 0.15
             REPLACE DIA_SW WITH 0.005
             REPLACE NO_TREE_SW WITH 0
             REPLACE VOLUME_SW WITH 0.0
          ENDIF
*
          IF .NOT. AGE_SB = 0
             REPLACE AGE_SB WITH 0
             REPLACE BA_SB WITH 0.0
             REPLACE HEIGHT_SB WITH 0.15
             REPLACE DIA_SB WITH 0.004
             REPLACE NO_TREE_SB WITH 0
             REPLACE VOLUME_SB WITH 0.0
          ENDIF
*
          IF .NOT. AGE_P = 0
             REPLACE AGE_P WITH -regyr
             REPLACE BA_P WITH 0.0
             REPLACE HEIGHT_P WITH 0.5
```

```
            REPLACE DIA_P WITH 0.08
            REPLACE NO_TREE_P WITH 80000
            REPLACE VOLUME_P WITH 0.0
         ENDIF
*
         IF .NOT. AGE_AP = 0
            REPLACE AGE_AP WITH 0
            REPLACE BA_AP WITH 0.0
            REPLACE HEIGHT_AP WITH 0.015
            REPLACE DIA_AP WITH 0.015
            REPLACE NO_TREE_AP WITH 0
            REPLACE VOLUME_AP WITH 0.0
         ENDIF
      ENDIF
ENDCASE
* HAPREGM2.PRG
*
DO CASE
   CASE UPPER(rep8) = "S"
      IF UPPER(rep9) = "P"
         IF .NOT. AGE_SW2 = 0
            REPLACE AGE_SW2 WITH - regyr + 3
            DO CASE
               CASE -regyr+3 < 1
                  REPLACE BA_SW2 WITH 0.0
                  REPLACE HEIGHT_SW2 WITH 0.15
                  REPLACE DIA_SW2 WITH 0.005
                  REPLACE NO_TRE_SW2 WITH 43000
                  REPLACE VOLUME_SW2 WITH 0.0
               CASE -regyr+3 = 1
                  REPLACE BA_SW2 WITH 0.0
                  REPLACE HEIGHT_SW2 WITH 0.[illegible]
                  REPLACE DIA_SW2 WITH 0.005
                  REPLACE NO_TRE_SW2 WITH 43000
                  REPLACE VOLUME_SW2 WITH 0.0
               CASE -regyr+3 = 2
                  REPLACE BA_SW2 WITH 0.0
                  REPLACE HEIGHT_SW2 WITH 0.17
                  REPLACE DIA_SW2 WITH 0.01
                  REPLACE NO_TRE_SW2 WITH 41498
                  REPLACE VOLUME_SW2 WITH 0.0
               CASE -regyr+3 = 3
                  REPLACE BA_SW2 WITH 0.0
                  REPLACE HEIGHT_SW2 WITH 0.19
                  REPLACE DIA_SW2 WITH 0.01
                  REPLACE NO_TRE_SW2 WITH 40045
                  REPLACE VOLUME_SW2 WITH 0.0
            ENDCASE
         ENDIF
*
         IF .NOT. AGE_SB2 = 0
            REPLACE AGE_SB2 WITH 0
            REPLACE BA_SB2 WITH 0.0
            REPLACE HEIGHT_SB2 WITH 0.15
            REPLACE DIA_SB2 WITH 0.004
            REPLACE NO_TRE_SB2 WITH 0
            REPLACE VOLUME_SB2 WITH 0.0
         ENDIF
*
         IF .NOT. AGE_P2 = 0
            REPLACE AGE_P2 WITH 0
            REPLACE BA_P2 WITH 0.0
            REPLACE HEIGHT_P2 WITH 0.5
            REPLACE DIA_P2 WITH 0.08
            REPLACE NO_TREE_P2 WITH 0
```

```
             REPLACE VOLUME_P2 WITH 0.0
          ENDIF
*
          IF .NOT. AGE_AP2 = 0
             REPLACE AGE_AP2 WITH 0
             REPLACE BA_AP2 WITH 0.0
             REPLACE HEIGHT_AP2 WITH 0.015
             REPLACE DIA_AP2 WITH 0.015
             REPLACE NO_TRE_AP2 WITH 0
             REPLACE VOLUME_AP2 WITH 0.0
          ENDIF
       ENDIF
       IF UPPER(rep9) = "S"
          IF .NOT. AGE_SW2 = 0
             REPLACE AGE_SW2 WITH -regyr
             REPLACE BA_SW2 WITH 0.0
             REPLACE HEIGHT_SW2 WITH 0.15
             REPLACE DIA_SW2 WITH 0.005
             REPLACE NO_TRE_SW2 WITH 43000
             REPLACE VOLUME_SW2 WITH 0.0
          ENDIF
*
          IF .NOT. AGE_SB2 = 0
             REPLACE AGE_SB2 WITH 0
             REPLACE BA_SB2 W[illegible]
             REPLACE HEIGHT_[illegible] 0.15
             REPLACE DIA_SB2 [illegible] 004
             REPLACE NO_TRE[illegible] 0
             REPLACE VOLUME_[illegible] 0.0
          ENDIF
*
          IF .NOT. AGE_P2 = 0
             REPLACE AGE_P2 WITH 0
             REPLACE BA_P2 WITH 0.0
             REPLACE HEIGHT_P2 WITH 0.5
             REPLACE DIA_P2 WITH 0.0[illegible]
             REPLACE NO_TREE_P2 WITH [illegible]
             REPLACE VOLUME_P2 WITH 0.0
          ENDIF
*
          IF .NOT. AGE_AP2 = 0
             REPLACE AGE_AP2 WITH 0
             REPLACE BA_AP2 WITH 0.0
             REPLACE HEIGHT_AP2 WITH 0.015
             REPLACE DIA_AP2 WITH 0.015
             REPLACE NO_TRE_AP2 WITH 0
             REPLACE VOLUME_AP2 WITH 0.0
          ENDIF
       ENDIF
    CASE UPPER(rep8) = "[illegible]"
       IF UPPER(rep9) = "P"
          IF .NOT. AGE_SW2 = 0
             REPLACE AGE_SW2 WITH 0
             REPLACE BA_SW2 WITH 0.0
             REPLACE HEIGHT_SW2 WITH 0.15
             REPLACE DIA_SW2 WITH 0.005
             REPLACE NO_TRE_SW2 WITH 0
             REPLACE VOLUME_SW2 WITH 0.0
          ENDIF
*
          IF .NOT. AGE_SB2 = 0
             REPLACE AGE_SB2 WITH 0
             REPLACE BA_SB2 WITH 0.0
             REPLACE HEIGHT_SB2 WITH 0.15
             REPLACE DIA_SB2 WITH 0.004
```

```
            REPLACE NO_TRE_SB2 WITH 0
            REPLACE VOLUME_SB2 WITH 0.0
         ENDIF
*
         IF .NOT. AGE_P2 = 0
            REPLACE AGE_P2 WITH -regyr + 3
            DO CASE
               CASE -regyr+3 < 1
                  REPLACE BA_P2 WITH 0.0
                  REPLACE HEIGHT_P2 WITH 0.5
                  REPLACE DIA_P2 WITH 0.08
                  REPLACE NO_TREE_P2 WITH 80000
                  REPLACE VOLUME_P2 WITH 0.0
               CASE -regyr+3 = [illegible]
                  REPLACE BA_P2 WITH 0.0
                  REPLACE HEIGHT_P2 WITH 0.5
                  REPLACE DIA_P2 WITH 0.08
                  REPLACE NO_TREE_P2 WITH 80000
                  REPLACE VOLUME_P2 WITH 0.0
               CASE -regyr+3 = 2
                  REPLACE BA_P2 WITH 0.0
                  REPLACE HEIGHT_P2 WITH 0.57
                  REPLACE DIA_P2 WITH 0.[illegible]
                  REPLACE NO_TREE_P2 WITH 76829
                  REPLACE VOLUME_P2 WITH 0.[illegible]
               CASE -regyr+3 = 3
                  REPLACE BA_P2 WITH 0.0
                  REPLACE HEIGHT_P2 WITH 0.[illegible]
                  REPLACE DIA_P2 WITH 0.11
                  REPLACE NO_TREE_P2 WITH 3749
                  REPLACE VOLUME_P2 WITH 0.0
            ENDCASE
         ENDIF
*
         IF .NOT. AGE_AP2 = 0
            REPLACE AGE_AP2 WITH 0
            REPLACE BA_AP2 WITH 0.0
            REPLACE HEIGHT_AP2 WITH 0.015
            REPLACE DIA_AP2 WITH 0.015
            REPLACE NO_TRE_AP2 WITH 0
            REPLACE VOLUME_AP2 WITH 0.0
         ENDIF
      ENDIF
      IF UPPER(rep9) = "S"
         IF .NOT. AGE_SW2 = 0
            REPLACE AGE_SW2 WITH 0
            REPLACE BA_SW2 WITH 0.0
            REPLACE HEIGHT_SW2 WITH 0.15
            REPLACE DIA_SW2 WITH 0.005
            REPLACE NO_TRE_SW2 WITH 0
            REPLACE VOLUME_SW2 WITH 0.0
         ENDIF
*
         IF .NOT. AGE_SB2 = 0
            REPLACE AGE_SB2 WITH 0
            REPLACE BA_SB2 WITH 0.0
            REPLACE HEIGHT_SB2 WITH 0.15
            REPLACE DIA_SB2 WITH 0.004
            REPLACE NO_TRE_SB2 WITH 0
            REPLACE VOLUME_SB2 WITH 0.0
         ENDIF
*
         IF .NOT. AGE_P2 = 0
            REPLACE AGE_P2 WITH -regyr
            REPLACE BA_P2 WITH 0.0
```

```
            REPLACE HEIGHT_P2 WITH 0.5
            REPLACE DIA_P2 WITH 0.08
            REPLACE NO_TREE_P2 WITH 80000
            REPLACE VOLUME_P2 WITH 0.0
         ENDIF
*
         IF .NOT. AGE_AP2 = 0
            REPLACE AGE_AP2 WITH 0
            REPLACE BA_AP2 WITH 0.0
            REPLACE HEIGHT_AP2 WITH 0.015
            REPLACE DIA_AP2 WITH 0.015
            REPLACE NO_TRE_AP2 WITH 0
            REPLACE VOLUME_AP2 WITH 0.0
         ENDIF
      ENDIF
     SE
     REGF1.PRG
*
DO CASE
   CASE UPPER(rep8) = "S"
      IF UPPER(rep9) = "P"
         IF .NOT. AGE_SW = 0
            REPLACE AGE_SW WITH - regyr + 3
            DO CASE
               CASE -regyr+3 < 1
                  REPLACE BA_SW WITH 0.0
                  REPLACE HEIGHT_SW WITH 0.2
                  REPLACE DIA_SW WITH 0.02
                  REPLACE NO_TREE_SW WITH 35000
                  REPLACE VOLUME_SW WITH 0.0
               CASE -regyr+3 = 1
                  REPLACE BA_SW WITH 0.0
                  REPLACE HEIGHT_SW WITH 0.2
                  REPLACE DIA_SW WITH 0.02
                  REPLACE NO_TREE_SW WITH 35000
                  REPLACE VOLUME_SW WITH 0.0
               CASE -regyr+3 = 2
                  REPLACE BA_SW WITH 0.0
                  REPLACE HEIGHT_SW WITH 0.216
                  REPLACE DIA_SW WITH 0.0235
                  REPLACE NO_TREE_SW WITH 33969
                  REPLACE VOLUME_SW WITH 0.0
               CASE -regyr+3 = 3
                  REPLACE BA_SW WITH 0.0
                  REPLACE HEIGHT_SW WITH 0.233
                  REPLACE DIA_SW WITH 0.0276
                  REPLACE NO_TREE_SW WITH p33966
                  REPLACE VOLUME_SW WITH 0.0
            ENDCASE
         ENDIF
*
         IF .NOT. AGE_SB = 0
            REPLACE AGE_SB WITH 0
            REPLACE BA_SB WITH 0.0
            REPLACE HEIGHT_SB WITH 0.35
            REPLACE DIA_SB WITH 0.003
            REPLACE NO_TREE_SB WITH 0
            REPLACE VOLUME_SB WITH 0.0
         ENDIF
*
         IF .NOT. AGE_P = 0
            REPLACE AGE_P WITH 0
            REPLACE BA_P WITH 0.0
            REPLACE HEIGHT_P WITH 0.25
            REPLACE DIA_P WITH 0.065
```

```
        REPLACE NO_TREE_P WITH 0
        REPLACE VOLUME_P WITH 0.0
      ENDIF
*
      IF .NOT. AGE_AP = 0
        REPLACE AGE_AP WITH 0
        REPLACE BA_AP WITH 0.0
        REPLACE HEIGHT_AP WITH 0.01
        REPLACE DIA_AP WITH 0.008
        REPLACE NO_TREE_AP WITH 0
        REPLACE VOLUME_AP WITH 0.0
      ENDIF
    ENDIF
    IF UPPER(rep9) = "S"
      IF .NOT. AGE_SW = 0
        REPLACE AGE_SW WITH -regyr
        REPLACE BA_SW WITH 0.0
        REPLACE HEIGHT_SW WITH 0.2
        REPLACE DIA_SW WITH 0.02
        REPLACE NO_TREE_SW WITH 35000
        REPLACE VOLUME_SW WITH 0.0
      ENDIF
*
      IF .NOT. AGE_SB = 0
        REPLACE AGE_SB WITH 0
        REPLACE BA_SB WITH 0.0
        REPLACE HEIGHT_SB WITH 0.35
        REPLACE DIA_SB WITH 0.003
        REPLACE NO_TREE_SB WITH 0
        REPLACE VOLUME_SB WITH 0.0
      ENDIF
*
      IF .NOT. AGE_P = 0
        REPLACE AGE_P WITH 0
        REPLACE BA_P WITH 0.0
        REPLACE HEIGHT_P WITH 0.25
        REPLACE DIA_P WITH 0.065
        REPLACE NO_TREE_P WITH 0
        REPLACE VOLUME_P WITH 0.0
      ENDIF
*
      IF .NOT. AGE_AP = 0
        REPLACE AGE_AP WITH 0
        REPLACE BA_AP WITH 0.0
        REPLACE HEIGHT_AP WITH 0.01
        REPLACE DIA_AP WITH 0.008
        REPLACE NO_TREE_AP WITH 0
        REPLACE VOLUME_AP WITH 0.0
      ENDIF
    ENDIF
  CASE UPPER(rep8) = "P"
    IF UPPER(rep9) = "P"
      IF .NOT. AGE_SW = 0
        REPLACE AGE_SW WITH 0
        REPLACE BA_SW WITH 0.0
        REPLACE HEIGHT_SW WITH 0.2
        REPLACE DIA_SW WITH 0.02
        REPLACE NO_TREE_SW WITH 0
        REPLACE VOLUME_SW WITH 0.0
      ENDIF
*
      IF .NOT. AGE_SB = 0
        REPLACE AGE_SB WITH 0
        REPLACE BA_SB WITH 0.0
        REPLACE HEIGHT_SB WITH 0.35
```

```
      REPLACE DIA_SB WITH 0.003
      REPLACE NO_TREE_SB WITH 0
      REPLACE VOLUME_SB WITH 0
   ENDIF
*
   IF .NOT. AGE_P = 0
      REPLACE AGE_P WITH -regyr + 3
      DO CASE
         CASE -regyr+3 < 1
            REPLACE BA_P WITH 0.0
            REPLACE HEIGHT_P WITH 0.2
            REPLACE DIA_P WITH 0.065
            REPLACE NO_TREE_P WITH 150000
            REPLACE VOLUME_P WITH 0.0
         CASE -regyr+3 = 1
            REPLACE BA_P WITH 0.0
            REPLACE HEIGHT_P WITH 0.2
            REPLACE DIA_P WITH 0.065
            REPLACE NO_TREE_P WITH 150000
            REPLACE VOLUME_P WITH 0.0
         CASE -regyr+3 = 2
            REPLACE BA_P WITH 0.0
            REPLACE HEIGHT_P WITH 0.22
            REPLACE DIA_P WITH 0.08
            REPLACE NO_TREE_P WITH 143183
            REPLACE VOLUME_P WITH 0.0
         CASE -regyr+3 = 3
            REPLACE BA_P WITH 0.0
            REPLACE HEIGHT_P WITH 0.25
            REPLACE DIA_P WITH 0.1
            REPLACE NO_TREE_P WITH 136593
            REPLACE VOLUME_P WITH 0.0
      ENDCASE
   ENDIF
*
   IF .NOT. AGE_AP = 0
      REPLACE AGE_AP WITH 0
      REPLACE BA_AP WITH 0.0
      REPLACE HEIGHT_AP WITH 0.01
      REPLACE DIA_AP WITH 0.008
      REPLACE NO_TREE_AP WITH 0
      REPLACE VOLUME_AP WITH 0.0
   ENDIF
ENDIF
IF UPPER(rep9) = "S"
   IF .NOT. AGE_SW = 0
      REPLACE AGE_SW WITH 0
      REPLACE BA_SW WITH 0.0
      REPLACE HEIGHT_SW WITH 0.2
      REPLACE DIA_SW WITH 0.02
      REPLACE NO_TREE_SW WITH 0
      REPLACE VOLUME_SW WITH 0.0
   ENDIF
*
   IF .NOT. AGE_Sb = 0
      REPLACE AGE_SB WITH 0
      REPLACE BA_SB WITH 0.0
      REPLACE HEIGHT_SB WITH 0.35
      REPLACE DIA_SB WITH 0.003
      REPLACE NO_TREE_SB WITH 0
      REPLACE VOLUME_SB WITH 0.0
   ENDIF
*
   IF .NOT. AGE_P = 0
      REPLACE AGE_P WITH -regyr
```

```
            REPLACE BA_P WITH 0.0
            REPLACE HEIGHT_P WITH 0.2
            REPLACE DIA_P WITH 0.065
            REPLACE NO_TREE_P WITH 150000
            REPLACE VOLUME_P WITH 0.0
         ENDIF
*
         IF .NOT. AGE_AP = 0
            REPLACE AGE_AP WITH 0
            REPLACE BA_AP WITH 0.0
            REPLACE HEIGHT_AP WITH 0.01
            REPLACE DIA_AP WITH 0.008
            REPLACE NO_TREE_AP WITH 0
            REPLACE VOLUME_AP WITH 0.0
         ENDIF
      ENDIF
ENDCASE
* HARREGF2.PRG
*
DO CASE
   CASE UPPER(rep8) = "S"
      IF UPPER(rep9) = "P"
         IF .NOT. AGE_SW2 = 0
            REPLACE AGE_SW2 WITH - regyr + 3
            DO CASE
               CASE -regyr+3 < 1
                  REPLACE BA_SW2 WITH 0.0
                  REPLACE HEIGHT_SW2 WITH 0.2
                  REPLACE DIA_SW2 WITH
                  REPLACE NO_TRE_SW2 WI[illegible]000
                  REPLACE VOLUME_SW2 WITH [illegible].0
               CASE -regyr+3 = 1
                  REPLACE BA_SW2 WITH 0.0
                  REPLACE HEIGHT_SW2 WITH 0.2
                  REPLACE DIA_SW2 WITH 0.02
                  REPLACE NO_TRE_SW2 WITH 35000
                  REPLACE VOLUME_SW2 WITH 0.0
               CASE -regyr+3 = 2
                  REPLACE BA_SW2 WITH 0.0
                  REPLACE HEIGHT_SW2 WITH 0.216
                  REPLACE DIA_SW2 WITH 0.0235
                  REPLACE NO_TRE_SW2 WITH 33969
                  REPLACE VOLUME_SW2 WITH 0.0
               CASE -regyr+3 = 3
                  REPLACE BA_SW2 WITH 0.0
                  REPLACE HEIGHT_SW2 WITH 0.233
                  REPLACE DIA_SW2 WITH 0.0276
                  REPLACE NO_TRE_SW2 WITH p33966
                  REPLACE VOLUME_SW2 WITH 0.0
            ENDCASE
         ENDIF
*
         IF .NOT. AGE_SB2 = 0
            REPLACE AGE_SB2 WITH 0
            REPLACE BA_SB2 WITH 0.0
            REPLACE HEIGHT_SB2 WITH 0.35
            REPLACE DIA_SB2 WITH 0.003
            REPLACE NO_TRE_SB2 WITH 0
            REPLACE VOLUME_SB2 WITH 0.0
         ENDIF
*
         IF .NOT. AGE_P2 - 0
            REPLACE AGE_P2 WITH 0
            REPLACE BA_P2 WITH 0.0
            REPLACE HEIGHT_P2 WITH 0.25
```

```
            REPLACE DIA_P2 WITH 0.065
            REPLACE NO_TREE_P2 WITH 0
            REPLACE VOLUME_P2 WITH 0.0
         ENDIF
*
         IF .NOT. AGE_AP2 = 0
            REPLACE AGE_AP2 WITH 0
            REPLACE BA_AP2 WITH 0.0
            REPLACE HEIGHT_AP2 WITH 0.01
            REPLACE DIA_AP2 WITH 0.008
            REPLACE NO_TRE_AP2 WITH 0
            REPLACE VOLUME_AP2 WITH 0.0
         ENDIF
      ENDIF
*
      IF UPPER(rep9) = "S"
         IF .NOT. AGE_SW2 = 0
            REPLACE AGE_SW2 WITH -regyr
            REPLACE BA_SW2 WITH 0.0
            REPLACE HEIGHT_SW2 WITH 0.2
            REPLACE DIA_SW2 WITH 0.02
            REPLACE NO_TRE_SW2 WITH 35000
            REPLACE VOLUME_SW2 WITH 0.0
         ENDIF
*
         IF .NOT. AGE_SB2 = 0
            REPLACE AGE_SB2 WITH 0
            REPLACE BA_SB2 WITH 0.0
            REPLACE HEIGHT_SB2 WITH 0.35
            REPLACE DIA_SB2 WITH 0.003
            REPLACE NO_TRE_SB2 WITH 0
            REPLACE VOLUME_SB2 WITH 0.0
         ENDIF
*
         IF .NOT. AGE_P2 = 0
            REPLACE AGE_P2 WITH 0
            REPLACE BA_P2 WITH 0.0
            REPLACE HEIGHT_P2 WITH 0.25
            REPLACE DIA_P2 WITH 0.065
            REPLACE NO_TREE_P2 WITH 0
            REPLACE VOLUME_P2 WITH 0.0
         ENDIF
*
         IF .NOT. AGE_AP2 = 0
            REPLACE AGE_AP2 WITH 0
            REPLACE BA_AP2 WITH 0.0
            REPLACE HEIGHT_AP2 WITH 0.01
            REPLACE DIA_AP2 WITH 0.008
            REPLACE NO_TRE_AP2 WITH 0
            REPLACE VOLUME_AP2 WITH 0.0
         ENDIF
      ENDIF
*
   CASE UPPER(rep8) = "P"
*
      IF UPPER(rep9) = "P"
         IF .NOT. AGE_SW2 = 0
            REPLACE AGE_SW2 WITH 0
            REPLACE BA_SW2 WITH 0.0
            REPLACE HEIGHT_SW2 WITH 0.2
            REPLACE DIA_SW2 WITH 0.02
            REPLACE NO_TRE_SW2 WITH 0
            REPLACE VOLUME_SW2 WITH 0.0
         ENDIF
*
```

```
         IF .NOT. AGE_SB2 = 0
            REPLACE AGE_SB2 WITH 0
            REPLACE BA_SB2 WITH 0.0
            REPLACE HEIGHT_SB2 WITH 0.35
            REPLACE DIA_SB2 WITH 0.003
            REPLACE NO_TRE_SB2 WITH 0
            REPLACE VOLUME_SB2 WITH 0.0
         ENDIF
*
         IF .NOT. AGE_P2 = 0
            REPLACE AGE_P2 WITH -regyr + 3
            DO CASE
               CASE -regyr+3 < 1
                  REPLACE BA_P2 WITH 0.0
                  REPLACE HEIGHT_P2 WITH 0.2
                  REPLACE DIA_P2 WITH 0.065
                  REPLACE NO_TREE_P2 WITH 150000
                  REPLACE VOLUME_P2 WITH 0.0
               CASE -regyr+3 = 1
                  REPLACE BA_P2 WITH 0.0
                  REPLACE HEIGHT_P2 WITH 0.2
                  REPLACE DIA_P2 WITH 0.065
                  REPLACE NO_TREE_P2 WITH 150000
                  REPLACE VOLUME_P2 WITH 0.0
               CASE -regyr+3 = 2
                  REPLACE BA_P2 WITH 0.0
                  REPLACE HEIGHT_P2 WITH 0.22
                  REPLACE DIA_P2 WITH 0.08
                  REPLACE NO_TREE_P2 WITH 143183
                  REPLACE VOLUME_P2 WITH 0.0
               CASE -regyr+3 = 3
                  REPLACE BA_P2 WITH 0.0
                  REPLACE HEIGHT_P2 WITH 0.25
                  REPLACE DIA_P2 WITH 0.1
                  REPLACE NO_TREE_P2 WITH 136593
                  REPLACE VOLUME_P2 WITH 0.0
            ENDCASE
         ENDIF
*
         IF .NOT. AGE_AP2 = 0
            REPLACE AGE_AP2 WITH 0
            REPLACE BA_AP2 WITH 0.0
            REPLACE HEIGHT_AP2 WITH 0.01
            REPLACE DIA_AP2 WITH 0.008
            REPLACE NO_TRE_AP2 WITH 0
            REPLACE VOLUME_AP2 WITH 0.0
         ENDIF
      ENDIF
      IF UPPER(rep9) = "S"
         IF .NOT. AGE_SW2 = 0
            REPLACE AGE_SW2 WITH 0
            REPLACE BA_SW2 WITH 0.0
            REPLACE HEIGHT_SW2 WITH 0.2
            REPLACE DIA_SW2 WITH 0.02
            REPLACE NO_TRE_SW2 WITH 0
            REPLACE VOLUME_SW2 WITH 0.0
         ENDIF
*
         IF .NOT. AGE_SB2 = 0
            REPLACE AGE_SB2 WITH 0
            REPLACE BA_SB2 WITH 0.0
            REPLACE HEIGHT_SB2 WITH 0.35
            REPLACE DIA_SB2 WITH 0.003
            REPLACE NO_TRE_SB2 WITH 0
            REPLACE VOLUME_SB2 WITH 0.0
```

```
         ENDIF
*
         IF .NOT. AGE_P2 = 0
            REPLACE AGE_P2 WITH  regyr
            REPLACE BA_P2 WITH 0.0
            REPLACE HEIGHT_P2 WITH 0.2
            REPLACE DIA_P2 WITH 0.065
            REPLACE NO_TREE_P2 WITH 15000.
            REPLACE VOLUME_P2 WITH 0.0
         ENDIF
*
         IF .NOT. AGE_AP2 = 0
            REPLACE AGE_AP2 WITH 0
            REPLACE BA_AP2 WITH 0.0
            REPLACE HEIGHT_AP2 WITH 0.01
            REPLACE DIA_AP2 WITH 0.008
            REPLACE NO_TRE_AP2 WITH 0
            REPLACE VOLUME_AP2 WITH 0.0
         ENDIF
      ENDIF
ENDCASE
Subroutine GROWTH
* GROW.PRG
SET TALK OFF
SET PROCEDURE TO Procedfl
* initialize memory variables
rep7 = SPACE(1)
rep8 = SPACE(1)
rep9 = SPACE(1)
CLEAR
@ 10,15 SAY "**************************************************"
@ 11,15 SAY "*                      GROW                      *"
@ 12,15 SAY "**************************************************"
?
ACCEPT "Do you want to simulate resource change over time "+;
       "now? (Y/N)" TO rep7
IF UPPER(rep7) = "Y"
   DO WHILE .T.
      CLEAR
      ACCEPT "Do you want to harvest forest during the simulation? "+;
         "(Y/N)" TO rep8
      ACCEPT "Do you want to regenerate forest during the "+;
         simulation?" " (Y/N)" TO rep9
      INPUT "Please enter the number of years you want to simulate."
         TO years
      STORE 0 TO t
      DO WHILE t <= years
         GO TOP
         DO WHILE .NOT. EOF()
            DO CASE
               CASE SITE = "G"
                  DO Growgsw
                  DO Growgsw2
                  DO Growgsb
                  DO Growgsb2
                  DO Growgp
                  DO Growgp2
                  DO Growgap
                  DO Growgap2
               CASE SITE = "M"
                  DO Growmsw
                  DO Growmsw2
                  DO Growmsb
                  DO Growmsb2
                  DO Growmp
```

```
                  DO Growmp2
                  DO Growmap
                  DO Growmap2
               CASE SITE = "F"
                  DO Growfsw
                  DO Growfsw2
                  DO Growfsb
                  DO Growfsb2
                  DO Growfp
                  DO Growfp2
                  DO Growfap
                  DO Growfap2
            ENDCASE
            DO Beauty
            DO Hunting
            DO Water
            DO Grawld
            IF UPPER(rep20) = "Y"
               DO Thinning
            ENDIF
            SKIP
         ENDDO
         DO Report
         IF UPPER(rep8) = "Y"
            DO Harvest
         ENDIF
         IF UPPER(rep9) = "Y"
            DO Regenerate
         ENDIF
         t = t + 1
      ENDDO
      CLEAR
      rep10 = " "
      DO WHILE .NOT. rep10$"YyNn"
         @ 10,15 SAY "Do you want to do more simulation? (Y/N)"    ^M
                  GET rep10
         READ
      ENDDO
      IF UPPER(rep10) = "N"
         RELEASE t
         EXIT
      ENDIF
   ENDDO
ENDIF
SET TALK ON
RETURN
*
* GROWGSW.PRG
*initialize memory variables
STORE deltat TO deltT
DO CASE
   CASE NO_TREE_SW > 0 .AND. AGE_SW > 0
      DO CASE
         CASE HEIGHT_SW <= 5.9
            Dhsw = -0.019808*HEIGHT_SW**2+0.1695*HEIGHT_SW
         CASE HEIGHT_SW > 5.9 .AND. HEIGHT_SW <= 24.4
            Dhsw = -0.002494*HEIGHT_SW**2+0.067196*HEIGHT_SW
         CASE HEIGHT_SW > 24.4 .AND. HEIGHT_SW < 34.3
            Dhsw = -0.000398*HEIGHT_SW**2+0.015893*HEIGHT_SW
         CASE HEIGHT_SW >= 34.3
            Dhsw = 0.075
      ENDCASE
      hsw = HEIGHT_SW+deltT*Dhsw
      IF UPPER(rep20) = "Y" .AND. t > rot .AND. AGE_SW >= thinyr
         tp = 1/thinper
```

```
   ELSE
      tp - 1
   ENDIF

   DO CASE
      CASE DIA_SW <- 2.8
         Ddsw = (-0.090478*DIA_SW**2+0.325874*DIA_SW)*tp
      CASE DIA_SW > 2.8 .AND. DIA_SW <- 22.9
         Ddsw = (-0.002993*DIA_SW**2+0.074792*DIA_SW)*tp
      CASE DIA_SW > 22.9 .AND. DIA_SW < 34.0
         Ddsw = (-0.000295*DIA_SW**2+0.012879*DIA_SW)*tp
      CASE DIA_SW >- 34.0
         Ddsw = 0.096*tp
   ENDCASE

   dsw - DIA_SW+deltT*Ddsw
   Dnsw - -NO_TREE_SW/(1+EXP(3.27+0.0655*dsw-1.529*Ddsw))
   nsw - INT(NO_TREE_SW+deltT*Dnsw)
   basw - 0.00007854*dsw**2*nsw
   fsw - -0.00067*AGE_SW+0.41
   vsw - (fsw*3.14*nsw*hsw*dsw**2)/40000
   REPLACE AGE_SW WITH AGE_SW + deltT
   REPLACE HEIGHT_SW WITH ROUND(hsw,6)
   REPLACE DIA_SW WITH ROUND(dsw,6)
   REPLACE NO_TREE_SW WITH nsw
   REPLACE BA_SW WITH ROUND(basw,2)
   REPLACE F_FACT_SW WITH ROUND(fsw,2)
   REPLACE VOLUME_SW WITH ROUND(vsw,1)

CASE NO_TREE_SW > 0 .AND. AGE_SW < 1

   REPLACE AGE_SW WITH AGE_SW + deltT
   IF AGE_SW > 0
      DO CASE
         CASE HEIGHT_SW <- 5.9
            Dhsw - -0.019808*HEIGHT_SW**2+0.1695*HEIGHT_SW
         CASE HEIGHT_SW > 5.9 .AND. HEIGHT_SW <- 24.4
            Dhsw - -0.002494*HEIGHT_SW**2+0.067196*HEIGHT_SW
         CASE HEIGHT_SW > 24.4 .AND. HEIGHT_SW < 34.3
            Dhsw - -0.000398*HEIGHT_SW**2+0.015893*HEIGHT_SW
         CASE HEIGHT_SW >- 34.3
            Dhsw - 0.075
      ENDCASE
      hsw - HEIGHT_SW+AGE_SW*Dhsw
      IF UPPER(rep20) - "Y" .AND. t > rot .AND. AGE_SW >- thinyr
         tp - 1/thinper
      ELSE
         tp - 1
      ENDIF

      DO CASE
         CASE DIA_SW <- 2.8
            Ddsw = (-0.090478*DIA_SW**2+0.325874*DIA_SW)*tp
         CASE DIA_SW > 2.8 .AND. DIA_SW <- 22.9
            Ddsw = (-0.002993*DIA_SW**2+0.074792*DIA_SW)*tp
         CASE DIA_SW > 22.9 .AND. DIA_SW < 34.0
            Ddsw = (-0.000295*DIA_SW**2+0.012879*DIA_SW)*tp
         CASE DIA_SW >- 34.0
            Ddsw = 0.096*tp
      ENDCASE
      dsw - DIA_SW+AGE_SW*Ddsw
      Dnsw - -NO_TREE_SW/(1+EXP(3.27+0.0655*dsw-1.529*Ddsw))
      nsw - INT(NO_TREE_SW+AGE_SW*Dnsw)
      basw - 0.00007854*dsw**2*nsw
```

```
      fsw = -0.00067*AGE_SW+0.41
      vsw = (fsw*3.14*nsw*hsw*dsw**2)/40000
      REPLACE HEIGHT_SW WITH ROUND(hsw,6)
      REPLACE DIA_SW WITH ROUND(dsw,6)
      REPLACE NO_TREE_SW WITH nsw
      REPLACE BA_SW WITH ROUND(basw,2)
      REPLACE F_FACT_SW WITH ROUND(fsw,2)
      REPLACE VOLUME_SW WITH ROUND(vsw,1)
    ENDIF
ENDCASE
* GROWGSW2.PRG
*initialize memory variables
STORE deltat TO deltT
DO CASE
  CASE NO_TRE_SW2 > 0 .AND. AGE_SW2 > 0
    DO CASE
      CASE HEIGHT_SW2 <= 5.9
        Dhsw2 = -0.019808*HEIGHT_SW2**2+0.1695*HEIGHT_SW2
      CASE HEIGHT_SW2 > 5.9 .AND. HEIGHT_SW2 <= 24.4
        Dhsw2 = -0.002494*HEIGHT_SW2**2+0.067196*HEIGHT_SW2
      CASE HEIGHT_SW2 > 24.4 .AND. HEIGHT_SW2 < 34.3
        Dhsw2 = -0.000398*HEIGHT_SW2**2+0.015893*HEIGHT_SW2
      CASE HEIGHT_SW2 >= 34.3
        Dhsw2 = 0.075
    ENDCASE
    hsw2 = HEIGHT_SW2+deltT*Dhsw2
    IF UPPER(rep20) = "Y" .AND. t > rot .AND. AGE_SW2 >= thinv
      tp2 = 1/thinper
    ELSE
      tp2 = 1
    ENDIF
    DO CASE
      CASE DIA_SW2 <= 2.8
        Ddsw2 = (-0.090478*DIA_SW2**2+0.325874*DIA_SW2)*tp2
      CASE DIA_SW2 > 2.8 .AND. DIA_SW2 <= 22.9
        Ddsw2 = (-0.002993*DIA_SW2**2+0.074792*DIA_SW2)*tp2
      CASE DIA_SW2 > 22.9 .AND. DIA_SW2 < 34.0
        Ddsw2 = (-0.000295*DIA_SW2**2+0.012879*DIA_SW2)*tp2
      CASE DIA_SW2 >= 34.0
        Ddsw2 = 0.096*tp2
    ENDCASE
    dsw2 = DIA_SW2+deltT*Ddsw2
    Dnsw2 = -NO_TRE_SW2/(1+EXP(3.27+0.0655*dsw2-1.529*Ddsw2))
    nsw2 = INT(NO_TRE_SW2+deltT*Dnsw2)
    basw2 = 0.00007854*dsw2**2*nsw2
    fsw2 = -0.00067*AGE_SW2+0.41
    vsw2 = (fsw2*3.14*nsw2*hsw2*dsw2**2)/40000
    REPLACE AGE_SW2 WITH AGE_SW2 + deltT
    REPLACE HEIGHT_SW2 WITH ROUND(hsw2,6)
    REPLACE DIA_SW2 WITH ROUND(dsw2,6)
    REPLACE NO_TRE_SW2 WITH nsw2
    REPLACE BA_SW2 WITH ROUND(basw2,2)
    REPLACE F_FACT_SW2 WITH ROUND(fsw2,2)
    REPLACE VOLUME_SW2 WITH ROUND(vsw2,1)
  CASE NO_TRE_SW2 > 0 .AND. AGE_SW2 < 1
    REPLACE AGE_SW2 WITH AGE_SW2 + deltT
    IF AGE_SW2 > 0
      DO CASE
        CASE HEIGHT_SW2 <= 5.9
          Dhsw2 = -0.019808*HEIGHT_SW2**2+0.1695*HEIGHT_SW2
        CASE HEIGHT_SW2 > 5.9 .AND. HEIGHT_SW2 <= 24.4
          Dhsw2 = -0.002494*HEIGHT_SW2**2+0.067196*HEIGHT_SW2
        CASE HEIGHT_SW2 > 24.4 .AND. HEIGHT_SW2 < 34.3
          Dhsw2 = -0.000398*HEIGHT_SW2**2+0.015893*HEIGHT_SW2
        CASE HEIGHT_SW2 >= 34.3
```

```
               Dhsw2 = 0.075
         ENDCASE
         hsw2 = HEIGHT_SW2+AGE_SW2*Dhsw2
         IF UPPER(rep20) = "Y" .AND. t > rot .AND. AGE_SW2 >=thinper
            tp2 = 1/thinper
         ELSE
            tp2 = 1
         ENDIF
         DO CASE
            CASE DIA_SW2 <= 2.8
               Ddsw2 = (-0.090478*DIA_SW2**2+0.325874*DIA_SW2)*tp2
            CASE DIA_SW2 > 2.8 .AND. DIA_SW2 <= 22.9
               Ddsw2 = (-0.002993*DIA_SW2**2+0.074792*DIA_SW2)*tp2
            CASE DIA_SW2 > 22.9 .AND. DIA_SW2 < 34.0
               Ddsw2 = (-0.000295*DIA_SW2**2+0.012879*DIA_SW2)*tp2
            CASE DIA_SW2 >= 34.0
               Ddsw2 = 0.096*tp2
         ENDCASE
         dsw2 = DIA_SW2+AGE_SW2*Ddsw2
         Dnsw2 = -NO_TRE_SW2/(1+EXP(3.27+0.0655*dsw2-1.529*Ddsw2))
         nsw2 = INT(NO_TRE_SW2+AGE_SW2*Dnsw2)
         basw2 = 0.00007854*dsw2**2*nsw2
         fsw2 = -0.00067*AGE_SW2+0.41
         vsw2 = (fsw2*3.14*nsw2*hsw2*dsw2**2)/40000
         REPLACE HEIGHT_SW2 WITH ROUND(hsw2,6)
         REPLACE DIA_SW2 WITH ROUND(dsw2,6)
         REPLACE NO_TRE_SW2 WITH nsw2
         REPLACE BA_SW2 WITH ROUND(basw2,2)
         REPLACE F_FACT_SW2 WITH ROUND(fsw2,2)
         REPLACE VOLUME_SW2 WITH ROUND(vsw2,1)
      ENDIF
ENDCASE
* GROWGSB.PRG
STORE deltat TO deltT
DO CASE
   CASE NO_TREE_SB > 0 .AND. AGE_SB > 0
      IF HEIGHT_SB <= 2.2
         Dhsb = -0.067148*HEIGHT_SB**2+0.206818*HEIGHT_SB
      ENDIF
      IF HEIGHT_SB > 2.2 .AND. HEIGHT_SB <= 16.9
         Dhsb = -0.003606*HEIGHT_SB**2+0.067024*HEIGHT_SB
      ENDIF
      IF HEIGHT_SB > 16.9 .AND. HEIGHT_SB < 24.9
         Dhsb = -0.000411*HEIGHT_SB**2+0.012879*HEIGHT_SB
      ENDIF
      IF HEIGHT_SB >= 24.9
         Dhsb = 0.065
      ENDIF
      hsb = HEIGHT_SB+deltT*Dhsb
      IF DIA_SB <= 1.4
         Ddsb = -0.218073*DIA_SB**2+0.398755*DIA_SB
      ENDIF
      IF DIA_SB > 1.4 .AND. DIA_SB <= 14.2
         Ddsb = -0.005279*DIA_SB**2+0.081691*DIA_SB
      ENDIF
      IF DIA_SB > 14.2 .AND. DIA_SB < 22.3
         Ddsb = -0.000411*DIA_SB**2+0.012174*DIA_SB
      ENDIF
      IF DIA_SB >= 22.3
         Ddsb = 0.067
      ENDIF
      dsb = DIA_SB+deltT*Ddsb
      Dnsb = -NO_TREE_SB/(1+EXP(3.245+0.145*dsb-6*Ddsb))
      nsb = INT(NO_TREE_SB+deltT*Dnsb)
      basb = 0.00007854*dsb**2*nsb
```

```
     fsb = -0.00078*AGE_SB+0.41
     vsb = (fsb*3.14*nsb*hsb*dsb**2)/40000
     REPLACE AGE_SB WITH AGE_SB + deltT
     REPLACE HEIGHT_SB WITH ROUND(hsb,6)
     REPLACE DIA_SB WITH ROUND(dsb,6)
     REPLACE NO_TREE_SB WITH nsb
     REPLACE BA_SB WITH ROUND(basb,2)
     REPLACE F_FACT_SB WITH ROUND(fsb,2)
     REPLACE VOLUME_SB WITH ROUND(vsb,1)
  CASE NO_TREE_SB > 0 .AND. AGE_SB < 1
     REPLACE AGE_SB WITH AGE_SB + deltT
     IF AGE_SB > 0
        IF HEIGHT_SB <= 2.2
           Dhsb = -0.067148*HEIGHT_SB**2+0.206818*HEIGHT_SB
        ENDIF
        IF HEIGHT_SB > 2.2 .AND. HEIGHT_SB <= 16.9
           Dhsb = -0.003606*HEIGHT_SB**2+0.067024*HEIGHT_SB
        ENDIF
        IF HEIGHT_SB > 16.9 .AND. HEIGHT_SB < 24.9
           Dhsb = -0.000411*HEIGHT_SB**2+0.012879*HEIGHT_SB
        ENDIF
        IF HEIGHT_SB >= 24.9
           Dhsb = 0.065
        ENDIF
        hsb = HEIGHT_SB+AGE_SB*Dhsb
        IF DIA_SB <= 1.4
           Ddsb = -0.218073*DIA_SB**2+0.398755*DIA_SB
        ENDIF
        IF DIA_SB > 1.4 .AND. DIA_SB <= 14.2
           Ddsb = -0.005279*DIA_SB**2+0.081691*DIA_SB
        ENDIF
        IF DIA_SB >14.2 .AND. DIA_SB < 22.3
           Ddsb = -0.000411*DIA_SB**2+0.012174*DIA_SB
        ENDIF
        IF DIA_SB >= 22.3
           Ddsb = 0.067
        ENDIF
        dsb = DIA_SB+AGE_SB*Ddsb
        Dnsb = -NO_TREE_SB/(1+EXP(3.245+0.145*dsb-6*Ddsb))
        nsb = INT(NO_TREE_SB+AGE_SB*Dnsb)
        basb = 0.00007854*dsb**2*nsb
        fsb = -0.00078*AGE_SB+0.41
        vsb = (fsb*3.14*nsb*hsb*dsb**2)/40000
        REPLACE HEIGHT_SB WITH ROUND(hsb,6)
        REPLACE DIA_SB WITH ROUND(dsb,6)
        REPLACE NO_TREE_SB WITH nsb
        REPLACE BA_SB WITH ROUND(basb,2)
        REPLACE F_FACT_SB WITH ROUND(fsb,2)
        REPLACE VOLUME_SB WITH ROUND(vsb,1)
     ENDIF
ENDCASE
* GROWGSB2.PRG
STORE deltat TO deltT
DO CASE
   CASE NO_TRE_SB2 > 0 .AND. AGE_SB2 > 0
      IF HEIGHT_SB2 <= 2.2
         Dhsb2 = -0.067148*HEIGHT_SB2**2+0.206818*HEIGHT_SB2
      ENDIF
      IF HEIGHT_SB2 > 2.2 .AND. HEIGHT_SB2 <= 16.9
         Dhsb2 = -0.003606*HEIGHT_SB2**2+0.067024*HEIGHT_SB2
      ENDIF
      IF HEIGHT_SB2 > 16.9 .AND. HEIGHT_SB2 < 24.9
         Dhsb2 = -0.000411*HEIGHT_SB2**2+0.012879*HEIGHT_SB2
      ENDIF
      IF HEIGHT_SB2 >= 24.9
```

```
    Dhsb2 = 0.065
  ENDIF
  hsb2 = HEIGHT_SB2+deltT*Dhsb2
  IF DIA_SB2 <= 1.4
    Ddsb2 = -0.218073*DIA_SB2**2+0.398755*DIA_SB2
  ENDIF
  IF DIA_SB2 > 1.4 .AND. DIA_SB2 <= 14.2
    Ddsb2 = -0.005279*DIA_SB2**2+0.081691*DIA_SB2
  ENDIF
  IF DIA_SB2 > 14.2 .AND. DIA_SB2 < 22.3
    Ddsb2 = -0.000411*DIA_SB2**2+0.012174*DIA_SB2
  ENDIF
  IF DIA_SB2 >= 22.3
    Ddsb2 = 0.067
  ENDIF
  dsb2 = DIA_SB2+deltT*Ddsb2
  Dnsb2 = -NO_TRE_SB2/(1+EXP(3.245+0.145*dsb2-6*Ddsb2))
  nsb2 = INT(NO_TRE_SB2+deltT*Dnsb2)
  basb2 = 0.00007854*dsb2**2*nsb2
  fsb2 = -0.00078*AGE_SB2+0.41
  vsb2 = (fsb2*3.14*nsb2*hsb2*dsb2**2)/40000
  REPLACE AGE_SB2 WITH AGE_SB2 + deltT
  REPLACE HEIGHT_SB2 WITH ROUND(hsb2,6)
  REPLACE DIA_SB2 WITH ROUND(dsb2,6)
  REPLACE NO_TRE_SB2 WITH nsb2
  REPLACE BA_SB2 WITH ROUND(basb2,2)
  REPLACE F_FACT_SB2 WITH ROUND(fsb2,2)
  REPLACE VOLUME_SB2 WITH ROUND(vsb2,1)

CASE NO_TRE_SB2 > 0 .AND. AGE_SB2 < 1
  REPLACE AGE_SB2 WITH AGE_SB2 + deltT
  IF AGE_SB2 > 0
    IF HEIGHT_SB2 <= 2.2
      Dhsb2 = -0.067148*HEIGHT_SB2**2+0.206818*HEIGHT_SB2
    ENDIF
    IF HEIGHT_SB2 > 2.2 .AND. HEIGHT_SB2 <= 16.9
      Dhsb2 = -0.003606*HEIGHT_SB2**2+0.067024*HEIGHT_SB2
    ENDIF
    IF HEIGHT_SB2 > 16.9 .AND. HEIGHT_SB2 < 24.9
      Dhsb2 = -0.000411*HEIGHT_SB2**2+0.012879*HEIGHT_SB2
    ENDIF
    IF HEIGHT_SB2 >= 24.9
      Dhsb2 = 0.065
    ENDIF
    hsb2 = HEIGHT_SB2+AGE_SB2*Dhsb2
    IF DIA_SB2 <= 1.4
    Ddsb2 = -0.218073*DIA_SB2**2+0.398755*DIA_SB2
    ENDIF
    IF DIA_SB2 > 1.4 .AND. DIA_SB2 <= 14.2
      Ddsb2 = -0.005279*DIA_SB2**2+0.081691*DIA_SB2
    ENDIF
    IF DIA_SB2 > 14.2 .AND. DIA_SB2 < 22.3
      Ddsb2 = -0.000411*DIA_SB2**2+0.012174*DIA_SB2
    ENDIF
    IF DIA_SB2 >= 22.3
      Ddsb2 = 0.067
    ENDIF
    dsb2 = DIA_SB2+AGE_SB2*Ddsb2
    Dnsb2 = -NO_TRE_SB2/(1+EXP(3.245+0.145*dsb2-6*Ddsb2))
    nsb2 = INT(NO_TRE_SB2+AGE_SB2*Dnsb2)
    basb2 = 0.00007854*dsb2**2*nsb2
    fsb2 = -0.00078*AGE_SB2+0.41
    vsb2 = (fsb2*3.14*nsb2*hsb2*dsb2**2)/40000
    REPLACE HEIGHT_SB2 WITH ROUND(hsb2,6)
    REPLACE DIA_SB2 WITH ROUND(dsb2,6)
```

```
         REPLACE NO_TRE_SB2 WITH nsb2
         REPLACE BA_SB2 WITH ROUND(basb2,2)
         REPLACE F_FACT_SB2 WITH ROUND(fsb2,2)
         REPLACE VOLUME_SB2 WITH ROUND(vsb2,1)
      ENDIF
ENDCASE
* GROWGP.PRG
STORE deltat TO deltT
DO CASE
   CASE NO_TREE_P > 0 .AND. AGE_P > 0
      DO CASE
         CASE HEIGHT_P <= 4.9
            Dhp = -0.016573*HEIGHT_P**2+0.142646*HEIGHT_P
         CASE HEIGHT_P > 4.9 .AND. HEIGHT_P <= 22.7
            Dhp = -0.003187*HEIGHT_P**2+0.076384*HEIGHT_P
         CASE HEIGHT_P > 22.7 .AND. HEIGHT_P < 28.7
            Dhp = -0.000415*HEIGHT_P**2+0.013409*HEIGHT_P
         CASE HEIGHT_P >= 28.7
            Dhp = 0.042
      ENDCASE
         hp = HEIGHT_P+deltT*Dhp
         IF UPPER(rep20) = "Y" .AND. t > rot .AND. AGE_P >= thinyr
            tp = 1/thinper
         ELSE
            tp = 1
         ENDIF
      DO CASE
         CASE DIA_P <= 3.5
            Ddp = (-0.05971*DIA_P**2+0.279986*DIA_P)*tp
         CASE DIA_P > 3.5 .AND. DIA_P <= 22.5
            Ddp = (-0.003276*DIA_P**2+0.079079*DIA_P)*tp
         CASE DIA_P > 22.5 .AND. DIA_P < 32.7
            Ddp = (-0.000261*DIA_P**2+0.011218*DIA_P)*tp
         CASE DIA_P >= 32.7
            Ddp = 0.087*tp
      ENDCASE
         dp = DIA_P+deltT*Ddp
         Dnp = -NO_TREE_P/(1+EXP(3.48+0.07*dp-3.5*Ddp))
         np = INT(NO_TREE_P+deltT*Dnp)
         bap = 0.00007854*dp**2*np
         fp = -0.00047*AGE_P+0.44
         vp = (fp*3.14*np*hp*dp**2)/40000
         REPLACE AGE_P WITH AGE_P + deltT
         REPLACE HEIGHT_P WITH ROUND(hp,6)
         REPLACE DIA_P WITH ROUND(dp,6)
         REPLACE NO_TREE_P WITH np
         REPLACE BA_P WITH ROUND(bap,2)
         REPLACE F_FACT_P WITH ROUND(fp,2)
         REPLACE VOLUME_P WITH ROUND(vp,1)
   CASE NO_TREE_P > 0 .AND. AGE_P < 1
      REPLACE AGE_P WITH AGE_P + deltT
      IF AGE_P > 0
         DO CASE
            CASE HEIGHT_P <= 4.9
               Dhp = -0.016573*HEIGHT_P**2+0.142646*HEIGHT_P
            CASE HEIGHT_P > 4.9 .AND. HEIGHT_P <= 22.7
               Dhp = -0.003187*HEIGHT_P**2+0.076384*HEIGHT_P
            CASE HEIGHT_P > 22.7 .AND. HEIGHT_P < 28.7
               Dhp = -0.000415*HEIGHT_P**2+0.013409*HEIGHT_P
            CASE HEIGHT_P >= 28.7
               Dhp = 0.042
         ENDCASE
         hp = HEIGHT_P+AGE_P*Dhp
         IF UPPER(rep20) = "Y" .AND. t > rot .AND. AGE_P >= thinyr
            tp = 1/thinper
```

```
         ELSE
            tp - 1
         ENDIF
         DO CASE
            CASE DIA_P <- 3.5
               Ddp - (-0.05971*DIA_P**2+0.279986*DIA_P)*tp
            CASE DIA_P > 3.5 .AND. DIA_P <- 22.5
               Ddp - (-0.003276*DIA_P**2+0.079079*DIA_P)*tp
            CASE DIA_P > 22.5 .AND. DIA_P < 32.7
               Ddp - (-0.000261*DIA_P**2+0.011218*DIA_P)*tp
            CASE DIA_P >- 32.7
               Ddp - 0.087*tp
         ENDCASE
         dp - DIA_P+AGE_P*Ddp
         Dnp - -NO_TREE_P/(1+EXP(3.48+0.07*dp-3.5*Ddp))
         np - INT(NO_TREE_P+AGE_P*Dnp)
         bap - 0.00007854*dp**2*np
         fp - -0.00047*AGE_P+0.44
         vp - (fp*3.14*np*hp*dp**2)/40000
         REPLACE HEIGHT_P WITH ROUND(hp,6)
         REPLACE DIA_P WITH ROUND(dp,6)
         REPLACE NO_TREE_P WITH np
         REPLACE BA_P WITH ROUND(bap,2)
         REPLACE F_FACT_P WITH ROUND(fp,2)
         REPLACE VOLUME_P WITH ROUND(vp,1)
      ENDIF
ENDCASE
* GROWGP2.PRG
STORE deltat TO deltT
DO CASE
   CASE NO_TREE_P2 > 0 .AND. AGE_P2 > 0
      DO CASE
         CASE HEIGHT_P2 <- 4.9
            Dhp2 - -0.016573*HEIGHT_P2**2+0.142646*HEIGHT_P2
         CASE HEIGHT_P2 > 4.9 .AND. HEIGHT_P2 <- 22.7
            Dhp2 - -0.003187*HEIGHT_P2**2+0.076384*HEIGHT_P2
         CASE HEIGHT_P2 > 22.7 .AND. HEIGHT_P2 < 28.7
            Dhp2 - -0.000415*HEIGHT_P2**2+0.013409*HEIGHT_P2
         CASE HEIGHT_P2 >- 28.7
            Dhp2 - 0.042
      ENDCASE
      hp2 - HEIGHT_P2+deltT*Dhp2
      IF UPPER(rep20) - "Y" .AND. t > rot .AND. AGE_P2 >- thinyr
         tp2 - 1/thinper
      ELSE
         tp2 - 1
      ENDIF
      DO CASE
         CASE DIA_P2 <- 3.5
            Ddp2 = (-0.05971*DIA_P2**2+0.279986*DIA_P2)*tp2
         CASE DIA_P2 > 3.5 .AND. DIA_P2 <- 22.5
            Ddp2 = (-0.003276*DIA_P2**2+0.079079*DIA_P2)*tp2
         CASE DIA_P2 > 22.5 .AND. DIA_P2 < 32.7
            Ddp2 = (-0.000261*DIA_P2**2+0.011218*DIA_P2)*tp2
         CASE DIA_P2 >- 32.7
            Ddp2 = 0.087*tp2
      ENDCASE
      dp2 - DIA_P2+deltT*Ddp2
      Dnp2 - -NO_TREE_P2/(1+EXP(3.48+0.07*dp2-3.5*Ddp2))
      np2 - INT(NO_TREE_P2+deltT*Dnp2)
      bap2 - 0.00007854*dp2**2*np2
      fp2 - -0.00047*AGE_P2+0.44
      vp2 - (fp2*3.14*np2*hp2*dp2**2)/40000
      REPLACE AGE_P2 WITH AGE_P2 + deltT
      REPLACE HEIGHT_P2 WITH ROUND(hp2,6)
```

```
      REPLACE DIA_P2 WITH ROUND(dp2,6)
      REPLACE NO_TREE_P2 WITH np2
      REPLACE BA_P2 WITH ROUND(bap2,2)
      REPLACE F_FACT_P2 WITH ROUND(fp2,2)
      REPLACE VOLUME_P2 WITH ROUND(vp2,1)
   CASE NO_TREE_P2 > 0 .AND. AGE_P2 < 1
      REPLACE AGE_P2 WITH AGE_P2 + deltT
      IF AGE_P2 > 0
         DO CASE
            CASE HEIGHT_P2 <= 4.9
               Dhp2 = -0.016573*HEIGHT_P2**2+0.142646*HEIGHT_P2
            CASE HEIGHT_P2 <= 22.7
               Dhp2 = -0.003187*HEIGHT_P2**2+0.076384*HEIGHT_P2
            CASE HEIGHT_P2 > 22.7 .AND. HEIGHT_P2 < 28.7
               Dhp2 = -0.000415*HEIGHT_P2**2+0.013409*HEIGHT_P2
            CASE HEIGHT_P2 >= 28.7
               Dhp2 = 0.042
         ENDCASE
         hp2 = HEIGHT_P2+AGE_P2*Dhp2
         IF UPPER(rep20) = "Y" .AND. t > rot .AND. AGE_P2 >= thinyr
            tp2 = 1/thinper
         ELSE
            tp2 = 1
         ENDIF
         DO CASE
            CASE DIA_P2 <= 3.5
               Ddp2 = (-0.05971*DIA_P2**2+0.279986*DIA_P2)*tp2
            CASE DIA_P2 > 3.5 .AND. DIA_P2 <= 22.5
               Ddp2 = (-0.003276*DIA_P2**2+0.079079*DIA_P2)*tp2
            CASE DIA_P2 > 22.5 .AND. DIA_P2 < 32.7
               Ddp2 = (-0.000261*DIA_P2**2+0.011218*DIA_P2)*tp2
            CASE DIA_P2 >= 32.7
               Ddp2 = 0.087*tp2
         ENDCASE
         dp2 = DIA_P2+AGE_P2*Ddp2
         Dnp2 = -NO_TREE_P2/(1+EXP(3.48+0.07*dp2-3.5*Ddp2))
         np2 = INT(NO_TREE_P2+AGE_P2*Dnp2)
         bap2 = 0.00007854*dp2**2*np2
         fp2 = -0.00047*AGE_P2+0.44
         vp2 = (fp2*3.14*np2*hp2*dp2**2)/40000
         REPLACE HEIGHT_P2 WITH ROUND(hp2,6)
         REPLACE DIA_P2 WITH ROUND(dp2,6)
         REPLACE NO_TREE_P2 WITH np2
         REPLACE BA_P2 WITH ROUND(bap2,2)
         REPLACE F_FACT_P2 WITH ROUND(fp2,2)
         REPLACE VOLUME_P2 WITH ROUND(vp2,1)
      ENDIF
ENDCASE
* GROWGAP.PRG
STORE deltat TO deltT
DO CASE
   CASE NO_TREE_AP > 0 .AND. AGE_AP > 0
      DO CASE
         CASE HEIGHT_AP <= 3.5
            Dhap = -0.234533*HEIGHT_AP**2+0.903693*HEIGHT_AP
         CASE HEIGHT_AP > 3.5 .AND. HEIGHT_AP <= 26.0
            Dhap = -0.003130*HEIGHT_AP**2+0.084528*HEIGHT_AP
         CASE HEIGHT_AP > 26.0 .AND. HEIGHT_AP < 31.3
            Dhap = -0.000358*HEIGHT_AP**2+0.012415*HEIGHT_AP
         CASE HEIGHT_AP >= 31.3
            Dhap = 0.036
      ENDCASE
      hap = HEIGHT_AP+deltT*Dhap
      IF UPPER(rep20) = "Y" .AND. t > rot .AND. (AGE_P >= thinyr .
         .OR. AGE_SW >= thinyr)
```

```
        tp = 1/thinper
    ELSE
        tp = 1
    ENDIF
    DO CASE
        CASE DIA_AP <= 2.5
            Ddap =( -0.264923*DIA_AP**2+0.746143*DIA_AP)*tp
        CASE DIA_AP > 2.5 .AND. DIA_AP <= 28.2
            Ddap = (-0.002570*DIA_AP**2+0.077142*DIA_AP)*tp
        CASE DIA_AP > 28.2 .AND. DIA_AP < 38.2
            Ddap = (-0.000253*DIA_AP**2+0.011768*DIA_AP)*tp
        CASE DIA_AP >= 38.2
            Ddap = 0.079*tp
    ENDCASE
    dap = DIA_AP+deltT*Ddap
    Dnap = -NO_TREE_AP/(1+EXP(3.105+0.135*dap-3.8*Ddap))
    nap = INT(NO_TREE_AP+deltT*Dnap)
    baap = 0.00007854*dap**2*nap
    fap = -0.00046*AGE_AP+0.41
    vap = (fap*3.14*nap*hap*dap**2)/40000
    REPLACE AGE_AP WITH AGE_AP + deltT
    REPLACE HEIGHT_AP WITH ROUND(hap,6)
    REPLACE DIA_AP WITH ROUND(dap,6)
    REPLACE NO_TREE_AP WITH nap
    REPLACE BA_AP WITH ROUND(baap,2)
    REPLACE F_FACT_AP WITH ROUND(fap,2)
    REPLACE VOLUME_AP WITH ROUND(vap,1)
CASE NO_TREE_AP > 0 .AND. AGE_AP < 1
    REPLACE AGE_AP WITH AGE_AP + deltT
    IF AGE_AP > 0
    DO CASE
        CASE HEIGHT_AP <= 3.5
            Dhap = -0.234533*HEIGHT_AP**2+0.903693*HEIGHT_AP
        CASE HEIGHT_AP > 3.5 .AND. HEIGHT_AP <= 26.0
            Dhap = -0.003130*HEIGHT_AP**2+0.084528*HEIGHT_AP
        CASE HEIGHT_AP > 26.0 .AND. HEIGHT_AP < 31.3
            Dhap = -0.000358*HEIGHT_AP**2+0.012415*HEIGHT_AP
        CASE HEIGHT_AP >= 31.3
            Dhap = 0.036
    ENDCASE
    hap = HEIGHT_AP+AGE_AP*Dhap
    IF UPPER(rep20) = "Y" .AND. t > rot .AND. (AGE_P >= thinyr ;
        .OR. AGE_SE >= thinyr)
        tp = 1/thinper
    ELSE
        tp = 1
    ENDIF
    DO CASE
        CASE DIA_AP <= 2.5
            Ddap = (-0.264923*DIA_AP**2+0.746143*DIA_AP)*tp
        CASE DIA_AP > 2.5 .AND. DIA_AP <= 28.2
            Ddap = (-0.002570*DIA_AP**2+0.077142*DIA_AP)*tp
        CASE DIA_AP > 28.2 .AND. DIA_AP < 38.2
            Ddap = (-0.000253*DIA_AP**2+0.011768*DIA_AP)*tp
        CASE DIA_AP >= 38.2
            Ddap = 0.079*tp
    ENDCASE
    dap = DIA_AP+AGE_AP*Ddap
    Dnap = -NO_TREE_AP/(1+EXP(3.105+0.135*dap-3.8*Ddap))
    nap = INT(NO_TREE_AP+AGE_AP*Dnap)
    baap = 0.00007854*dap**2*nap
    fap = -0.00046*AGE_AP+0.41
    vap = (fap*3.14*nap*hap*dap**2)/40000
    REPLACE HEIGHT_AP WITH ROUND(hap,6)
    REPLACE DIA_AP WITH ROUND(dap,6)
```

```
        REPLACE NO_TREE_AP WITH nap
        REPLACE BA_AP WITH ROUND(baap,2)
        REPLACE F_FACT_AP WITH ROUND(fap,2)
        REPLACE VOLUME_AP WITH ROUND(vap,1)
    ENDIF
ENDCASE
* GROWGAP2.PRG
STORE deltat TO deltT
DO CASE
    CASE NO_TRE_AP2 > 0 .AND. AGE_AP2 > 0
        DO CASE
            CASE HEIGHT_AP2 <= 3.5
                Dhap2 = -0.234533*HEIGHT_AP2**2+0.903693*HEIGHT_AP2
            CASE HEIGHT_AP2 > 3.5 .AND. HEIGHT_AP2 <= 26.0
                Dhap2 = -0.003130*HEIGHT_AP2**2+0.084528*HEIGHT_AP2
            CASE HEIGHT_AP2 > 26.0 .AND. HEIGHT_AP2 < 31.3
                Dhap2 = -0.000358*HEIGHT_AP2**2+0.012415*HEIGHT_AP2
            CASE HEIGHT_AP2 >= 31.3
                Dhap2 = 0.036
        ENDCASE
        hap2 = HEIGHT_AP2+deltT*Dhap2
        IF UPPER(rep20) = "Y" .AND. t >rot .AND. (AGE_P2 >= thinyr ;
           .OR. AGE_SW2 >= thinyr)
           tp2 = 1/thinper
        ELSE
           tp2 = 1
        ENDIF
        DO CASE
            CASE DIA_AP2 <= 2.5
                Ddap2 = (-0.264923*DIA_AP2**2+0.746143*DIA_AP2)*tp2
            CASE DIA_AP2 > 2.5 .AND. DIA_AP2 <= 28.2
                Ddap2 = (-0.002570*DIA_AP2**2+0.077142*DIA_AP2)*tp2
            CASE DIA_AP2 > 28.2 .AND. DIA_AP2 < 38.2
                Ddap2 = (-0.000253*DIA_AP2**2+0.011768*DIA_AP2)*tp2
            CASE DIA_AP2 >= 38.2
                Ddap2 = 0.079*tp2
        ENDCASE
        dap2 = DIA_AP2+deltT*Ddap2
        Dnap2 = -NO_TRE_AP2/(1+EXP(3.105+0.135*dap2-3.8*Ddap2))
        nap2 = INT(NO_TRE_AP2+deltT*Dnap2)
        baap2 = 0.00007854*dap2**2*nap2
        fap2 = -0.00046*AGE_AP2+0.41
        vap2 = (fap2*3.14*nap2*hap2*dap2**2)/40000
        REPLACE AGE_AP2 WITH AGE_AP2 + deltT
        REPLACE HEIGHT_AP2 WITH ROUND(hap2,6)
        REPLACE DIA_AP2 WITH ROUND(dap2,6)
        REPLACE NO_TRE_AP2 WITH nap2
        REPLACE BA_AP2 WITH ROUND(baap2,2)
        REPLACE F_FACT_AP2 WITH ROUND(fap2,2)
        REPLACE VOLUME_AP2 WITH ROUND(vap2,1)
    CASE NO_TRE_AP2 > 0 .AND. AGE_AP2 < 1
        REPLACE AGE_AP2 WITH AGE_AP2 + deltT
        IF AGE_AP2 > 0
        DO CASE
            CASE HEIGHT_AP2 <= 3.5
                Dhap2 = -0.234533*HEIGHT_AP2**2+0.903693*HEIGHT_AP2
            CASE HEIGHT_AP2 > 3.5 .AND. HEIGHT_AP2 <= 26.0
                Dhap2 = -0.003130*HEIGHT_AP2**2+0.084528*HEIGHT_AP2
            CASE HEIGHT_AP2 > 26.0 .AND. HEIGHT_AP2 < 31.3
                Dhap2 = -0.000358*HEIGHT_AP2**2+0.012415*HEIGHT_AP2
            CASE HEIGHT_AP2 >= 31.3
                Dhap2 = 0.036
        ENDCASE
        hap2 = HEIGHT_AP2+AGE_AP2*Dhap2
        IF UPPER(rep20) = "Y" .AND. t > rot .AND. (AGE_P2 >= thinyr ;
```

```
          OR  AGE_SW2 >= thinyr)
         tp2 = 1/thinper
      ELSE
         tp2 = 1
      ENDIF
      DO CASE
         CASE DIA_AP2 <= 2.5
            Ddap2 = (-0.264923*DIA_AP2**2+0.746143*DIA_AP2)*tp2
         CASE DIA_AP2 > 2.5 .AND. DIA_AP2 <= 28.2
            Ddap2 = (-0.002570*DIA_AP2**2+0.077142*DIA_AP2)*tp2
         CASE DIA_AP2 > 28.2 .AND. DIA_AP2 < 38.2
            Ddap2 = (-0.000253*DIA_AP2**2+0.011768*DIA_AP2)*tp2
         CASE DIA_AP2 >= 38.2
            Ddap2 = 0.079*tp2
      ENDCASE
      dap2 = DIA_AP2+AGE_AP2*Ddap2
      Dnap2 = -NO_TRE_AP2/(1+EXP(3.105+0.135*dap2-3.8*Ddap2))
      nap2 = INT(NO_TRE_AP2+AGE_AP2*Dnap2)
      baap2 = 0.00007854*dap2**2*nap2
      fap2 = -0.00046*AGE_AP2+0.41
      vap2 = (fap2*3.14*nap2*hap2*dap2**2)/40000
      REPLACE HEIGHT_AP2 WITH ROUND(hap2,6)
      REPLACE DIA_AP2 WITH ROUND(dap2,6)
      REPLACE NO_TRE_AP2 WITH nap2
      REPLACE BA_AP2 WITH ROUND(baap2,2)
      REPLACE F_FACT_AP2 WITH ROUND(fap2,2)
      REPLACE VOLUME_AP2 WITH ROUND(vap2,1)
   ENDIF
ENDCASE
* GROWMSW.PRG
STORE deltat TO deltT
DO CASE
   CASE NO_TREE_SW > 0 .AND. AGE_SW > 0
      DO CASE
         CASE HEIGHT_SW <= 6.0
            Dhsw = -0.016873*HEIGHT_SW**2+0.141141*HEIGHT_SW
         CASE HEIGHT_SW > 6.0 .AND. HEIGHT_SW <= 18.0
            Dhsw = -0.002635*HEIGHT_SW**2+0.05585*HEIGHT_SW
         CASE HEIGHT_SW > 18.0 .AND. HEIGHT_SW < 28.5
            Dhsw = -0.000529*HEIGHT_SW**2+0.017859*HEIGHT_SW
         CASE HEIGHT_SW >= 28.5
            Dhsw = 0.079
      ENDCASE
      hsw = HEIGHT_SW+deltT*Dhsw
      IF UPPER(rep20) = "Y" .AND. t > rot .AND. AGE_SW >= thinyr
         tp = 1/thinper
      ELSE
         tp = 1
      ENDIF
      DO CASE
         CASE DIA_SW <= 1.1
            Ddsw = (-0.284323*DIA_SW**2+0.41863*DIA_SW)*tp
         CASE DIA_SW > 1.1 .AND. DIA_SW <= 17.2
            Ddsw = (-0.005682*DIA_SW**2+0.103766*DIA_SW)*tp
         CASE DIA_SW > 17.2 .AND. DIA_SW < 26.9
            Ddsw = (-0.000268*DIA_SW**2+0.010428*DIA_SW)*tp
         CASE DIA_SW >= 26.9
            Ddsw = 0.086*tp
      ENDCASE
      dsw = DIA_SW+deltT*Ddsw
      Dnsw = -NO_TREE_SW/(1+EXP(3.325+0.09*dsw-3.2*Ddsw))
      nsw = INT(NO_TREE_SW+deltT*Dnsw)
      basw = 0.00007854*dsw**2*nsw
      fsw = -0.001*AGE_SW+0.5
      vsw = (fsw*3.14*nsw*hsw*dsw**2)/40000
```

```
      REPLACE AGE_SW WITH AGE_SW + deltT
      REPLACE HEIGHT_SW WITH ROUND(hsw,6)
      REPLACE DIA_SW WITH ROUND(dsw,6)
      REPLACE NO_TREE_SW WITH nsw
      REPLACE BA_SW WITH ROUND(basw,2)
      REPLACE F_FACT_SW WITH ROUND(fsw,2)
      REPLACE VOLUME_SW WITH ROUND(vsw,1)
    CASE NO_TREE_SW > 0 .AND. AGE_SW < 1
       REPLACE AGE_SW WITH AGE_SW +deltT
       IF AGE_SW > 0
          DO CASE
             CASE HEIGHT_SW <= 6.0
                Dhsw = -0.016873*HEIGHT_SW**2+0.141141*HEIGHT_SW
             CASE HEIGHT_SW > 6.0 .AND. HEIGHT_SW <= 18.0
                Dhsw = -0.002635*HEIGHT_SW**2+0.05585*HEIGHT_SW
             CASE HEIGHT_SW > 18.0 .AND. HEIGHT_SW < 28.5
                Dhsw = -0.000529*HEIGHT_SW**2+0.017859*HEIGHT_SW
             CASE HEIGHT_SW >= 28.5
                Dhsw = 0.079
         ENDCASE
         hsw = HEIGHT_SW+deltT*Dhsw
         IF UPPER(rep20) = "Y" .AND. t > rot .AND. AGE_SW <= thinvr
            tp = 1/thinper
         ELSE
            tp = 1
         ENDIF
         DO CASE
            CASE DIA_SW <= 1.1
               Ddsw = (-0.284323*DIA_SW**2+0.41863*DIA_SW)*tp
            CASE DIA_SW > 1.1 .AND. DIA_SW <= 17.2
               Ddsw = (-0.005682*DIA_SW**2+0.103766*DIA_SW)*tp
            CASE DIA_SW > 17.2 .AND. DIA_SW < 26.9
               Ddsw = (-0.000268*DIA_SW**2-0.010428*DIA_SW)*tp
            CASE DIA_SW >= 26.9
               Ddsw = 0.086*tp
         ENDCASE
         dsw = DIA_SW+AGE_SW*Ddsw
         Dnsw = -NO_TREE_SW/(1+EXP(3.325+0.09*dsw-3.2*Ddsw))
         nsw = INT(NO_TREE_SW+AGE_SW*Dnsw)
         basw = 0.00007854*dsw**2*nsw
         fsw = -0.001*AGE_SW+0.5
         vsw = (fsw*3.14*nsw*hsw*dsw**2)/40000
         REPLACE HEIGHT_SW WITH ROUND(hsw,6)
         REPLACE DIA_SW WITH ROUND(dsw,6)
         REPLACE NO_TREE_SW WITH nsw
         REPLACE BA_SW WITH ROUND(basw,2)
         REPLACE F_FACT_SW WITH ROUND(fsw,2)
         REPLACE VOLUME_SW WITH ROUND(vsw,1)
      ENDIF
ENDCASE
 GROWMSW2.PRG
STORE deltat TO deltT
DO CASE
   CASE NO_TRE_SW2 > 0 .AND. AGE_SW2 > 0
      DO CASE
         CASE HEIGHT_SW2 <= 6.0
            Dhsw2 = -0.016873*HEIGHT_SW2**2+0.141141*HEIGHT_SW2
         CASE HEIGHT_SW2 > 6.0 .AND. HEIGHT_SW2 <= 18.0
            Dhsw2 = -0.002635*HEIGHT_SW2**2+0.05585*HEIGHT_SW2
         CASE HEIGHT_SW2 > 18.0 .AND. HEIGHT_SW2 < 28.5
            Dhsw2 = -0.000529*HEIGHT_SW2**2+0.017859*HEIGHT_SW2
         CASE HEIGHT_SW2 >= 28.5
            Dhsw2 = 0.079
      ENDCASE
         hsw2 = HEIGHT_SW2+deltT*Dhsw2
```

```
      IF UPPER(rep20) = "Y" .AND. t > rot .AND. AGE_SW2 >= thinyr
         tp2 = 1/thinper
      ELSE
         tp2 = 1
      ENDIF
   DO CASE
      CASE DIA_SW2 <= 1.1
         Ddsw2 = (-0.284323*DIA_SW2**2+0.418630*DIA_SW2)*tp2
      CASE DIA_SW2 > 1.1 .AND. DIA_SW2 <= 17.2
         Ddsw2 = (-0.005682*DIA_SW2**2+0.103766*DIA_SW2)*tp2
      CASE DIA_SW2 > 17.2 .AND. DIA_SW2 < 26.9
         Ddsw2 = (-0.000268*DIA_SW2**2+0.010428*DIA_SW2)*tp2
      CASE DIA_SW2 >= 26.9
         Ddsw2 = 0.086*tp2
   ENDCASE
   dsw2 = DIA_SW2+deltT*Ddsw2
   Dnsw2 = -NO_TRE_SW2/(1+EXP(3.325+0.09*dsw2-3.2*Ddsw2))
   nsw2 = INT(NO_TRE_SW2+deltT*Dnsw2)
   basw2 = 0.00007854*dsw2**2*nsw2
   fsw2 = -0.001*AGE_SW2+0.5
   vsw2 = (fsw2*3.14*nsw2*hsw2*dsw2**2)/40000
   REPLACE AGE_SW2 WITH AGE_SW2 + deltT
   REPLACE HEIGHT_SW2 WITH ROUND(hsw2,6)
   REPLACE DIA_SW2 WITH ROUND(dsw2,6)
   REPLACE NO_TRE_SW2 WITH nsw2
   REPLACE BA_SW2 WITH ROUND(basw2,2)
   REPLACE F_FACT_SW2 WITH ROUND(fsw2,2)
   REPLACE VOLUME_SW2 WITH ROUND(vsw2,1)
CASE NO_TRE_SW2 > 0 .AND. AGE_SW2 < 1
   REPLACE AGE_SW2 WITH AGE_SW2 +deltT
   IF AGE_SW2 > 0
      DO CASE
         CASE HEIGHT_SW2 <= 6.0
            Dhsw2 = -0.016873*HEIGHT_SW2**2+0.141141*HEIGHT_SW2
         CASE HEIGHT_SW2 > 6.0 .AND. HEIGHT_SW2 <= 18.0
            Dhsw2 = -0.002635*HEIGHT_SW2**2+0.055850*HEIGHT_SW2
         CASE HEIGHT_SW2 > 18.0 .AND. HEIGHT_SW2 < 28.5
            Dhsw2 = -0.000529*HEIGHT_SW2**2+0.017859*HEIGHT_SW2
         CASE HEIGHT_SW2 >= 28.5
            Dhsw2 = 0.079
      ENDCASE
      hsw2 = HEIGHT_SW2+deltT*Dhsw2
      IF UPPER(rep20) = "Y" .AND. t > rot .AND. AGE_SW2 >= thinyr
         tp2 = 1/thinper
      ELSE
         tp2 = 1
      ENDIF
      DO CASE
         CASE DIA_SW2 <= 1.1
            Ddsw2 = (-0.284323*DIA_SW2**2+0.418630*DIA_SW2)*tp2
         CASE DIA_SW2 > 1.1 .AND. DIA_SW2 <= 17.2
            Ddsw2 = (-0.005682*DIA_SW2**2+0.103766*DIA_SW2)*tp2
         CASE DIA_SW2 > 17.2 .AND. DIA_SW2 < 26.9
            Ddsw2 = (-0.000268*DIA_SW2**2+0.010428*DIA_SW2)*tp2
         CASE DIA_SW2 >= 26.9
            Ddsw2 = 0.086*tp2
      ENDCASE
      dsw2 = DIA_SW2+AGE_SW2*Ddsw2
      Dnsw2 = -NO_TRE_SW2/(1+EXP(3.325+0.09*dsw2-3.2*Ddsw2))
      nsw2 = INT(NO_TRE_SW2+AGE_SW2*Dnsw2)
      basw2 = 0.00007854*dsw2**2*nsw2
      fsw2 = -0.001*AGE_SW2+0.5
      vsw2 = (fsw2*3.14*nsw2*hsw2*dsw2**2)/40000
      REPLACE HEIGHT_SW2 WITH ROUND(hsw2,6)
      REPLACE DIA_SW2 WITH ROUND(dsw2,6)
```

```
         REPLACE NO_TRE_SW2 WITH nsw2
         REPLACE BA_SW2 WITH ROUND(basw2,2)
         REPLACE F_FACT_SW2 WITH ROUND(fsw2,2)
         REPLACE VOLUME_SW2 WITH ROUND(vsw2,1)
      ENDIF
ENDCASE
* GROWMSB.PRG
STORE deltat TO deltT
DO CASE
   CASE NO_TREE_SB > 0 .AND. AGE_SB > 0
      IF HEIGHT_SB <= 2.6
         Dhsb = -0.035259*HEIGHT_SB**2+0.135341*HEIGHT_SB
      ENDIF
      IF HEIGHT_SB > 2.6 .AND. HEIGHT_SB <= 10.5
         Dhsb = -0.004154*HEIGHT_SB**2+0.052290*HEIGHT_SB
      ENDIF
      IF HEIGHT_SB > 10.5 .AND. HEIGHT_SB < 17.8
         Dhsb = -0.000633*HEIGHT_SB**2+0.015212*HEIGHT_SB
      ENDIF
      IF HEIGHT_SB >= 17.8
         Dhsb = 0.07
      ENDIF
      hsb = HEIGHT_SB+deltT*Dhsb
      IF DIA_SB <= 0.3
         Ddsb = -0.412500*DIA_SB**2+0.482500*DIA_SB
      ENDIF
      IF DIA_SB > 0.3 .AND. DIA_SB <= 8.8
         Ddsb = -0.007900*DIA_SB**2+0.077900*DIA_SB
      ENDIF
      IF DIA_SB > 8.8 .AND. DIA_SB < 15.0
         Ddsb = -0.000737*DIA_SB**2+0.014439*DIA_SB
      ENDIF
      IF DIA_SB >= 15.0
         Ddsb = 0.05
      ENDIF
      dsb = DIA_SB+deltT*Ddsb
      Dnsb = -NO_TREE_SB/(1+EXP(2.25+0.28*dsb-2.8*Ddsb))
      nsb = INT(NO_TREE_SB+deltT*Dnsb)
      basb = 0.00007854*dsb**2*nsb
      fsb = -0.0012*AGE_SB+0.55
      vsb = (fsb*3.14*nsb*hsb*dsb**2)/40000
      REPLACE AGE_SB WITH AGE_SB + deltT
      REPLACE HEIGHT_SB WITH ROUND(hsb,6)
      REPLACE DIA_SB WITH ROUND(dsb,6)
      REPLACE NO_TREE_SB WITH nsb
      REPLACE BA_SB WITH ROUND(basb,2)
      REPLACE F_FACT_SB WITH ROUND(fsb,2)
      REPLACE VOLUME_SB WITH ROUND(vsb,1)
    CASE NO_TREE_SB > 0 .AND. AGE_SB < 1
      REPLACE AGE_SB WITH AGE_SB + deltT
      IF AGE_SB > 0
         IF HEIGHT_SB <= 2.6
            Dhsb = -0.035259*HEIGHT_SB**2+0.135341*HEIGHT_SB
         ENDIF
         IF HEIGHT_SB > 2.6 .AND. HEIGHT_SB <= 10.5
            Dhsb = -0.004154*HEIGHT_SB**2+0.052290*HEIGHT_SB
         ENDIF
         IF HEIGHT_SB > 10.5 .AND. HEIGHT_SB < 17.8
            Dhsb = -0.000633*HEIGHT_SB**2+0.015212*HEIGHT_SB
         ENDIF
         IF HEIGHT_SB >= 17.8
            Dhsb = 0.07
         ENDIF
         hsb = HEIGHT_SB+deltT*Dhsb
```

```
         IF DIA_SB <= 0.3
            Ddsb = -0.412500*DIA_SB**2+0.482500*DIA_SB
         ENDIF
         IF DIA_SB > 0.3 .AND. DIA_SB <= 8.8
            Ddsb = -0.007900*DIA_SB**2+0.077900*DIA_SB
         ENDIF
         IF DIA_SB > 8.8 .AND. DIA_SB < 15.0
            Ddsb = -0.000737*DIA_SB**2+0.014439*DIA_SB
         ENDIF
         IF DIA_SB >= 15.0
            Ddsb = 0.05
         ENDIF
         dsb = DIA_SB+AGE_SB*Ddsb
         Dnsb = -NO_TREE_SB/(1+EXP(2.25+0.28*dsb-2.8*Ddsb))
         nsb = INT(NO_TREE_SB+AGE_SB*Dnsb)
         basb = 0.00007854*dsb**2*nsb
         fsb = -0.0012*AGE_SB+0.55
         vsb = (fsb*3.14*nsb*hsb*dsb**2)/40000
         REPLACE HEIGHT_SB WITH ROUND(hsb,6)
         REPLACE DIA_SB WITH ROUND(dsb,6)
         REPLACE NO_TREE_SB WITH nsb
         REPLACE BA_SB WITH ROUND(basb,2)
         REPLACE F_FACT_SB WITH ROUND(fsb,2)
         REPLACE VOLUME_SB WITH ROUND(vsb,1)
      ENDIF
ENDCASE
* GROWMSB2.PRG
STORE deltat TO deltT
DO CASE
   CASE NO_TRE_SB2 > 0 .AND. AGE_SB2 > 0
      IF HEIGHT_SB2 <= 2.6
         Dhsb2 = -0.035259*HEIGHT_SB2**2+0.135341*HEIGHT_SB2
      ENDIF
      IF HEIGHT_SB2 > 2.6 .AND. HEIGHT_SB2 <= 10.5
         Dhsb2 = -0.004154*HEIGHT_SB2**2+0.052290*HEIGHT_SB2
      ENDIF
      IF HEIGHT_SB2 > 10.5 .AND. HEIGHT_SB2 < 17.8
         Dhsb2 = -0.000633*HEIGHT_SB2**2+0.015212*HEIGHT_SB2
      ENDIF
      IF HEIGHT_SB2 >=17.8
         Dhsb2 = 0.07
      ENDIF
      hsb2 = HEIGHT_SB2+deltT*Dhsb2
      IF DIA_SB2 <= 0.3
         Ddsb2 = -0.412500*DIA_SB2**2+0.482500*DIA_SB2
      ENDIF
      IF DIA_SB2 > 0.3 .AND. DIA_SB2 <= 8.8
         Ddsb2 = -0.007900*DIA_SB2**2+0.077900*DIA_SB2
      ENDIF
      IF DIA_SB2 > 8.8 .AND. DIA_SB2 < 15.0
         Ddsb2 = -0.000737*DIA_SB2**2+0.014439*DIA_SB2
      ENDIF
      IF DIA_SB2 >= 15.0
         Ddsb2 = 0.05
      ENDIF
      dsb2 = DIA_SB2+deltT*Ddsb2
      Dnsb2 = -NO_TRE_SB2/(1+EXP(2.25+0.28*dsb2-2.8*Ddsb2))
      nsb2 = INT(NO_TRE_SB2+deltT*Dnsb2)
      basb2 = 0.00007854*dsb2**2*nsb2
      fsb2 = -0.0012*AGE_SB2+0.55
      vsb2 = (fsb2*3.14*nsb2*hsb2*dsb2**2)/40000
      REPLACE AGE_SB2 WITH AGE_SB2 + deltT
      REPLACE HEIGHT_SB2 WITH ROUND(hsb2,6)
      REPLACE DIA_SB2 WITH ROUND(dsb2,6)
      REPLACE NO_TRE_SB2 WITH nsb2
```

```
      REPLACE BA_SB2 WITH ROUND(basb2,2)
      REPLACE F_FACT_SB2 WITH ROUND(fsb2,2)
      REPLACE VOLUME_SB2 WITH ROUND(vsb2,1)
   CASE NO_TRE_SB2 > 0 .AND. AGE_SB2 < 1
      REPLACE AGE_SB2 WITH AGE_SB2 + deltT
      IF AGE_SB2 > 0
         IF HEIGHT_SB2 <- 2.6
            Dhsb2 = -0.035259*HEIGHT_SB2**2+0.135341*HEIGHT_SB2
         ENDIF
         IF HEIGHT_SB2 > 2.6 .AND. HEIGHT_SB2 <- 10.5
            Dhsb2 = -0.004154*HEIGHT_SB2**2+0.052290*HEIGHT_SB2
         ENDIF
         IF HEIGHT_SB2 > 10.5 .AND. HEIGHT_SB2 < 17.8
            Dhsb2 = -0.000633*HEIGHT_SB2**2+0.015212*HEIGHT_SB2
         ENDIF
         IF HEIGHT_SB2 >-17.8
            Dhsb2 = 0.07
         ENDIF
         hsb2 - HEIGHT_SB2+deltT*Dhsb2
         IF DIA_SB2 <- 0.3
            Ddsb2 - -0.412500*DIA_SB2**2+0.482500*DIA_SB2
         ENDIF
         IF DIA_SB2 > 0.3 .AND. DIA_SB2 <- 8.8
            Ddsb2 - -0.007900*DIA_SB2**2+0.077900*DIA_SB2
         ENDIF
         IF DIA_SB2 > 8.8 .AND. DIA_SB2 < 15.0
            Ddsb2 - -0.000737*DIA_SB2**2+0.014439*DIA_SB2
         ENDIF
         IF DIA_SB2 >- 15.0
            Ddsb2 - 0.05
         ENDIF
         dsb2 - DIA_SB2+AGE_SB2*Ddsb2
         Dnsb2 - -NO_TRE_SB2/(1+EXP(2.25+0.28*dsb2-2.8*Ddsb2))
         nsb2 - INT(NO_TRE_SB2+AGE_SB2*Dnsb2)
         basb2 - 0.00007854*dsb2**2*nsb2
         fsb2 - -0.0012*AGE_SB2+0.55
         vsb2 - (fsb2*3.14*nsb2*hsb2*dsb2**2)/40000
         REPLACE HEIGHT_SB2 WITH ROUND(hsb2,6)
         REPLACE DIA_SB2 WITH ROUND(dsb2,6)
         REPLACE NO_TRE_SB2 WITH nsb2
         REPLACE BA_SB2 WITH ROUND(basb2,2)
         REPLACE F_FACT_SB2 WITH ROUND(fsb2,2)
         REPLACE VOLUME_SB2 WITH ROUND(vsb2,1)
      ENDIF
ENDCASE

* GROWMP.PRG
STORE deltat TO deltT
DO CASE
   CASE NO_TREE_P > 0 .AND. AGE_P > 0
      DO CASE
         CASE HEIGHT_P <- 3.5
            Dhp - -0.027604*HEIGHT_P**2+0.153986*HEIGHT_P
         CASE HEIGHT_P > 3.5 .AND. HEIGHT_P <- 18.0
            Dhp - -0.003638*HEIGHT_P**2+0.069626*HEIGHT_P
         CASE HEIGHT_P > 18.0 .AND. HEIGHT_P < 23.9
            Dhp - -0.000333*HEIGHT_P**2+0.009891*HEIGHT_P
         CASE HEIGHT_P >- 23.9
            Dhp - 0.046
      ENDCASE
      hp - HEIGHT_P+deltT*Dhp
      IF UPPER(rep20) - "Y" .AND. t > rot .AND. AGE_P >- thinyr
         tp - 1/thinper
      ELSE
```

```
      tp = 1
   ENDIF
   DO CASE
      CASE DIA_P <= 2.2
         Ddp = (-0.05331*DIA_P**2+0.20031*DIA_P)*tp
      CASE DIA_P > 2.2 .AND. DIA_P <= 16.5
         Ddp = (-0.005185*DIA_P**2+0.091066*DIA_P)*tp
      CASE DIA_P > 16.5 .AND. DIA_P < 24.2
         Ddp = (-0.000351*DIA_P**2+0.011258*DIA_P)*tp
      CASE DIA_P >= 24.2
         Ddp = 0.066*tp
   ENDCASE
   dp = DIA_P+deltT*Ddp
   Dnp = -NO_TREE_P/(1+EXP(3.25+0.125*dp-4.73*Ddp))
   np = INT(NO_TREE_P+deltT*Dnp)
   bap = 0.00007854*dp**2*np
   fp = -0.00059*AGE_P+0.493
   vp = (fp*3.14*np*hp*dp**2)/40000
   REPLACE AGE_P WITH AGE_P + deltT
   REPLACE HEIGHT_P WITH ROUND(hp,6)
   REPLACE DIA_P WITH ROUND(dp,6)
   REPLACE NO_TREE_P WITH np
   REPLACE BA_P WITH ROUND(bap,2)
   REPLACE F_FACT_P WITH ROUND(fp,2)
   REPLACE VOLUME_P WITH ROUND(vp,1)
CASE NO_TREE_P > 0 .AND. AGE_P < 1
   REPLACE AGE_P WITH AGE_P + deltT
   IF AGE_P > 0
      DO CASE
         CASE HEIGHT_P <= 3.5
            Dhp = -0.027604*HEIGHT_P**2+0.153986*HEIGHT_P
         CASE HEIGHT_P > 3.5 .AND. HEIGHT_P <= 18.0
            Dhp = -0.003638*HEIGHT_P**2+0.069626*HEIGHT_P
         CASE HEIGHT_P > 18.0 .AND. HEIGHT_P < 23.9
            Dhp = -0.000333*HEIGHT_P**2+0.009891*HEIGHT_P
         CASE HEIGHT_P >= 23.9
            Dhp = 0.046
      ENDCASE
      hp = HEIGHT_P+deltT*Dhp
      IF UPPER(rep20) = "Y" .AND. t > rot .AND. AGE_P >= thinyr
         tp = 1/thinper
      ELSE
         tp = 1
      ENDIF
      DO CASE
         CASE DIA_P <= 2.2
            Ddp = (-0.05331*DIA_P**2+0.20031*DIA_P)*tp
         CASE DIA_P > 2.2 .AND. DIA_P <= 16.5
            Ddp = (-0.005185*DIA_P**2+0.091066*DIA_P)*tp
         CASE DIA_P > 16.5 .AND. DIA_P < 24.2
            Ddp = (-0.000351*DIA_P**2+0.011258*DIA_P)*tp
         CASE DIA_P >= 24.2
            Ddp = 0.066*tp
      ENDCASE
      dp = DIA_P+AGE_P*Ddp
      Dnp = -NO_TREE_P/(1+EXP(3.25+0.125*dp-4.73*Ddp))
      np = INT(NO_TREE_P+AGE_P*Dnp)
      bap = 0.00007854*dp**2*np
      fp = -0.00059*AGE_P+0.493
      vp = (fp*3.14*np*hp*dp**2)/40000
      REPLACE HEIGHT_P WITH ROUND(hp,6)
      REPLACE DIA_P WITH ROUND(dp,6)
      REPLACE NO_TREE_P WITH np
      REPLACE BA_P WITH ROUND(bap,2)
      REPLACE F_FACT_P WITH ROUND(fp,2)
```

```
         REPLACE VOLUME_P WITH ROUND(vp,1)
      ENDIF
ENDCASE
* GROWMP2.PRG
STORE deltat TO deltT
DO CASE
   CASE NO_TREE_P2 > 0 .AND. AGE_P2 > 0
      DO CASE
         CASE HEIGHT_P2 <= 3.5
            Dhp2 = -0.027604*HEIGHT_P2**2+0.153986*HEIGHT_P2
         CASE HEIGHT_P2 > 3.5 .AND. HEIGHT_P2 <= 18.0
            Dhp2 = -0.003638*HEIGHT_P2**2+0.069626*HEIGHT_P2
         CASE HEIGHT_P2 > 18.0 .AND. HEIGHT_P2 < 23.9
            Dhp2 = -0.000333*HEIGHT_P2**2+0.009891*HEIGHT_P2
         CASE HEIGHT_P2 >= 23.9
            Dhp2 = 0.046
      ENDCASE
      hp2 = HEIGHT_P2+deltT*Dhp2
      IF UPPER(rep20) = "Y" .AND. t > rot .AND. AGE_P2 >= thinyr
         tp2 = 1/thinper
      ELSE
         tp2 = 1
      ENDIF
      DO CASE
         CASE DIA_P2 <= 2.2
            Ddp2 = (-0.05331*DIA_P2**2+0.20031*DIA_P2)*tp2
         CASE DIA_P2 > 2.2 .AND. DIA_P2 <= 16.5
            Ddp2 = (-0.005185*DIA_P2**2+0.091066*DIA_P2)*tp2
         CASE DIA_P2 > 16.5 .AND. DIA_P2 < 24.2
            Ddp2 = (-0.000351*DIA_P2**2+0.011258*DIA_P2)*tp2
         CASE DIA_P2 >= 24.2
            Ddp2 = 0.066*tp2
      ENDCASE
      dp2 = DIA_P2+deltT*Ddp2
      Dnp2 = -NO_TREE_P2/(1+EXP(3.25+0.125*dp2-4.73*Ddp2))
      np2 = INT(NO_TREE_P2+deltT*Dnp2)
      bap2 = 0.00007854*dp2**2*np2
      fp2 = -0.00059*AGE_P2+0.493
      vp2 = (fp2*3.14*np2*hp2*dp2**2)/40000
      REPLACE AGE_P2 WITH AGE_P2 + deltT
      REPLACE HEIGHT_P2 WITH ROUND(hp2,6)
      REPLACE DIA_P2 WITH ROUND(dp2,6)
      REPLACE NO_TREE_P2 WITH np2
      REPLACE BA_P2 WITH ROUND(bap2,2)
      REPLACE F_FACT_P2 WITH ROUND(fp2,2)
      REPLACE VOLUME_P2 WITH ROUND(vp2,1)
   CASE NO_TREE_P2 > 0 .AND. AGE_P2 < 1
      REPLACE AGE_P2 WITH AGE_P2 + deltT
      IF AGE_P2 > 0
         DO CASE
            CASE HEIGHT_P2 <= 3.5
               Dhp2 = -0.027604*HEIGHT_P2**2+0.153986*HEIGHT_P2
            CASE HEIGHT_P2 > 3.5 .AND. HEIGHT_P2 <= 18.0
               Dhp2 = -0.003638*HEIGHT_P2**2+0.069626*HEIGHT_P2
            CASE HEIGHT_P2 > 18.0 .AND. HEIGHT_P2 < 23.9
               Dhp2 = -0.000333*HEIGHT_P2**2+0.009891*HEIGHT_P2
            CASE HEIGHT_P2 >= 23.9
               Dhp2 = 0.046
         ENDCASE
         hp2 = HEIGHT_P2+deltT*Dhp2
         IF UPPER(rep20) = "Y" .AND. t > rot .AND. AGE_P2 >= thinyr
            tp2 = 1/thinper
         ELSE
            tp2 = 1
         ENDIF
```

```
            DO CASE
               CASE DIA_P2 <= 2.2
                  Ddp2 = (-0.05331*DIA_P2**2+0.20031*DIA_P2)*tp2
               CASE DIA_P2 > 2.2 .AND. DIA_P2 <= 16.5
                  Ddp2 = (-0.005185*DIA_P2**2+0.091066*DIA_P2)*tp2
               CASE DIA_P2 > 16.5 .AND. DIA_P2 < 24.2
                  Ddp2 = (-0.000351*DIA_P2**2+0.011258*DIA_P2)*tp2
               CASE DIA_P2 >= 24.2
                  Ddp2 = 0.066*tp2
            ENDCASE
            dp2 = DIA_P2+AGE_P2*Ddp2
            Dnp2 = -NO_TREE_P2/(1+EXP(3.25+0.125*dp2-4.73*Ddp2))
            np2 = INT(NO_TREE_P2+AGE_P2*Dnp2)
            bap2 = 0.00007854*dp2**2*np2
            fp2 = -0.00059*AGE_P2+0.493
            vp2 = (fp2*3.14*np2*hp2*dp2**2)/40000
            REPLACE HEIGHT_P2 WITH ROUND(hp2,6)
            REPLACE DIA_P2 WITH ROUND(dp2,6)
            REPLACE NO_TREE_P2 WITH np2
            REPLACE BA_P2 WITH ROUND(bap2,2)
            REPLACE F_FACT_P2 WITH ROUND(fp2,2)
            REPLACE VOLUME_P2 WITH ROUND(vp2,1)
         ENDIF
ENDCASE
* GROWMAP.PRG
STORE deltat TO deltT
DO CASE
   CASE NO_TREE_AP > 0 .AND. AGE_AP > 0
      DO CASE
         CASE HEIGHT_AP <= 2.5
            Dhap = -0.285766*HEIGHT_AP**2+0.807153*HEIGHT_AP
         CASE HEIGHT_AP > 2.5 .AND. HEIGHT_AP <= 21.1
            Dhap = -0.003062*HEIGHT_AP**2+0.069295*HEIGHT_AP
         CASE HEIGHT_AP > 21.1 .AND. HEIGHT_AP < 26.1
            Dhap = -0.000661*HEIGHT_AP**2+0.018569*HEIGHT_AP
         CASE HEIGHT_AP >= 26.1
            Dhap = 0.034
      ENDCASE
      hap = HEIGHT_AP+deltT*Dhap
      IF UPPER(rep20) = "Y" .AND. t > rot .AND. (AGE_P >= thinyr ;
         .OR. AGE_SW >= thinyr)
         tp = 1/thinper
      ELSE
         tp =1
      ENDIF
      DO CASE
         CASE DIA_AP <= 1.6
            Ddap = (-0.403726*DIA_AP**2+0.730745*DIA_AP)*tp
         CASE DIA_AP > 1.6 .AND. DIA_AP <= 21.1
            Ddap = (-0.003920*DIA_AP**2+0.087056*DIA_AP)*tp
         CASE DIA_AP > 21.1 .AND. DIA_AP < 28.3
            Ddap = (-0.000321*DIA_AP**2+0.011042*DIA_AP)*tp
         CASE DIA_AP >= 28.3
            Ddap = 0.055*tp
      ENDCASE
      dap = DIA_AP+deltT*Ddap
      Dnap = -NO_TREE_AP/(1+EXP(3.075+0.16*dap-4.4*Ddap))
      nap = INT(NO_TREE_AP+deltT*Dnap)
      baap = 0.00007854*dap**2*nap
      fap = -0.00053*AGE_AP+0.46
      vap = (fap*3.14*nap*hap*dap**2)/40000
      REPLACE AGE_AP WITH AGE_AP + deltT
      REPLACE HEIGHT_AP WITH ROUND(hap,6)
      REPLACE DIA_AP WITH ROUND(dap,6)
      REPLACE NO_TREE_AP WITH nap
```

```
      REPLACE BA_AP WITH ROUND(baap,2)
      REPLACE F_FACT_AP WITH ROUND(fap,2)
      REPLACE VOLUME_AP WITH ROUND(vap,1)

   CASE NO_TREE_AP > 0 .AND. AGE_AP < 1
      REPLACE AGE_AP WITH AGE_AP + deltT
      IF AGE_AP > 0
         DO CASE
            CASE HEIGHT_AP <= 2.6
               Dhap = -0.285766*HEIGHT_AP**2+0.807153*HEIGHT_AP
            CASE HEIGHT_AP > 2.6 .AND. HEIGHT_AP <= 21.1
               Dhap = -0.003062*HEIGHT_AP**2+0.069295*HEIGHT_AP
            CASE HEIGHT_AP > 21.1 .AND. HEIGHT_AP < 26.1
               Dhap = -0.000661*HEIGHT_AP**2+0.018569*HEIGHT_AP
            CASE HEIGHT_AP >= 26.1
               Dhap = 0.034
         ENDCASE
         hap = HEIGHT_AP+deltT*Dhap
         IF UPPER(rep20) = 'Y" .AND. t > rot .AND. (AGE_P >= thinyr .
            .OR. AGE_SW >= thinyr)
            tp = 1/thinper
         ELSE
            tp = 1
         ENDIF
         DO CASE
            CASE DIA_AP <= 1.6
               Ddap = (-0 403726*DIA_AP**2+0.730745*DIA_AP)*tp
            CASE DIA_AP > 1.6 .AND. DIA_AP <= 21.1
               Ddap = (-0.003920*DIA_AP**2+0.087056*DIA_AP)*tp
            CASE DIA_AP > 21.1 .AND. DIA_AP < 28.3
               Ddap = (-0.000321*DIA_AP**2+0.011042*DIA_AP)*tp
            CASE DIA_AP >= 28.3
               Ddap = 0.055*tp
         ENDCASE
         dap = DIA_AP+AGE_AP*Ddap
         Dnap = -NO_TREE_AP/(1+EXP(3.075+0.16*dap-4.4*Ddap))
         nap = INT(NO_TREE_AP+AGE_AP*Dnap)
         baap = 0.00007854*dap**2*nap
         fap = -0.00053*AGE_AP+0.46
         vap = (fap*3.14*nap*hap*dap**2)/40000
         REPLACE HEIGHT_AP WITH ROUND(hap,6)
         REPLACE DIA_AP WITH ROUND(dap,6)
         REPLACE NO_TREE_AP WITH nap
         REPLACE BA_AP WITH ROUND(baap,2)
         REPLACE F_FACT_AP WITH ROUND(fap,2)
         REPLACE VOLUME_AP WITH ROUND(vap,1)
      ENDIF
ENDCASE

* GROWMAP2.PRG
*initialize memory variables
STORE deltat TO deltT
DO CASE
   CASE NO_TRE_AP2 > 0 .AND. AGE_AP2 > 0
      DO CASE
         CASE HEIGHT_AP2 <= 2.5
            Dhap2 = -0.285766*HEIGHT_AP2**2+0.807153*HEIGHT_AP2
         CASE HEIGHT_AP2 > 2.5 .AND. HEIGHT_AP2 <= 21.1
            Dhap2 = -0.003062*HEIGHT_AP2**2+0.069295*HEIGHT_AP2
         CASE HEIGHT_AP2 > 21.1 .AND. HEIGHT_AP2 < 26.1
            Dhap2 = -0.000661*HEIGHT_AP2**2+0.018569*HEIGHT_AP2
         CASE HEIGHT_AP2 >= 26.1
            Dhap2 = 0.034
      ENDCASE
```

```
   hap2 = HEIGHT_AP2+deltT*Dhap2
   IF UPPER(rep20) ="Y" .AND. t > rot .AND. (AGE_P2 >= thinyr ;
      .OR. AGE_SW2 >= thinyr)
      tp2 = 1/thinper
   ELSE
      tp2 = 1
   ENDIF
   DO CASE
      CASE DIA_AP2 <= 1.6
         Ddap2 = (-0.403726*DIA_AP2**2+0.730745*DIA_AP2)*tp2
      CASE DIA_AP2 > 1.6 .AND. DIA_AP2 <= 21.1
         Ddap2 = (-0.003920*DIA_AP2**2+0.087056*DIA_AP2)*tp2
      CASE DIA_AP2 > 21.1 .AND. DIA_AP2 < 28.3
         Ddap2 = (-0.000321*DIA_AP2**2+0.011042*DIA_AP2)*tp2
      CASE DIA_AP2 >= 28.3
         Ddap2 = 0.055*tp2
   ENDCASE
   dap2 = DIA_AP2+deltT*Ddap2
   Dnap2 = -NO_TRE_AP2/(1+EXP(3.075+0.16*dap2-4.4*Ddap2))
   nap2 = INT(NO_TRE_AP2+deltT*Dnap2)
   baap2 = 0.00007854*dap2**2*nap2
   fap2 = -0.00053*AGE_AP2+0.46
   vap2 = (fap2*3.14*nap2*hap2*dap2**2)/40000
   REPLACE AGE_AP2 WITH AGE_AP2 + deltT
   REPLACE HEIGHT_AP2 WITH ROUND(hap2,6)
   REPLACE DIA_AP2 WITH ROUND(dap2,6)
   REPLACE NO_TRE_AP2 WITH nap2
   REPLACE BA_AP2 WITH ROUND(baap2,2)
   REPLACE F_FACT_AP2 WITH ROUND(fap2,2)
   REPLACE VOLUME_AP2 WITH ROUND(vap2,1)
CASE NO_TRE_AP2 > 0 .AND. AGE_AP2 < 1
   REPLACE AGE_AP2 WITH AGE_AP2 + deltT
   IF AGE_AP2 > 0
   DO CASE
      CASE HEIGHT_AP2 <= 2.5
         Dhap2 = -0.285766*HEIGHT_AP2**2+0.807153*HEIGHT_AP2
      CASE HEIGHT_AP2 > 2.5 .AND. HEIGHT_AP2 <= 21.1
         Dhap2 = -0.003062*HEIGHT_AP2**2+0.069295*HEIGHT_AP2
      CASE HEIGHT_AP2 > 21.1 .AND. HEIGHT_AP2 < 26.1
         Dhap2 = -0.000661*HEIGHT_AP2**2+0.018569*HEIGHT_AP2
      CASE HEIGHT_AP2 >= 26.1
         Dhap2 = 0.034
   ENDCASE
   hap2 = HEIGHT_AP2+deltT*Dhap2
   IF UPPER(rep20) = "Y" .AND. t > rot .AND. (AGE_P2 >= thinyr ;
      .OR. AGE_SW2 >= thinyr)
      tp2 = 1/thinper
   ELSE
      tp2 = 1
   ENDIF
   DO CASE
      CASE DIA_AP2 <= 1.6
         Ddap2 = (-0.403726*DIA_AP2**2+0.730745*DIA_AP2)*tp2
      CASE DIA_AP2 > 1.6 .AND. DIA_AP2 <= 21.1
         Ddap2 = (-0.003920*DIA_AP2**2+0.087056*DIA_AP2)*tp2
      CASE DIA_AP2 > 21.1 .AND. DIA_AP2 < 28.3
         Ddap2 = (-0.000321*DIA_AP2**2+0.011042*DIA_AP2)*tp2
      CASE DIA_AP2 >= 28.3
         Ddap2 = 0.055*tp2
   ENDCASE
   dap2 = DIA_AP2+AGE_AP2*Ddap2
   Dnap2 = -NO_TRE_AP2/(1+EXP(3.075+0.16*dap2-4.4*Ddap2))
   nap2 = INT(NO_TRE_AP2+AGE_AP2*Dnap2)
   baap2 = 0.00007854*dap2**2*nap2
   fap2 = -0.00053*AGE_AP2+0.46
```

```
      vap2 = (fap2*3.14*nap2*hap2*dap2**2)/40000
      REPLACE HEIGHT_AP2 WITH ROUND(hap2,6)
      REPLACE DIA_AP2 WITH ROUND(dap2,6)
      REPLACE NO_TRE_AP2 WITH nap2
      REPLACE BA_AP2 WITH ROUND(baap2,2)
      REPLACE F_FACT_AP2 WITH ROUND(fap2,2)
      REPLACE VOLUME_AP2 WITH ROUND(vap2,1)
   ENDIF
ENDCASE
* GROWFSW.PRG
*initialize memory variables
STORE deltat TO deltT
DO CASE
   CASE NO_TREE_SW > 0 .AND. AGE_SW > 0
      DO CASE
         CASE HEIGHT_SW <= 6.0
            Dhsw = -0.008929*HEIGHT_SW**2+0.080843*HEIGHT_SW
         CASE HEIGHT_SW > 6.0 .AND. HEIGHT_SW <= 10.8
            Dhsw = -0.003071*HEIGHT_SW**2+0.045345*HEIGHT_SW
         CASE HEIGHT_SW > 10.8 .AND. HEIGHT_SW < 20.9
            Dhsw = -0.000752*HEIGHT_SW**2+0.020144*HEIGHT_SW
         CASE HEIGHT_SW >= 20.9
            Dhsw = 0.092
      ENDCASE
      hsw = HEIGHT_SW+deltT*Dhsw
      IF UPPER(rep20) = "Y" .AND. t > rot .AND. AGE_SW = thinyr
         tp = 1/thinper
      ELSE
         tp = 1
      ENDIF
      DO CASE
         CASE DIA_SW <= 1.7
            Ddsw = (-0.065187*DIA_SW**2+0.188593*DIA_SW)*tp
         CASE DIA_SW > 1.7 .AND. DIA_SW <= 11.2
            Ddsw = (-0.006927*DIA_SW**2+0.086055*DIA_SW)*tp
         CASE DIA_SW > 11.2 .AND. DIA_SW < 19.7
            Ddsw = (-0.000453*DIA_SW**2+0.013093*DIA_SW)*tp
         CASE DIA_SW >= 19.7
            Ddsw = 0.082*tp
      ENDCASE
      dsw = DIA_SW+deltT*Ddsw
      Dnsw = -NO_TREE_SW/(1+EXP(3.508+0.111*dsw-4*Ddsw))
      nsw = INT(NO_TREE_SW+deltT*Dnsw)
      basw = 0.00007854*dsw**2*nsw
      fsw = -0.0018*AGE_SW+0.708
      vsw = (fsw*3.14*nsw*hsw*dsw**2)/40000
      REPLACE AGE_SW WITH AGE_SW + deltT
      REPLACE HEIGHT_SW WITH ROUND(hsw,6)
      REPLACE DIA_SW WITH ROUND(dsw,6)
      REPLACE NO_TREE_SW WITH nsw
      REPLACE BA_SW WITH ROUND(basw,2)
      REPLACE F_FACT_SW WITH ROUND(fsw,2)
      REPLACE VOLUME_SW WITH ROUND(vsw,1)
   CASE NO_TREE_SW > 0 .AND. AGE_SW < 1
      REPLACE AGE_SW WITH AGE_SW + deltT
      IF AGE_SW > 0
         DO CASE
            CASE HEIGHT_SW <= 6.0
               Dhsw = -0.008929*HEIGHT_SW**2+0.080843*HEIGHT_SW
            CASE HEIGHT_SW > 6.0 .AND. HEIGHT_SW <= 10.8
               Dhsw = -0.003071*HEIGHT_SW**2+0.045345*HEIGHT_SW
            CASE HEIGHT_SW > 10.8 .AND. HEIGHT_SW < 20.9
               Dhsw = -0.000752*HEIGHT_SW**2+0.020144*HEIGHT_SW
            CASE HEIGHT_SW >= 20.9
               Dhsw = 0.092
```

```
         ENDCASE
         hsw = HEIGHT_SW+deltT*Dhsw
         IF UPPER(rep20) ="Y" .AND. t > rot .AND. AGE_SW >= thi
            tp = 1/thinper
         ELSE
            tp = 1
         ENDIF
         DO CASE
            CASE DIA_SW <= 1.7
               Ddsw = (-0.065187*DIA_SW**2+0.188593*DIA_SW)*tp
            CASE DIA_SW > 1.7 .AND. DIA_SW <= 11.2
               Ddsw = (-0.006927*DIA_SW**2+0.086055*DIA_SW)*tp
            CASE DIA_SW > 11.2 .AND. DIA_SW < 19.7
               Ddsw = (-0.000453*DIA_SW**2+0.013093*DIA_SW)*tp
            CASE DIA_SW >= 19.7
               Ddsw = 0.082*tp
         ENDCASE
         dsw = DIA_SW+AGE_SW*Ddsw
         Dnsw = -NO_TREE_SW/(1+EXP(3.508+0.111*dsw-4*Ddsw))
         nsw = INT(NO_TREE_SW+AGE_SW*Dnsw)
         basw = 0.00007854*dsw**2*nsw
         fsw = -0.0018*AGE_SW+0.708
         vsw = (fsw*3.14*nsw*hsw*dsw**2)/40000
         REPLACE HEIGHT_SW WITH ROUND(hsw,6)
         REPLACE DIA_SW WITH ROUND(dsw,6)
         REPLACE NO_TREE_SW WITH nsw
         REPLACE BA_SW WITH ROUND(basw,2)
         REPLACE F_FACT_SW WITH ROUND(fsw,2)
         REPLACE VOLUME_SW WITH ROUND(vsw,1)
      ENDIF
ENDCASE
* GROWFSW2.PRG
*initialize memory variables
STORE deltat TO deltT
DO CASE
   CASE NO_TRE_SW2 > 0 .AND. AGE_SW2 > 0
      DO CASE
         CASE HEIGHT_SW2 <= 6.0
            Dhsw2 = -0.008929*HEIGHT_SW2**2+0.080843*HEIGHT_SW2
         CASE HEIGHT_SW2 > 6.0 .AND. HEIGHT_SW2 <= 10.8
            Dhsw2 = -0.003071*HEIGHT_SW2**2+0.045345*HEIGHT_SW2
         CASE HEIGHT_SW2 > 10.8 .AND. HEIGHT_SW2 < 20.9
            Dhsw2 = -0.000752*HEIGHT_SW2**2+0.020144*HEIGHT_SW2
         CASE HEIGHT_SW2 >= 20.9
            Dhsw2 = 0.092
      ENDCASE
      hsw2 = HEIGHT_SW2+deltT*Dhsw2
      IF UPPER(rep20) = "Y" .AND. t > rot .AND. AGE_SW2 >= thinyr
         tp2 = 1/thinper
      ELSE
         tp2 = 1
      ENDIF
      DO CASE
         CASE DIA_SW2 <= 1.7
            Ddsw2 = (-0.065187*DIA_SW2**2+0.188593*DIA_SW2)*tp2
         CASE DIA_SW2 > 1.7 .AND. DIA_SW2 <= 11.2
            Ddsw2 = (-0.006927*DIA_SW2**2+0.086055*DIA_SW2)*tp2
         CASE DIA_SW2 > 11.2 .AND. DIA_SW2 < 19.7
            Ddsw2 = (-0.000453*DIA_SW2**2+0.013093*DIA_SW2)*tp2
         CASE DIA_SW2 >= 19.7
            Ddsw2 = 0.082*tp2
      ENDCASE
      dsw2 = DIA_SW2+deltT*Ddsw2
      Dnsw2 = -NO_TRE_SW2/(1+EXP(3.508+0.111*dsw2-4*Ddsw2))
```

```
      nsw2 = INT(NO_TRE_SW2+deltT*Dnsw2)
      basw2 = 0.00007854*dsw2**2*nsw2
      fsw2 = -0.0018*AGE_SW2+0.708
      vsw2 = (fsw2*3.14*nsw2*hsw2*dsw2 [illegible]40000
      REPLACE AGE_SW2 WITH AGE_SW2 + deltT
      REPLACE HEIGHT_SW2 WITH ROUND(hsw2,6)
      REPLACE DIA_SW2 WITH ROUND(dsw2,6)
      REPLACE NO_TRE_SW2 WITH nsw2
      REPLACE BA_SW2 WITH ROUND(basw2,2)
      REPLACE F_FACT_SW2 WITH ROUND(fsw2,2)
      REPLACE VOLUMF_SW2 WITH ROUND(vsw2,1)
   CASE NO_TRE_SW2 > 0 .AND. AGE_SW2 < 1
      REPLACE AGE_SW2 WITH AGE_SW2 + deltT
      IF AGE_SW2 > 0
         DO CASE
            CASE HEIGHT_SW2 <= 6.0
               Dhsw2 = -0.008929*HEIGHT_SW2**2+0.080843*HEIGHT_SW2
            CASE HEIGHT_SW2 > 6.0 .AND. HEIGHT_SW2 <= 10.8
               Dhsw2 = -0.003071*HEIGHT_SW2**2+0.045345*HEIGHT_SW2
            CASE HEIGHT_SW2 > 10.8 .AND. HEIGHT_SW2 < 20.9
               Dhsw2 = -0.000752*HEIGHT_SW2**2+0.020144*HEIGHT_SW2
            CASE HEIGHT_SW2 >= 20.9
               Dhsw2 = 0.092
         ENDCASE
         hsw2 = HEIGHT_SW2+deltT*Dhsw2
         IF UPPER(rep20) = "Y" .AND. t > rot .AND. AGE_SW2 >= thinyr
            tp2 = 1/thinper
         ELSE
            tp2 = 1
         ENDIF
         DO CASE
            CASE DIA_SW2 <= 1.7
               Ddsw2 = (-0.065187*DIA_SW2**2+0.188593*DIA_SW2)*tp2
            CASE DIA_SW2 > 1.7 .AND. DIA_SW2 <= 11.2
               Ddsw2 = (-0.006927*DIA_SW2**2+0.086055*DIA_SW2)*tp2
            CASE DIA_SW2 > 11.2 .AND. DIA_SW2 < 19.7
               Ddsw2 = (-0.000453*DIA_SW2**2+0.013093*DIA_SW2)*tp2
            CASE DIA_SW2 >= 19.7
               Ddsw2 = 0.082*tp2
         ENDCASE
         dsw2 = DIA_SW2+AGE_SW2*Ddsw2
         Dnsw2 = -NO_TRE_SW2/(1+EXP(3.508+0.111*dsw2-4*Ddsw2))
         nsw2 = INT(NO_TRE_SW2+AGE_SW2*Dnsw2)
         basw2 = 0.00007854*dsw2**2*nsw2
         fsw2 = -0.0018*AGE_SW2+0.708
         vsw2 = (fsw2*3.14*nsw2*hsw2*dsw2**2)/40000
         REPLACE HEIGHT_SW2 WITH ROUND(hsw2,6)
         REPLACE DIA_SW2 WITH ROUND(dsw2,6)
         REPLACE NO_TRE_SW2 WITH nsw2
         REPLACE BA_SW2 W1.H ROUND(basw2,2)
         REPLACE F_FACT_SW2 WITH ROUND(fsw2,2)
         REPLACE VOLUME_SW2 WITH ROUND(vsw2,1)
      ENDIF
ENDCASE
* GROWFSB.PRG
*initialize memory variables
STORE deltat TO deltT
DO CASE
   CASE NO_TREE_SB > 0 .AND. AGE_SB > 0
      IF HEIGHT_SB <= 3.2
         Dhsb = -0.011786*HEIGHT_SB**2+0.062132*HEIGHT_SB
      ENDIF
      IF HEIGHT_SB > 3.2 .AND. HEIGHT_SB <= 6.3
         Dhsb = -0.004216*HEIGHT_SB**2+0.037834*HEIGHT_SB
```

```
    ENDIF
    IF HEIGHT_SB > 6.3 .AND. HEIGHT_SB < 11.8
       Dhsb = -0.001107*HEIGHT_SB**2+0.018249*HEIGHT_SB
    ENDIF
    IF HEIGHT_SB >= 11.8
       Dhsb = 0.061
    ENDIF
    hsb = HEIGHT_SB+deltT*Dhsb
    IF DIA_SB <= 0.8
       Ddsb = -0.132352*DIA_SB**2+0.206617*DIA_SB
    ENDIF
    IF DIA_SB > 0.8 .AND. DIA_SB <= 4.7
       Ddsb = -0.021582*DIA_SB**2+0.112462*DIA_SB
    ENDIF
    IF DIA_SB > 4.7 .AND. DIA_SB < 9.9
       Ddsb = -0.001068*DIA_SB**2+0.015638*DIA_SB
    ENDIF
    IF DIA_SB >= 9.9
       Ddsb = 0.05
    ENDIF
    dsb = DIA_SB+deltT*Ddsb
    Dnsb = -NO_TREE_SB/(1+EXP(3.5+0.17*dsb-7.45*Ddsb))
    nsb = INT(NO_TREE_SB+deltT*Dnsb)
    basb = 0.00007854*dsb**2*nsb
    fsb = -0.0019*AGE_SB+0.75
    vsb = (fsb*3.14*nsb*hsb*dsb**2)/40000
    REPLACE AGE_SB WITH AGE_SB + deltT
    REPLACE HEIGHT_SB WITH ROUND(hsb,6)
    REPLACE DIA_SB WITH ROUND dsb,6)
    REPLACE NO_TREE_SB WITH nsb
    REPLACE BA_SB WITH ROUND(basb,2)
    REPLACE F_FACT_SB WITH ROUND(fsb,2)
    REPLACE VOLUME_SB WITH ROUND(vsb,1)
CASE NO_TREE_SB > 0 .AND. AGE_SB < 1
    REPLACE AGE_SB WITH AGE_SB + deltT
    IF AGE_SB > 0
       IF HEIGHT_SB <= 3.2
          Dhsb = -0.011786*HEIGHT_SB**2+0.062132*HEIGHT_SB
       ENDIF
       IF HEIGHT_SB > 3.2 .AND. HEIGHT_SB <= 6.3
          Dhsb = -0.004216*HEIGHT_SB**2+0.037834*HEIGHT_SB
       ENDIF
       IF HEIGHT_SB > 6.3 .AND. HEIGHT_SB < 11.8
          Dhsb = -0.001107*HEIGHT_SB**2+0.018249*HEIGHT_SB
       ENDIF
       IF HEIGHT_SB >= 11.8
          Dhsb = 0.061
       ENDIF
       hsb = HEIGHT_SB+deltT*Dhsb
       IF DIA_SB <= 0.8
          Ddsb = -0.132352*DIA_SB**2+0.206617*DIA_SB
       ENDIF
       IF DIA_SB > 0.8 .AND. DIA_SB <= 4.7
          Ddsb = -0.021582*DIA_SB**2+0.112462*DIA_SB
       ENDIF
       IF DIA_SB > 4.7 .AND. DIA_SB < 9.9
          Ddsb = -0.001068*DIA_SB**2+0.015638*DIA_SB
       ENDIF
       IF DIA_SB >= 9.9
          Ddsb = 0.05
       ENDIF
       dsb = DIA_SB+AGE_SB*Ddsb
       Dnsb = -NO_TREE_SB/(1+EXP(3.5+0.17*dsb-7.45*Ddsb))
```

```
         nsb = INT(NO_TREE_SB+AGE_SB*Dnsb)
         basb = 0.00007854*dsb**2*nsb
         fsb = -0.0019*AGE_SB+0.75
         vsb = (fsb*3.14*nsb*hsb*dsb**2)/40000
         REPLACE HEIGHT_SB WITH ROUND(hsb,6)
         REPLACE DIA_SB WITH ROUND(dsb,6)
         REPLACE NO_TREE_SB WITH nsb
         REPLACE BA_SB WITH ROUND(basb,2)
         REPLACE F_FACT_SB WITH ROUND(fsb,2)
         REPLACE VOLUME_SB WITH ROUND(vsb,1)
      ENDIF
ENDCASE
* GROWFSB2.PRG
*initialize memory variables
STORE deltat TO deltT
DO CASE
   CASE NO_TRE_SB2 > 0 .AND. AGE_SB2 > 0
      IF HEIGHT_SB2 <= 3.2
         Dhsb2 = -0.011786*HEIGHT_SB2**2+0.062132*HEIGHT_SB2
      ENDIF
      IF HEIGHT_SB2 > 3.2 .AND. HEIGHT_SB2 <= 6.3
         Dhsb2 = -0.004216*HEIGHT_SB2**2+0.037834*HEIGHT_SB2
      ENDIF
      IF HEIGHT_SB2 > 6.3 .AND. HEIGHT_SB2 < 11.8
         Dhsb2 = -0.001107*HEIGHT_SB2**2+0.018249*HEIGHT_SB2
      ENDIF
      IF HEIGHT_SB2 >= 11.8
         Dhsb2 = 0.061
      ENDIF
      hsb2 = HEIGHT_SB2+deltT*Dhsb2
      IF DIA_SB2 <= 0.8
         Ddsb2 = -0.132352*DIA_SB2**2+0.206617*DIA_SB2
      ENDIF
      IF DIA_SB2 > 0.8 .AND. DIA_SB2 <= 4.7
         Ddsb2 = -0.021582*DIA_SB2**2+0.112462*DIA_SB2
      ENDIF
      IF DIA_SB2 > 4.7 .AND. DIA_SB2 < 9.9
         Ddsb2 = -0.001068*DIA_SB2**2+0.015638*DIA_SB2
      ENDIF
      IF DIA_SB2 >= 9.9
         Ddsb2 = 0.05
      ENDIF
      dsb2 = DIA_SB2+deltT*Ddsb2
      Dnsb2 = -NO_TRE_SB2/(1+EXP(3.5+0.17*dsb2-7.45*Ddsb2))
      nsb2 = INT(NO_TRE_SB2+deltT*Dnsb2)
      basb2 = 0.00007854*dsb2**2*nsb2
      fsb2 = -0.0019*AGE_SB2+0.75
      vsb2 = (fsb2*3.14*nsb2*hsb2*dsb2**2)/40000
      REPLACE AGE_SB2 WITH AGE_SB2 + deltT
      REPLACE HEIGHT_SB2 WITH ROUND(hsb2,6)
      REPLACE DIA_SB2 WITH ROUND(dsb2,6)
      REPLACE NO_TRE_SB2 WITH nsb2
      REPLACE BA_SB2 WITH ROUND(basb2,2)
      REPLACE F_FACT_SB2 WITH ROUND(fsb2,2)
      REPLACE VOLUME_SB2 WITH ROUND(vsb2,1)
   CASE NO_TRE_SB2 > 0 .AND. AGE_SB2 < 1
      REPLACE AGE_SB2 WITH AGE_SB2 + deltT
      IF AGE_SB2 > 0
         IF HEIGHT_SB2 <= 3.2
            Dhsb2 = -0.011786*HEIGHT_SB2**2+0.062132*HEIGHT_SB2
         ENDIF
         IF HEIGHT_SB2 > 3.2 .AND. HEIGHT_SB2 <= 6.3
            Dhsb2 = -0.004216*HEIGHT_SB2**2+0.037834*HEIGHT_SB2
         ENDIF
```

```
         IF HEIGHT_SB2 > 6.3 .AND. HEIGHT_SB2 < 11.8
            Dhsb2 = -0.001107*HEIGHT_SB2**2+0.018249*HEIGHT_SB2
         ENDIF
         IF HEIGHT_SB2 >= 11.8
            Dhsb2 = 0.061
         ENDIF
         hsb2 = HEIGHT_SB2+deltT*Dhsb2
         IF DIA_SB2 <= 0.8
            Ddsb2 = -0.132352*DIA_SB2**2+0.206617*DIA_SB2
         ENDIF
         IF DIA_SB2 > 0.8 .AND. DIA_SB2 <= 4.7
            Ddsb2 = -0.021582*DIA_SB2**2+0.112462*DIA_SB2
         ENDIF
         IF DIA_SB2 > 4.7 .AND. DIA_SB2 < 9.9
            Ddsb2 = -0.001068*DIA_SB2**2+0.015638*DIA_SB2
         ENDIF
         IF DIA_SB2 >= 9.9
            Ddsb2 = 0.05
         ENDIF
         dsb2 = DIA_SB2+AGE_SB2*Ddsb2
         Dnsb2 = -NO_TRE_SB2/(1+EXP(3.5+0.17*dsb2-7.45*Ddsb2))
         nsb2 = INT(NO_TRE_SB2+AGE_SB2*Dnsb2)
         basb2 = 0.00007854*dsb2**2*nsb2
         fsb2 = -0.0019*AGE_SB2+0.75
         vsb2 = (fsb2*3.14*nsb2*hsb2*dsb2**2)/40000
         REPLACE HEIGHT_SB2 WITH ROUND(hsb2,6)
         REPLACE DIA_SB2 WITH ROUND(dsb2,6)
         REPLACE NO_TRE_SB2 WITH nsb2
         REPLACE BA_SB2 WITH ROUND(basb2,2)
         REPLACE F_FACT_SB2 WITH ROUND(fsb2,2)
         REPLACE VOLUME_SB2 WITH ROUND(vsb2,1)
      ENDIF
ENDCASE
* GROWFP.PRG
*initialize memory variables
STORE deltat TO deltT
DO CASE
   CASE NO_TREE_P > 0 .AND. AGE_P > 0
      DO CASE
         CASE HEIGHT_P <= 2.4
            Dhp = -0.023831*HEIGHT_P**2+0.1196*HEIGHT_P
         CASE HEIGHT_P > 2.4 .AND. HEIGHT_P <= 13.2
            Dhp = -0.004976*HEIGHT_P**2+0.073218*HEIGHT_P
         CASE HEIGHT_P > 13.2 .AND. HEIGHT_P < 18.7
            Dhp = -0.000916*HEIGHT_P**2+0.019458*HEIGHT_P
         CASE HEIGHT_P >= 18.7
            Dhp = 0.043
      ENDCASE
      hp = HEIGHT_P+deltT*Dhp
      IF UPPER(rep20) = "Y" .AND. t > rot .AND. AGE_P >= thinyr
         tp = 1/thinper
      ELSE
         tp = 1
      ENDIF
      DO CASE
         CASE DIA_P <= 1.2
            Ddp = (-0.125076*DIA_P**2+0.230411*DIA_P)*tp
         CASE DIA_P > 1.2 .AND. DIA_P <= 11.2
            Ddp = (-0.00641*DIA_P**2+0.078518*DIA_P)*tp
         CASE DIA_P > 11.2 .AND. DIA_P < 17.2
            Ddp = (-0.000532*DIA_P**2+0.012208*DIA_P)*tp
         CASE DIA_P >= 17.2
            Ddp = 0.053*tp
      ENDCASE
```

```
      dp = DIA_P+deltT*Ddp
      Dnp = -NO_TREE_P/(1+EXP(3.108+0.15*dp-5.2*Ddp))
      np = INT(NO_TREE_P+deltT*Dnp)
      bap = 0.00007854*dp**2*np
      fp = -0.0012*AGE_P+0.61
      vp = (fp*3.14*np*hp*dp**2)/40000
      REPLACE AGE_P WITH AGE_P + deltT
      REPLACE HEIGHT_P WITH ROUND(hp,6)
      REPLACE DIA_P WITH ROUND(dp,6)
      REPLACE NO_TREE_P WITH np
      REPLACE BA_P WITH ROUND(bap,2)
      REPLACE F_FACT_P WITH ROUND(fp,2)
      REPLACE VOLUME_P WITH ROUND(vp,1)
   CASE NO_TREE_P > 0 .AND. AGE_P < 1
      REPLACE AGE_P WITH AGE_P + deltT
      IF AGE_P > 0
         DO CASE
            CASE HEIGHT_P <= 2.4
               Dhp = -0.[illegible]3831*HEIGHT_P**2+0.1196*HEIGHT_P
            CASE HEIGHT_P > 2.4 .AND. HEIGHT_P <= 13.2
               Dhp = -0.004976*HEIGHT_P**2+0.07[illegible]GHT_P
            CASE HEIGHT_P > 13.2 .AND. HEIGHT_P [illegible]
               Dhp = -0.000916*HEIGHT_P**2+0.01[illegible]HT_P
            CASE HEIGHT_P >= 18.7
               Dhp = 0.043
         ENDCASE
         hp = HEIGHT_P+deltT*Dhp
         IF UPPER(rep20) = "Y" .AND. t > rot .AND. AGE_P >= thinyr
            tp = 1/thinper
         ELSE
            tp = 1
         ENDIF
         DO CASE
            CASE DIA_P <= 1.2
               Ddp = (-0.125076*DIA_P**2+0.230411*DIA_P)*tp
            CASE DIA_P > 1.2 .AND. DIA_P <= 11.2
               Ddp = (-0.00641*DIA_P**2+0.078518*DIA_P)*tp
            CASE DIA_P > 11.2 .AND. DIA_P < 17.2
               Ddp = (-0.000532*DIA_P**2+0.012208*DIA_P)*tp
            CASE DIA_P >= 17.2
               Ddp = 0.053*tp
         ENDCASE
         dp = DIA_P+AGE_P*Ddp
         Dnp = -NO_TREE_P/(1+EXP(3.108+0.15*dp-5.2*Ddp))
         np = INT(NO_TREE_P+AGE_P*Dnp)
         bap = 0.00007854*dp**2*np
         fp = -0.0012*AGE_P+0.61
         vp = (fp*3.14*np*hp*dp**2)/40000
         REPLACE HEIGHT_P WITH ROUND(hp,6)
         REPLACE DIA_P WITH ROUND(dp,6)
         REPLACE NO_TREE_P WITH np
         REPLACE BA_P WITH ROUND(bap,2)
         REPLACE F_FACT_P WITH ROUND(fp,2)
         REPLACE VOLUME_P WITH ROUND(vp,1)
      ENDIF
ENDCASE
* GROWFP2.PRG
*initialize memory variables
STORE deltat TO deltT
DO CASE
   CASE NO_TREE_P2 > 0 .AND. AGE_P2 > 0
      DO CASE
         CASE HEIGHT_P2 <= 2.4
            Dhp2 = -0.023831*HEIGHT_P2**2+0.1196*HEIGHT_P2
```

```
      CASE HEIGHT_P2 > 2.4 .AND. HEIGHT_P2 <= 13.2
         Dhp2 = -0.004976*HEIGHT_P2**2+0.073218*HEIGHT_P2
      CASE HEIGHT_P2 > 13.2 .AND. HEIGHT_P2 < 18.7
         Dhp2 = -0.000916*HEIGHT_P2**2+0.019458*HEIGHT_P2
      CASE HEIGHT_P2 >= 18.7
         Dhp2 = 0.043
   ENDCASE
   hp2 = HEIGHT_P2+deltT*Dhp2
   IF UPPER(rep20) = "Y" .AND. t > rot .AND. AGE_P2 >= thinyr
      tp2 = 1/thinper
   ELSE
      tp2 = 1
   ENDIF
   DO CASE
      CASE DIA_P2 <= 1.2
         Ddp2 = (-0.125076*DIA_P2**2+0.230411*DIA_P2)*tp2
      CASE DIA_P2 > 1.2 .AND. DIA_P2 <= 11.2
         Ddp2 = (-0.00641*DIA_P2**2+0.078518*DIA_P2)*tp2
      CASE DIA_P2 > 11.2 .AND. DIA_P2 < 17.2
         Ddp2 = (-0.000532*DIA_P2**2+0.012208*DIA_P2)*tp2
      CASE DIA_P2 >= 17.2
         Ddp2 = 0.053*tp2
   ENDCASE
   dp2 = DIA_P2+deltT*Ddp2
   Dnp2 = -NO_TREE_P2/(1+EXP(3.108+0.15*dp2-5.2*Ddp2))
   np2 = INT(NO_TREE_P2+deltT*Dnp2)
   bap2 = 0.00007854*dp2**2*np2
   fp2 = -0.0012*AGE_P2+0.61
   vp2 = (fp2*3.14*np2*hp2*dp2**2)/40000
   REPLACE AGE_P2 WITH AGE_P2 + deltT
   REPLACE HEIGHT_P2 WITH ROUND(hp2,6)
   REPLACE DIA_P2 WITH ROUND(dp2,6)
   REPLACE NO_TREE_P2 WITH np2
   REPLACE BA_P2 WITH ROUND(bap2,2)
   REPLACE F_FACT_P2 WITH ROUND(fp2,2)
   REPLACE VOLUME_P2 WITH ROUND(vp2,1)
CASE NO_TREE_P2 > 0 .AND. AGE_P2 < 1
   REPLACE AGE_P2 WITH AGE_P2 + deltT
   IF AGE_P2 > 0
      DO CASE
         CASE HEIGHT_P2 <= 2.4
            Dhp2 = -0.023831*HEIGHT_P2**2+0.1196*HEIGHT_P2
         CASE HEIGHT_P2 > 2.4 .AND. HEIGHT_P2 <= 13.2
            Dhp2 = -0.004976*HEIGHT_P2**2+0.073218*HEIGHT_P2
         CASE HEIGHT_P2 > 13.2 .AND. HEIGHT_P2 < 18.7
            Dhp2 = -0.000916*HEIGHT_P2**2+0.019458*HEIGHT_P2
         CASE HEIGHT_P2 >= 18.7
            Dhp2 = 0.043
      ENDCASE
      hp2 = HEIGHT_P2+deltT*Dhp2
      IF UPPER(rep20) = "Y" .AND. t > rot .AND. AGE_P2 >= thinyr
         tp2 = 1/thinper
      ELSE
         tp2 = 1
      ENDIF
      DO CASE
         CASE DIA_P2 <= 1.2
            Ddp2 = (-0.125076*DIA_P2**2+0.230411*DIA_P2)*tp2
         CASE DIA_P2 > 1.2 .AND. DIA_P2 <= 11.2
            Ddp2 = (-0.00641*DIA_P2**2+0.078518*DIA_P2)*tp2
         CASE DIA_P2 > 11.2 .AND. DIA_P2 < 17.2
            Ddp2 = (-0.000532*DIA_P2**2+0.012208*DIA_P2)*tp2
         CASE DIA_P2 >= 17.2
            Ddp2 = 0.053*tp2
      ENDCASE
```

```
            dp2 = DIA_P2+AGE_P2*Ddp2
            Dnp2 = -NO_TREE_P2/(1+EXP(3.108+0.15*dp2-5.2*Ddp2))
            np2 = INT(NO_TREE_P2+AGE_P2*Dnp2)
            bap2 = 0.00007854*dp2**2*np2
            fp2 = -0.0012*AGE_P2+0.61
            vp2 = (fp2*3.14*np2*hp2*dp2**2)/40000
            REPLACE HEIGHT_P2 WITH ROUND(hp2,6)
            REPLACE DIA_P2 WITH ROUND(dp2,6)
            REPLACE NO_TREE_P2 WITH np2
            REPLACE BA_P2 WITH ROUND(bap2,2)
            REPLACE F_FACT_P2 WITH ROUND(fp2,2)
            REPLACE VOLUME_P2 WITH ROUND(vp2,1)
         ENDIF
ENDCASE

* GROWFAP.PRG
*initialize memory variables
STORE deltat TO deltT
DO CASE
   CASE NO_TREE_AP > 0 .AND. AGE_AP > 0

      DO CASE
         CASE HEIGHT_AP <= 1.9
            Dhap = -0.503346*HEIGHT_AP**2+1.050334*HEIGHT_AP
         CASE HEIGHT_AP > 1.9 .AND. HEIGHT_AP <= 16.1
            Dhap = -0.004213*HEIGHT_AP**2+0.072034*HEIGHT_AP
         CASE HEIGHT_AP > 16.1 .AND. HEIGHT_AP < 20.5
            Dhap = -0.000567*HEIGHT_AP**2+0.013191*HEIGHT_AP
         CASE HEIGHT_AP >= 20.5
            Dhap = 0.031
      ENDCASE
      hap = HEIGHT_AP+deltT*Dhap
      IF UPPER(rep20) = "Y" .AND. t > rot .AND. (AGE_P >= thinyr ;
         .OR. AGE_SW >= thinyr)
         tp = 1/thinper
      ELSE
         tp = 1
      ENDIF
      DO CASE
         CASE DIA_AP <= 0.9
            Ddap = (-0.879120*DIA_AP**2+0.887912*DIA_AP)*tp
         CASE DIA_AP > 0.9 .AND. DIA_AP <= 15.0
            Ddap = (-0.005895*DIA_AP**2+0.093277*DIA_AP)*tp
         CASE DIA_AP > 15.0 .AND. DIA_AP < 20.4
            Ddap = (-0.000441*DIA_AP**2+0.010966*DIA_AP)*tp
         CASE DIA_AP >= 20.4
            Ddap = 0.04*tp
      ENDCASE
      dap = DIA_AP+deltT*Ddap
      Dnap = -NO_TREE_AP/(1+EXP(3.156+0.16*dap-5.4*Ddap))
      nap = INT(NO_TREE_AP+deltT*Dnap)
      baap = 0.00007854*dap**2*nap
      fap = -0.00068*AGE_AP+0.53
      vap = (fap*3.14*nap*hap*dap**2)/40000
      REPLACE AGE_AP WITH AGE_AP + deltT
      REPLACE HEIGHT_AP WITH ROUND(hap,6)
      REPLACE DIA_AP WITH ROUND(dap,6)
      REPLACE NO_TREE_AP WITH nap
      REPLACE BA_AP WITH ROUND(baap,2)
      REPLACE F_FACT_AP WITH ROUND(fap,2)
      REPLACE VOLUME_AP WITH ROUND(vap,1)
   CASE NO_TREE_AP > 0 .AND. AGE_AP < 1
      REPLACE AGE_AP WITH AGE_AP + deltT
      IF AGE_AP > 0
```

```
         DO CASE
            CASE HEIGHT_AP <= 1.9
               Dhap = -0.503346*HEIGHT_AP**2+1.050334*HEIGHT_AP
            CASE HEIGHT_AP > 1.9 .AND. HEIGHT_AP <= 16.1
               Dhap = -0.004213*HEIGHT_AP**2+0.072034*HEIGHT_AP
            CASE HEIGHT_AP > 16.1 .AND. HEIGHT_AP < 20.5
               Dhap = -0.000567*HEIGHT_AP**2+0.013191*HEIGHT_AP
            CASE HEIGHT_AP >= 20.5
               Dhap = 0.031
         ENDCASE
         hap = HEIGHT_AP+deltT*Dhap
         IF UPPER(rep20) = "Y" .AND. t > rot .AND. (AGE_P2 >= thinyr ;
            .OR. AGE_SW >= thinyr)
            tp = 1/thinper
         ELSE
            tp = 1
         ENDIF
         DO CASE
            CASE DIA_AP <= 0.9
               Ddap = (-0.879120*DIA_AP**2+0.887912*DIA_AP)*tp
            CASE DIA_AP > 0.9 .AND. DIA_AP <= 15.0
               Ddap = (-0.005895*DIA_AP**2+0.093277*DIA_AP)*tp
            CASE DIA_AP > 15.0 .AND. DIA_AP < 20.4
               Ddap = (-0.000441*DIA_AP**2+0.010966*DIA_AP)*tp
            CASE DIA_AP >= 20.4
               Ddap = 0.04*tp
         ENDCASE
         dap = DIA_AP+AGE_AP*Ddap
         Dnap = -NO_TREE_AP/(1+EXP(3.156+0.16*dap-5.4*Ddap))
         nap = INT(NO_TREE_AP+AGE_AP*Dnap)
         baap = 0.00007854*dap**2*nap
         fap = -0.00068*AGE_AP+0.53
         vap = (fap*3.14*nap*hap*dap**2)/40000
         REPLACE HEIGHT_AP WITH ROUND(hap,6)
         REPLACE DIA_AP WITH ROUND(dap,6)
         REPLACE NO_TREE_AP WITH nap
         REPLACE BA_AP WITH ROUND(baap,2)
         REPLACE F_FACT_AP WITH ROUND(fap,2)
         REPLACE VOLUME_AP WITH ROUND(vap,1)
      ENDIF
ENDCASE
* GROWFAP2.PRG
*initialize memory variables
STORE deltat TO deltT
DO CASE
   CASE NO_TRE_AP2 > 0 .AND. AGE_AP2 > 0
      DO CASE
         CASE HEIGHT_AP2 <= 1.9
            Dhap2 = -0.503346*HEIGHT_AP2**2+1.050334*HEIGHT_AP2
         CASE HEIGHT_AP2 > 1.9 .AND. HEIGHT_AP2 <= 16.1
            Dhap2 = -0.004213*HEIGHT_AP2**2+0.072034*HEIGHT_AP2
         CASE HEIGHT_AP2 > 16.1 .AND. HEIGHT_AP2 < 20.5
            Dhap2 = -0.000567*HEIGHT_AP2**2+0.013191*HEIGHT_AP2
         CASE HEIGHT_AP2 >= 20.5
            Dhap2 = 0.031
      ENDCASE
      hap2 = HEIGHT_AP2+deltT*Dhap2
      IF UPPER(rep20) = "Y" .AND. t > rot .AND. (AGE_P >= thinyr ;
         .OR. AGE_SW >= thinyr)
         tp2 = 1/thinper
      ELSE
         tp2 = 1
      ENDIF
      DO CASE
         CASE DIA_AP2 <= 0.9
```

```
            Ddap2 = (-0.879120*DIA_AP2**2+0.887912*DIA_AP2)*tp2
         CASE DIA_AP2 > 0.9 .AND. DIA_AP2 <= 15.0
            Ddap2 = (-0.005895*DIA_AP2**2+0.093277*DIA_AP2)*tp2
         CASE DIA_AP2 > 15.0 .AND. DIA_AP2 < 20.4
            Ddap2 = (-0.000441*DIA_AP2**2+0.010966*DIA_AP2)*tp2
         CASE DIA_AP2 >= 20.4
            Ddap2 = 0.04*tp2
      ENDCASE
      dap2 = DIA_AP2+deltT*Ddap2
      Dnap2 = -NO_TRE_AP2/(1+EXP(3.156+0.16*dap2-5.4*Ddap2))
      nap2 = INT(NO_TRE_AP2+deltT*Dnap2)
      baap2 = 0.00007854*dap2**2*nap2
      fap2 = -0.00068*AGE_AP2+0.53
      vap2 = (fap2*3.14*nap2*hap2*dap2**2)/40000
      REPLACE AGE_AP2 WITH AGE_AP2 + deltT
      REPLACE HEIGHT_AP2 WITH ROUND(hap2,6)
      REPLACE DIA_AP2 WITH ROUND(dap2,6)
      REPLACE NO_TRE_AP2 WITH nap2
      REPLACE BA_AP2 WITH ROUND(baap2,2)
      REPLACE F_FACT_AP2 WITH ROUND(fap2,2)
      REPLACE VOLUME_AP2 WITH ROUND(vap2,1)
   CASE NO_TRE_AP2 > 0 .AND. AGE_AP2 < 1
      REPLACE AGE_AP2 WITH AGE_AP2 + deltT
      IF AGE_AP2 > 0
         DO CASE
            CASE HEIGHT_AP2 <= 1.9
               Dhap2 = -0.503346*HEIGHT_AP2**2+1.050334*HEIGHT_AP2
            CASE HEIGHT_AP2 > 1.9 .AND. HEIGHT_AP2 <= 16.1
               Dhap2 = -0.004213*HEIGHT_AP2**2+0.072034*HEIGHT_AP2
            CASE HEIGHT_AP2 > 16.1 .AND. HEIGHT_AP2 < 20.5
               Dhap2 = -0.000567*HEIGHT_AP2**2+0.013191*HEIGHT_AP2
            CASE HEIGHT_AP2 >= 20.5
               Dhap2 = 0.031
         ENDCASE
         hap2 = HEIGHT_AP2+deltT*Dhap2
         IF UPPER(rep20) = "Y" .AND. t > rot .AND. (AGE_P2 >= thinyr ;
            .OR. AGE_SW >= thinyr)
            tp2 = 1/thinper
         ELSE
            tp2 = 1
         ENDIF
         DO CASE
            CASE DIA_AP2 <= 0.9
               Ddap2 = (-0.879120*DIA_AP2**2+0.887912*DIA_AP2)*tp2
            CASE DIA_AP2 > 0.9 .AND. DIA_AP2 <= 15.0
               Ddap2 = (-0.005895*DIA_AP2**2+0.093277*DIA_AP2)*tp2
            CASE DIA_AP2 > 15.0 .AND. DIA_AP2 < 20.4
               Ddap2 = (-0.000441*DIA_AP2**2+0.010966*DIA_AP2)*tp2
            CASE DIA_AP2 >= 20.4
               Ddap2 = 0.04*tp2
         ENDCASE
         dap2 = DIA_AP2+AGE_AP2*Ddap2
         Dnap2 = -NO_TRE_AP2/(1+EXP(3.156+0.16*dap2-5.4*Ddap2))
         nap2 = INT(NO_TRE_AP2+AGE_AP2*Dnap2)
         baap2 = 0.00007854*dap2**2*nap2
         fap2 = -0.00068*AGE_AP2+0.53
         vap2 = (fap2*3.14*nap2*hap2*dap2**2)/40000
         REPLACE HEIGHT_AP2 WITH ROUND(hap2,6)
         REPLACE DIA_AP2 WITH ROUND(dap2,6)
         REPLACE NO_TRE_AP2 WITH nap2
         REPLACE BA_AP2 WITH ROUND(baap2,2)
         REPLACE F_FACT_AP2 WITH ROUND(fap2,2)
         REPLACE VOLUME_AP2 WITH ROUND(vap2,1)
      ENDIF
ENDCASE
```

Subroutine ESTHETIC

```
* BEAUTY.PRG
DO CASE
   CASE NO_TREE_P > 0 .AND. AGE_P > 0
      IF AGE_P < 15
         mr = 2*AGE_P/15
      ENDIF
      IF AGE_P >= 15 .AND. AGE_P <= 70
         mr = (AGE_P-15)/11+2
      ENDIF
      IF AGE_P > 70
         mr = -(AGE_P-70)/55+7
      ENDIF
      REPLACE BEAUTY WITH ROUND(mr,2)
   CASE NO_TREE_SW > 0 .AND. AGE_SW > 0
      IF AGE_SW < 15
         mr = 2*AGE_P/15
      ENDIF
      IF AGE_SW >= 15 .AND. AGE_SW <= 70
         mr = (AGE_SW-15)/11+2
      ENDIF
      IF AGE_SW > 70
         mr = -(AGE_SW-70)/55+7
      ENDIF
      REPLACE BEAUTY WITH ROUND(mr,2)
   OTHERWISE
      REPLACE BEAUTY WITH 0.00
ENDCASE
DO CASE
   CASE NO_TREE_P2 > 0 .AND. AGE_P2 > 0
      IF AGE_P2 < 15
         mr2 = 2*AGE_P2/15
      ENDIF
      IF AGE_P2 >= 15 .AND. AGE_P2 <= 70
         mr2 = (AGE_P2-15)/11+2
      ENDIF
      IF AGE_P2 > 70
         mr2 = -(AGE_P2-70)/55+7
      ENDIF
      REPLACE BEAUTY2 WITH ROUND(mr2,2)
   CASE NO_TRE_SW2 > 0 .AND. AGE_SW2 > 0
      IF AGE_SW2 < 15
         mr2 = 2*AGE_P2/15
      ENDIF
      IF AGE_SW2 >= 15 .AND. AGE_SW2 <= 70
         mr2 = (AGE_SW2-15)/11+2
      ENDIF
      IF AGE_SW2 > 70
         mr2 = -(AGE_SW2-70)/55+7
      ENDIF
      REPLACE BEAUTY2 WITH ROUND(mr2,2)
   OTHERWISE
      REPLACE BEAUTY2 WITH 0.00
ENDCASE
```

Subroutine HUNTING

```
* HUNTING.PRG
*initialize the memory variables
vm = 0
vc = 0
vm2 = 0
vc2 = 0
*
IF MOOSE >= 0.384
   vm = 10*(MOOSE - 0.384)
ELSE
```

```
   vm = 0
ENDIF
IF MOOSE2 >= 0.384
   vm2 = 10*(MOOSE2 - 0.384)
ELSE
   vm2 = 0
ENDIF
IF CARIBOU >= 0.064
   vc = 10*(CARIBOU - 0.064)
ELSE
   vc = 0
ENDIF
IF CARIBOU2 >= 0.064
   vc2 = 10*(CARIBOU2 - 0.064)
ELSE
   vc2 = 0
ENDIF
REPLACE HUNTINM WITH ROUND(vm,3)
REPLACE HUNTINM2 WITH ROUND(vm2,3)
REPLACE HUNTINC WITH ROUND(vc,3)
REPLACE HUNTINC2 WITH ROUND(vc2,3)
vh = vm + vc
REPLACE HUNTING WITH ROUND(vh,3)
vh2 = vm2 + vc2
REPLACE HUNTING2 WITH ROUND(vh,3)
RETURN
Subroutine WATERYLD
* WATER.PRG
*initialize the memory variables
h = 0
h2 = 0
ba = 0
ba2 = 0
pf = 0
pf2 = 0
et = 0
et2 = 0
q = 0
q2 = 0
IF NO_TREE_SW > 0 .AND. AGE_SW > 0 .OR. NO_TREE_SB > 0 .AND. ;
   AGE_SB > 0 .OR. NO_TREE_P > 0 .AND. AGE_P > 0 .OR. ;
   NO_TREE_AP > 0 .AND. AGE_AP > 0
   h = (HEIGHT_SW+HEIGHT_SB+HEIGHT_P+HEIGHT_AP)/4
   IF NO_TREE_P >= NO_TREE_SW
      ba = (BA_P+BA_AP)
   ELSE
      ba = (BA_SW+BA_AP)
   ENDIF
   DO CASE
      CASE BASIN = "L_SMOKY"
         IF h < 0.071*PRECIP
            pf = 0.843*PRECIP+4.565*h
         ELSE
            pf = 1.157*PRECIP
         ENDIF
         et = 284+0.5*ba+4.84*h
         q = pf - et
         REPLACE WATER WITH INT(q)
      CASE BASIN = "SIMONET"
         IF h < 0.069*PRECIP
            pf = 0.843*PRECIP+4.664*h
         ELSE
            pf = 1.157*PRECIP
         ENDIF
```

```
         et = 245+0.5*ba+5.57*h
         q = pf - et
         REPLACE WATER WITH INT(q)
      CASE BASIN = "BOLTON"
         IF h < 0.075*PRECIP
            pf = 0.843*PRECIP+4.318*h
         ELSE
            pf = 1.157*PRECIP
         ENDIF
         et = 245+0.5*ba+5.56*h
         q = pf - et
         REPLACE WATER WITH INT(q)
  ENDCASE
ELSE
   DO CASE
      CASE BASIN = "L_SMOKY"
         q = PRECIP - 284
         REPLACE WATER WITH INT(q)
      CASE BASIN = "SIMONET"
         q = PRECIP - 245
         REPLACE WATER WITH INT(q)
      CASE BASIN = "BOLTON"
         q = PRECIP - 245
         REPLACE WATER WITH INT(q)
   ENDCASE
ENDIF
IF NO_TRE_SW2 > 0 .AND. AGE_SW2 > 0 .OR. NO_TRE_SB2 > 0 .AND. ;
   AGE_SB2 > 0 .OR. NO_TREE_P2 > 0 .AND. AGE_P2 > 0 .OR. ;
   NO_TRE_AP2 > 0 .AND. AGE_AP2 > 0
   h2 = (HEIGHT_SW2+HEIGHT_SB2+HEIGHT_P2+HEIGHT_AP2)/4
   IF NO_TREE_P2 >= NO_TRE_SW2
      ba2 = (BA_P2+BA_AP2)
   ELSE
      ba2 = (BA_SW2+BA_AP2)
   ENDIF
   DO CASE
      CASE BASIN = "L_SMOKY"
         IF h2 < 0.071*PRECIP
            pf2 = 0.843*PRECIP+4.565*h2
         ELSE
            pf2 = 1.157*PRECIP
         ENDIF
         et2 = 284+0.5*ba2+4.84*h2
         q2 = pf2 - et2
         REPLACE WATER2 WITH INT(q2)
      CASE BASIN = "SIMONET"
         IF h2 < 0.069*PRECIP
            pf2 = 0.843*PRECIP+4.664*h2
         ELSE
            pf2 = 1.157*PRECIP
         ENDIF
         et2 = 245+0.5*ba2+5.57*h2
         q2 = pf2 - et2
         REPLACE WATER2 WITH INT(q2)
      CASE BASIN = "BOLTON"
         IF h2 < 0.075*PRECIP
            pf2 = 0.843*PRECIP+4.318*h2
         ELSE
            pf2 = 1.157*PRECIP
         ENDIF
         et2 = 245+0.5*ba2+5.56*h2
         q2 = pf2 - et2
         REPLACE WATER2 WITH INT(q2)
   ENDCASE
ELSE
```

```
    DO CASE
       CASE BASIN = "L_SMOKY"
          q2 = PRECIP - 284
          REPLACE WATER2 WITH INT(q2)
       CASE BASIN = "SIMONET"
          q2 = PRECIP - 245
          REPLACE WATER2 WITH INT(q2)
       CASE BASIN = "BOLTON"
          q2 = PRECIP - 245
          REPLACE WATER2 WITH INT(q2)
    ENDCASE
 ENDIF
 RETURN
 Subroutine GRAWLD
 * GRAWLD.PRG
 *initialize the memory variables
 ba = 0
 ba2 = 0
 aum = 0
 aum2 = 0
 STORE 0 TO km, km2, dnm, dnm2, nm, nm2, dne, dne2, ne, ne2
 IF NO_TREE_SW > 0 .AND. AGE_SW > 0 .OR. NO_TREE_SB > 0 .AND. ;
    AGE_SB > 0 .OR. NO_TREE_P > 0 .AND. AGE_P > 0 .OR. ;
    NO_TREE_AP > 0 .AND. AGE_AP > 0
    IF NO_TREE_P >= NO_TREE_SW
       ba = (BA_P+BA_AP)
    ELSE
       ba = (BA_SW+BA_AP)
    ENDIF
    IF ba < 100
       aum = (4.94*(0+EXP(ROUND((-0.005*(ba-1.5)**2),4)))+0.99)*64
    ELSE
       aum = 63.36
    ENDIF
    IF ba < 100
       km = (0.768*(EXP(ROUND((-0.0049*(ba-1.5)**2),4)))+0.192)*1.5
    ELSE
       km = 0.192*1.5
    ENDIF
    dnm = 0.5*(MOOSE-0.1*HUNTINM)-(0.5/km)*(MOOSE-0.1*HUNTINM)**2
    nm = (MOOSE-0.1*HUNTINM)+deltat*dnm
    DO CASE
       CASE MAX(AGE_P, AGE_SW) < 30
          kc = 0.032*1.5
       CASE MAX(AGE_P, AGE_SW) >= 30 .AND. MAX(AGE_P, AGE_SW) < 170
          kc = (0.128*((MAX(AGE_P, AGE_SW))^8/(100^8+(MAX(AGE_P,
                AGE_SW))^8))+0.032)*1.5
       CASE MAX(AGE_P, AGE_SW) >= 170
          kc = 0.16*1.5
    ENDCASE
    dnc = 0.5*(CARIBOU-0.1*HUNTINC)-(0.5/kc)*(CARIBOU-0.1*HUNTINC)**2
    nc = (CARIBOU-0.1*HUNTINC)+deltat*dnc
    REPLACE GRAZING WITH ROUND(aum,3)
    REPLACE MOOSE WITH ROUND(nm,3)
    IF nc > 0.032
       REPLACE CARIBOU WITH ROUND(nc,3)
    ELSE
       REPLACE CARIBOU WITH 0.032
    ENDIF
  ELSE
    aum = 316.16
    km = 0.768*1.5
    dnm = 0.5*(MOOSE-0.1*HUNTINM)-(0.5/km)*(MOOSE-0.1*HUNTINM)**2
    nm = (MOOSE-0.1*HUNTINM)+deltat*dnm
    kc = 0.032*1.5
```

```
   dnc = 0.5*(CARIBOU-0.1*HUNTINC)-(0.5/kc)*(CARIBOU-0.1*HUNTINC)**2
   nc = (CARIBOU-0.1*HUNTINC)+deltat*dnc
   REPLACE GRAZING WITH ROUND(aum,3)
   REPLACE MOOSE WITH ROUND(nm,3)
   REPLACE CARIBOU WITH ROUND(nc,3)
ENDIF
IF NO_TRE_SW2 > 0 .AND. AGE_SW2 > 0 .OR. NO_TRE_SB2 > 0 .AND. ;
   AGE_SB2 > 0 .OR. NO_TREE_P2 > 0 .AND. AGE_P2 > 0 .OR. ;
   NO_TRE_AP2 > 0 .AND. AGE_AP2 > 0
   IF NO_TREE_P2 >= NO_TRE_SW2
      ba2 = (BA_P2+BA_AP2)
   ELSE
      ba2 = (BA_SW2+BA_AP2)
   ENDIF
   IF ba2 < 100
      aum2 = (4.94*(0+EXP(ROUND((-0.005*(ba2-1.5)**2),4)))+0.99)*64
   ELSE
      aum2 = 0.99*64
   ENDIF
   IF ba2 < 100
      km2 = (0.768*(0+EXP(ROUND((-0.0049*(ba2-1.5)**2),4)))+0.192)*1.5
   ELSE
      km2 = 0.192*1.5
   ENDIF
   dnm2 = 0.5*(MOOSE2-0.1*HUNTINM2)-(0.5/km2)*(MOOSE2-0.1*;
          HUNTINM2)**2
   nm2 = (MOOSE2-0.1*HUNTINM2)+deltat*dnm2
   DO CASE
      CASE MAX(AGE_P2, AGE_SW2) < 30
         kc2 = 0.032*1.5
      CASE MAX(AGE_P2, AGE_SW2) >= 30 .AND. MAX(AGE_P2, AGE_SW2);
         < 170
         kc2 = (0.128*((MAX(AGE_P2, AGE_SW2))^8/(100^8+(MAX(AGE_P2,
               AGE_SW2))^8))+0.032)*1.5
      CASE MAX(AGE_P2, AGE_SW2) >= 170
         kc2 = 0.16*1.5
   ENDCASE
   dnc2 = 0.5*(CARIBOU2-0.1*HUNTINC2)-(0.5/kc2)*(CARIBOU2-0.1*;
          HUNTINC2)**2
   nc2 = (CARIBOU2-0.1*HUNTINC2)+deltat*dnc2
   REPLACE GRAZING2 WITH ROUND(aum2,3)
   REPLACE MOOSE2 WITH ROUND(nm2,3)
   IF nc2 > 0.032
      REPLACE CARIBOU2 WITH ROUND(nc2,3)
   ELSE
      REPLACE CARIBOU2 WITH 0.032
   ENDIF
ELSE
   aum2 = 4.94*64
   km2 = 0.768*1.5
   dnm2 = 0.5*(MOOSE2-0.1*HUNTINM2)-(0.5/km2)*(MOOSE2-0.1*;
          HUNTINM2)**2
   nm2 = (MOOSE2-0.1*HUNTINM2)+deltat*dnm2
   kc2 = 0.032*1.5
   dnc2 = 0.5*(CARIBOU2-0.1*HUNTINC2)-(0.5/kc2)*(CARIBOU2-0.1*;
          HUNTINC2)**2
   nc2 = (CARIBOU2-0.1*HUNTINC2)+deltat*dnc2
   REPLACE GRAZING2 WITH ROUND(aum2,3)
   REPLACE MOOSE2 WITH ROUND(nm2,3)
   REPLACE CARIBOU2 WITH ROUND(nc2,3)
ENDIF
RETURN
Subroutine THINNING
* THINNING.PRG
```

```
*
DO CASE
    CASE UPPER(rep20) = "Y" .AND. AGE_P = thinyr .AND. thintr <> 0
      NO_TREE_P + NO_TREE_AP
      IF NO_TREE_P >= thintr
         REPLACE NO_TREE_P WITH thintr
         REPLACE NO_TREE_AP WITH 0
         REPLACE BA_AP WITH 0
         REPLACE VOLUME_AP WITH 0
      ELSE
         REPLACE NO_TREE_AP WITH thintr - NO_TREE_P
         bapt = thinper*BA_AP
         REPLACE BA_AP WITH ROUND(bapt,2)
         dapt = 2*SQRT(BA_AP*40000/(NO_TREE_AP*3.14))
         REPLACE DIA_AP WITH ROUND(dapt,4)
         vapt = F_FACT_AP*HEIGHT_AP*BA_AP
         REPLACE VOLUME_AP WITH ROUND(vapt,1)
      ENDIF
      bpt = thinper*BA_P
      REPLACE BA_P WITH ROUND(bpt,2)
      dpt = 2*SQRT(BA_P*40000/(NO_TREE_P*3.14))
      REPLACE DIA_P WITH ROUND(dpt,4)
      vpt = F_FACT_P*HEIGHT_P*BA_P
      REPLACE VOLUME_P WITH ROUND(vpt,1)
    CASE UPPER(rep20) = "Y" .AND. AGE_SW = thinyr .AND. thintr <> 0
      NO_TREE_SW + NO_TREE_AP
      IF NO_TREE_SW >= thintr
         REPLACE NO_TREE_SW WITH thintr
         REPLACE NO_TREE_AP WITH 0
         REPLACE BA_AP WITH 0
         REPLACE VOLUME_AP WITH 0
      ELSE
         REPLACE NO_TREE_AP WITH thintr - NO_TREE_SW
         bapt = thinper*BA_AP
         REPLACE BA_AP WITH ROUND(bapt,2)
         dapt = 2*SQRT(BA_AP*40000/(NO_TREE_AP*3.14))
         REPLACE DIA_AP WITH ROUND(dapt,4)
         vapt = F_FACT_AP*HEIGHT_AP*BA_AP
         REPLACE VOLUME_AP WITH ROUND(vapt,1)
      ENDIF
      bswt = thinper*BA_SW
      REPLACE BA_SW WITH ROUND(bswt,2)
      dswt = 2*SQRT(BA_SW*40000/(NO_TREE_SW*3.14))
      REPLACE DIA_SW WITH ROUND(dswt,4)
      vswt = F_FACT_SW*HEIGHT_SW*BA_SW
      REPLACE VOLUME_SW WITH ROUND(vswt,1)
ENDCASE
DO CASE
    CASE UPPER(rep20) = "Y" .AND. AGE_P2 = thinyr .AND. thintr <> 0
      NO_TREE_P2 + NO_TRE_AP2
      IF NO_TREE_P2 >= thintr
         REPLACE NO_TREE_P2 WITH thintr
         REPLACE NO_TRE_AP2 WITH 0
         REPLACE BA_AP2 WITH 0
         REPLACE VOLUME_AP2 WITH 0
      ELSE
         REPLACE NO_TRE_AP2 WITH thintr - NO_TREE_P2
         bapt2 = thinper*BA_AP2
         REPLACE BA_AP2 WITH ROUND(bapt2,2)
         dapt2 = 2*SQRT(BA_AP2*40000/(NO_TRE_AP2*3.14))
         REPLACE DIA_AP2 WITH ROUND(dapt2,4)
         vapt2 = F_FACT_P2*HEIGHT_P2*BA_P2
         REPLACE VOLUME_P2 WITH ROUND(vapt2,1)
      ENDIF
```

```
      bpt2 = thinper*BA_P2
      REPLACE BA_P2 WITH ROUND(bpt2,2)
      dpt2 = 2*SQRT(BA_P2*40000/(NO_TREE_P2*3.14))
      REPLACE DIA_P2 WITH ROUND(dpt2,4)
      vpt2 = F_FACT_P2*HEIGHT_P2*BA_P2
      REPLACE VOLUME_P2 WITH ROUND(vpt2,1)
   CASE UPPER(rep20) = "Y" .AND. AGE_SW2 = thinyr .AND. thintr < ;
      NO_TRE_SW2 + NO_TRE_AP2
      IF NO_TRE_SW2 >= thintr
         REPLACE NO_TRE_SW2 WITH thintr
         REPLACE NO_TRE_AP2 WITH 0
         REPLACE BA_AP2 WITH 0
         REPLACE VOLUME_AP2 WITH 0
      ELSE
         REPLACE NO_TRE_AP2 WITH thintr - NO_TRE_SW2
         bapt2 = thinper*BA_AP2
         REPLACE BA_AP2 WITH ROUND(bapt2,2)
         dapt2 = 2*SQRT(BA_AP2*40000/(NO_TRE_AP2*3.14))
         REPLACE DIA_AP2 WITH ROUND(dapt2,4)
         vapt2 = F_FACT_AP2*HEIGHT_AP2*BA_AP2
         REPLACE VOLUME_AP2 WITH ROUND(vapt2,1)
      ENDIF
      bswt2 = thinper*BA_SW2
      REPLACE BA_SW2 WITH ROUND(bswt2,2)
      dswt2 = 2*SQRT(BA_SW2*40000/(NO_TRE_SW2*3.14))
      REPLACE DIA_SW2 WITH ROUND(dswt2,4)
      vswt2 = F_FACT_SW2*HEIGHT_SW2*BA_SW2
      REPLACE VOLUME_SW2 WITH ROUND(vswt2,1)
ENDCASE
RETURN
Subroutine IRPRPT
* REPORTB.PRG
CALCULATE ALL AVG(BEAUTY), AVG(BEAUTY2) TO b1, b2
   Vol[1,6] = (b1 + b2)/2
CALCULATE ALL SUM(HUNTINM), SUM(HUNTINM2) TO hm1, hm2
   Vol[1,7] = (hm1 + hm2)
CALCULATE ALL SUM(HUNTINC), SUM(HUNTINC2) TO hc1, hc2
   Vol[1,8] = (hc1 + hc2)
CALCULATE ALL SUM(GRAZING), SUM(GRAZING2) TO g1, g2
   Vol[1,9] = (g1 + g2)
CALCULATE ALL SUM(MOOSE), SUM(MOOSE2) TO m1, m2
   Vol[1,10] = (m1 + m2)
CALCULATE ALL SUM(CARIBOU), SUM(CARIBOU2) TO c1, c2
   Vol[1,11] = (c1 + c2)
CALCULATE ALL AVG(WATER), AVG(WATER2) TO w1, w2
   Vol[1,12] = (w1 + w2)/2
STORE t TO Vol[1,1]
SELECT 6
APPEND FROM ARRAY Vol
SELECT 1
RETURN
```

APPENDIX B.3: A SAMPLE OF THE MODEL INPUT SCREEN

MAIN MENU

1. Add Record
2. Modify Record
3. Modify Database Structure
4. Simulation A
5. Simulation B
6. Simulation C
7. Exit

SIMULATION A

Enter the number of years to simulate: __

Enter the time interval of simulation:

Enter the rotation length: ______

Enter the number of years to regenerate forest after harvesting: ______

Enter the regeneration species, white spruce or pine? (S/P): ______

Enter the method to regenerate forest, planting or seeding? (P/S): ______

Do you want to do precommercial thinning? (Y/N):

Enter the precommercial thinning age: ___

Enter the percentage of basal area to be left after thinning: ______

Enter the stand density after thinning: ____

APPENDIX B.4: A SAMPLE OF THE SIMULATION REPORT

THE SIMULATION REPORT A

Total area of a unit = 64 ha. Rotation = 50

Number of units = 100

YEAR	W.SPRUCE VOLUME (cu m)	B.SPRUCE VOLUME (cu.m)	PINE VOLUME (cu.m)	ASPEN VOLUME (cu.m)	ESTHETICS (Mean Rating)
1	1708.8	11040.0	4492.8	460.8	6.3
2	[illegible]	4051.2	7916.8	1625.6	6.2
3	13977.6	4441.6	41939.2	6169.6	6.0
4	34195.2	3923.2	23468.8	12064.0	5.9
5	3443.2	3712.0	45657.6	1337.6	5.8
6	1107.2	5593.6	39968.0	1280.0	5.7
7	774.4	4505.6	35212.8	275.2	5.6
8	320.0	9664.0	12787.2	12.8	5.4
9	1478.4	13747.2	35904.0	1414.4	5.3
10	18304.0	5574.4	20588.8	339.2	5.2
11	7174.4	4672.0	40313.6	11033.6	5.1
12	870.4	4889.6	24505.6	2092.8	5.0
13	1286.4	11763.2	55923.2	1292.8	4.9
14	7782.4	1401.6	42009.6	992.0	4.9
15	736.0	2732.8	36588.8	1062.4	4.8
16	15129.6	2675.2	26604.8	1683.2	4.7
17	28940.8	1126.4	19456.0	1145.6	4.6
18	28697.6	1907.2	30560.0	7360.0	4.5
19	652.8	2572.8	28748.8	1433.6	4.4
20	4550.4	2931.2	8665.6	2137.6	4.3
21	1164.8	2764.8	18316.8	2073.6	4.2
22	1292.8	2803.2	25529.6	2060.8	4.2
23	16800.0	3436.8	28576.0	3520.0	4.1
24	3091.2	5356.8	33356.8	0.0	4.0
25	7104.0	486.4	21529.6	71449.6	3.9
26	2272.0	12672.0	5472.0	0.0	3.9
27	0.0	4243.2	8780.8	1958.4	3.8
28	19820.8	4281.6	48307.2	6822.4	3.8
29	40812.8	3424.0	25619.2	20236.8	3.7
30	3398.4	3116.8	51872.0	1209.6	3.6
31	262.4	5139.2	41728.0	563.2	3.6
32	19.2	4275.2	38252.8	0.0	3.6
33	0.0	10470.4	13932.8	0.0	3.5
34	742.4	14163.2	39456.0	1017.6	3.3
35	19641.6	5139.2	22304.0	12.8	3.3
36	7942.4	4473.6	44217.6	16825.6	3.3
37	0.0	4294.4	25702.4	1267.2	3.3
38	755.2	12473.6	60691.2	492.8	3.2
39	8480.0	556.8	45920.0	595.2	3.2
40	268.8	1798.4	39654.4	761.6	3.2
41	16057.6	1747.2	29734.4	1510.4	3.2
42	29228.8	364.8	21337.6	684.8	3.1
43	28819.2	915.2	33132.8	6924.8	3.2
44	211.2	1632.0	30374.4	953.6	3.1
45	4441.6	1875.2	8601.6	1440.0	3.1
46	294.4	1772.8	20115.2	1305.6	3.1
47	256.0	1740.8	27500.8	1267.2	3.1
48	18329.6	2841.6	30406.4	2457.6	3.1
49	3296.0	4652.8	34956.8	0.0	3.1
50	7008.0	268.8	23872.0	84320.0	3.1
51	0.0	0.0	54950.4	0.0	3.1

52	0.0	0.0	47059.2	0.0	3.1
53	0.0	0.0	54950.4	0.0	3.1
54	0.0	0.0	57369.6	0.0	3.1
55	0.0	0.0	43257.6	0.0	3.1
56	0.0	0.0	31564.8	0.0	3.1
57	0.0	0.0	23673.6	0.0	3.1
58	0.0	0.0	31564.8	0.0	3.1
59	0.0	0.0	33984.0	0.0	3.1
60	0.0	0.0	36403.2	0.0	3.1
61	0.0	0.0	43257.6	0.0	3.1
62	0.0	0.0	36403.2	0.0	3.1
63	0.0	0.0	43257.6	0.0	3.1
64	0.0	0.0	26092.8	0.0	3.1
65	0.0	0.0	33984.0	0.0	3.1
66	0.0	0.0	36403.2	0.0	3.1
67	0.0	0.0	36403.2	0.0	3.1
68	0.0	0.0	38822.4	0.0	3.1
69	0.0	0.0	43257.6	0.0	3.1
70	0.0	0.0	33984.0	0.0	3.1
71	0.0	0.0	38822.4	0.0	3.1
72	0.0	0.0	36403.2	0.0	3.1
73	0.0	0.0	57369.6	0.0	3.1
74	0.0	0.0	31564.8	0.0	3.1
75	0.0	0.0	41241.6	0.0	3.1
-----	---------	---------	---------	---------	---------
Total	413235	226106	2477608	286944	3.8

THE SIMULATION REPORT B

Total area of a unit - 64 ha.

Rotation - 50

Number of units - 100

YEAR	MOOSE HUNTING (V.Days)	CARIBOU HUNTING (V.Days)	DOMESTIC GRAZING (AUMs)	MOOSE POP. (No.)	CARIBOU POP. (No.)	WATER YIELD (mm)
1	0.0	352.0	42470.4	51.1	15.0	173
2	0.0	22.7	42489.9	66.0	15.0	173
3	1.0	24.1	43414.7	82.9	15.0	173
4	129.0	24.7	44346.9	86.6	15.0	174
5	148.0	25.4	45222.2	88.0	15.0	174
6	155.8	25.4	45875.4	88.7	14.9	175
7	158.6	25.3	46692.2	89.4	15.0	175
8	163.9	26.6	46946.2	89.5	14.9	175
9	164.9	26.0	47485.3	89.7	14.8	175
10	165.3	25.9	48076.0	90.0	14.7	176
11	168.6	25.9	48876.7	90.3	14.6	176
12	170.9	25.8	49347.1	90.6	14.6	177
13	173.7	25.6	50356.6	91.1	14.5	177
14	177.4	25.8	50978.6	91.5	14.5	177
15	181.0	25.8	51308.8	91.8	14.3	178
16	182.6	25.0	51611.4	92.1	14.2	178
17	184.7	24.5	51919.1	92.4	14.2	178
18	186.6	24.7	52487.3	92.9	14.0	179
19	188.5	23.9	52729.2	93.2	13.9	179
20	192.3	23.3	52562.8	93.1	13.8	179
21	191.5	22.8	52494.1	93.1	13.7	180
22	190.8	22.2	52395.2	93.1	13.5	180
23	190.2	21.6	52622.0	93.3	13.5	180
24	190.8	21.7	52911.7	93.5	13.4	181
25	192.9	21.1	53624.9	93.7	13.2	181
26	192.4	20.3	53059.4	93.9	13.1	182
27	194.8	19.4	52564.8	93.7	13.0	182
28	193.1	19.2	53013.1	93.8	12.8	182
29	194.5	18.4	53462.3	94.1	12.7	183
30	197.3	17.6	53665.1	94.4	12.6	183
31	199.8	16.9	53577.9	94.4	12.4	183
32	199.9	16.0	53607.7	94.7	12.3	184
33	203.0	15.4	53011.0	94.5	12.2	184
34	201.6	14.7	52635.9	94.3	12.0	184
35	199.2	13.8	52283.0	94.3	11.9	184
36	199.6	13.2	52099.2	94.2	11.7	184
37	198.7	12.3	51556.9	94.1	11.6	185
38	197.8	11.5	51546.6	94.2	11.4	185
39	197.6	10.5	51160.7	94.2	11.3	185
40	197.2	9.	50495.9	94.1	11.2	185
41	194.2	8.[illegible]	49817.1	93.9	11.0	186
42	191.5	8.0	49161.5	93.7	10.9	186
43	188.3	7.1	48796.0	93.7	10.7	186
44	185.2	6.3	48144.2	93.5	10.6	186
45	184.2	5.4	47124.3	92.8	10.4	187
46	178.2	4.5	46243.8	92.2	10.3	187
47	172.2	3.6	45381.7	91.7	10.1	187
48	166.9	2.7	44915.4	91.3	9.9	187
49	163.0	1.8	44580.6	90.9	9.8	187
50	160.7	0.9	44715.1	90.6	9.6	188
51	156.0	0.0	44856.4	90.7	9.6	188
52	156.5	0.0	44717.9	90.7	9.6	188
53	155.9	0.0	44901.6	90.7	9.6	188
54	155.7	0.0	45091.6	90.7	9.6	188
55	156.8	0.0	45282.8	90.8	9.6	188

56	158.0	0.0	45456.4	90.9	9.6	188
57	159.5	0.0	45310.7	90.8	9.6	188
58	159.7	0.0	45444.4	90.8	9.6	188
59	160.0	0.0	45546.8	90.9	9.6	188
60	161.1	0.0	45614.4	91.0	9.6	188
61	162.2	0.0	45665.6	91.0	9.6	188
62	162.2	0.0	4565[illegible].5	91.2	9.6	188
63	163.9	0.0	45611.6	91.2	9.6	188
64	163.7	0.0	45207.7	91.2	9.6	188
65	164.2	0.0	45076.7	91.1	9.6	188
66	163.7	0.0	44916.8	91.1	9.6	188
67	163.9	0.0	44763.1	91.1	9.6	188
68	163.7	0.0	44630.8	91.1	9.6	188
69	163.4	0.0	44555.9	91.0	9.6	188
70	161.7	0.0	44504.5	91.0	9.6	188
71	161.7	0.0	44484.2	91.0	9.6	188
72	161.3	0.0	44497.2	90.9	9.6	188
73	160.2	0.0	44558.3	90.8	9.6	188
74	158.3	0.0	44631.6	90.9	9.6	188
75	158.3	0.0	44715.1	90.9	9.6	188
------	--------	---------	----------	-------	--------	------
Aver.	168.2	16.2	48154.1	90.9	11.8	183

APPENDIX B.5: ORDER AND CONTENTS OF THE SIMULATION DATABASE

Field	Field_Name	Field_Type	Width	Dec	Description of field name
1	NUMBER	Character	4		Identification number of planning unit.
2	TOWNSHIP	Numeric	3		Name of township, range, meridian, section, quarter section, and stand in which planning unit is located.
3	RANGE	Numeric	3		
4	MERIDIAN	Numeric	3		
5	SECTION	Numeric	3		
6	QSECTION	Numeric	2		
7	STAND	Character	4		
8	FCT	Character	12		Forest cover type.
9	SITE	Character	2		Site class.
10	AREA	Numeric	9	1	Total area of stand in which planning unit is located.
11	ORIGIN	Numeric	6		Year when tree growth starts.
12	AGE_SW	Numeric	4		Age of white spruce in the subunit 1 and 2.
13	AGE_SW2	Numeric	4		
14	SI_SW	Numeric	4	1	Site index of white spruce.
15	HEIGHT_SW	Numeric	9	6	Site height (m) of white spruce in the subunit 1 and 2.
16	HEIGHT_SW2	Numeric	9	6	
17	DIA_SW	Numeric	9	6	Average DBH (cm) of all live white spruce trees in the subunit 1 and 2.
18	DIA_SW2	Numeric	9	6	
19	NO_TREE_SW	Numeric	12	4	Number of live white spruce trees in the subunit 1 and 2.
20	NO_TRE_SW2	Numeric	12	4	
21	BA_SW	Numeric	5	2	Basal area (sq. m) of white spruce in the subunit 1 and 2.
22	BA_SW2	Numeric	5	2	
23	F_FACT_SW	Numeric	5	2	Form factor of white spruce in the subunit 1 and 2. Will be a function of tree age.
24	F_FACT_SW2	Numeric	5	2	
25	VOLUME_SW	Numeric	8	1	Gross volume (cu. m) of white spruce in the subunit 1 and 2.
26	VOLUME_SW2	Numeric	8	1	
27	AGE_SB	Numeric	4		Age of black spruce in the subunit 1 and 2.
28	AGE_SB2	Numeric	4		
29	SI_SB	Numeric	4	1	Site index of black spruce.
30	HEIGHT_SB	Numeric	9	6	Site height (m) of black spruce in the subunit 1 and 2.
31	HEIGHT_SB2	Numeric	9	6	
32	DIA_SB	Numeric	9	6	Average DBH (cm) of all live black spruce trees in the subunit 1 and 2.
33	DIA_SB2	Numeric	9	6	
34	NO_TREE_SB	Numeric	12	4	Number of live black spruce trees in the subunit 1 and 2.
35	NO_TRE_SB2	Numeric	12	4	
36	BA_SB	Numeric	5	2	Basal area (sq. m) of black spruce in the subunit 1 and 2.
37	BA_SB2	Numeric	5	2	

38	F_FACT_SB	Numeric	5	2	Form factor of black spruce in the subunit 1 and 2. Will be a function of tree age.
39	F_FACT_SB2	Numeric	5	2	
40	VOLUME_SB	Numeric	8	1	Gross volume (cu. m) of black spruce in the subunit 1 and 2.
41	VOLUME_SB2	Numeric	8	1	
42	AGE_P	Numeric	4		Age of pine in the subunit 1 and 2.
43	AGE_P2	Numeric	4		
44	SI_P	Numeric	4	1	Site index of pine.
45	HEIGHT_P	Numeric	9	6	Site height (m) of pine in the subunit 1 and 2.
46	HEIGHT_P2	Numeric	9	6	
47	DIA_P	Numeric	9	6	Average DBH (cm) of all live pine trees in the subunit 1 and 2.
48	DIA_P2	Numeric	9	6	
49	NO_TREE_P	Numeric	12	4	Number of live pine trees in the subunit 1 and 2.
50	NO_TREE_P2	Numeric	12	4	
51	BA_P	Numeric	5	2	Basal area (sq. m) of pine in the subunit 1 and 2.
52	BA_P2	Numeric	5	2	
53	F_FACT_P	Numeric	5	2	Form factor of pine in the subunit 1 and 2. Will be a function of tree age.
54	F_FACT_P2	Numeric	5	2	
55	VOLUME_P	Numeric	8	1	Gross volume (cu. m) of pine in the subunit 1 and 2.
56	VOLUME_P2	Numeric	8	1	
57	AGE_AP	Numeric	4		Age of aspen in the subunit 1 and 2.
58	AGE_AP2	Numeric	4		
59	SI_AP	Numeric	4	1	Site index of aspen.
60	HEIGHT_AP	Numeric	9	6	Site height (m) of aspen in the subunit 1 and 2.
61	HEIGHT_AP2	Numeric	9	6	
62	DIA_AP	Numeric	9	6	Average DBH (cm) of all live aspen trees in the subunit 1 and 2.
63	DIA_AP2	Numeric	9	6	
64	NO_TREE_AP	Numeric	12	4	Number of live aspen trees in the subunit 1 and 2.
65	NO_TRE_AP2	Numeric	12	4	
66	BA_AP	Numeric	5	2	Basal area (sq. m) of aspen in the subunit 1 and 2.
6	BA_AP2	Numeric	5	2	
68	F_FACT_AP	Numeric	5	2	Form factor of aspen in the subunit 1 and 2. Will be a function of tree age.
69	F_FACT_AP2	Numeric	5	2	
70	VOLUME_AP	Numeric	8	1	Gross volume (cu. m) of aspen in the subunit 1 and 2.
71	VOLUME_AP2	Numeric	8	1	
72	HUNTINM	Numeric	8	3	Visitor-days of moose hunting in the subunit 1 and 2.
73	HUNTINM2	Numeric	8	3	
74	HUNTINC	Numeric	8	3	Visitor-days of caribou hunting in the subunit 1 and 2.
75	HUNTINC2	Numeric	8	3	
76	BEAUTY	Numeric	5	2	Mean rating of natural scenic beauty of subunit 1 and 2.
77	BEAUTY2	Numeric	5	2	

78	GRAZING	Numeric	10	2	Carrying capacity (AUM's) of the subunit 1 and 2 available for domestic grazing.
79	GRAZING2	Numeric	10	2	
80	CARIBOU	Numeric	10	3	Caribou density (animals/q. section) of the subunit 1 and 2.
81	CARIBOU2	Numeric	10	3	
82	MOOSE	Numeric	10	3	Moose density (animals/q. section) of the subunit 1 and 2.
83	MOOSE2	Numeric	10	3	
84	BASIN	Character	7		Name of basin in which planning unit is located.
85	PRECIP	Numeric	4		Annual average precipitation (mm) of the planning unit.
86	WATER	Numeric	4		Annual average water yield (mm) of the subunit 1 and 2.
87	WATER2	Numeric	4		

APPENDIX C: VALIDATION CURVES FOR SIMULATION MODELS

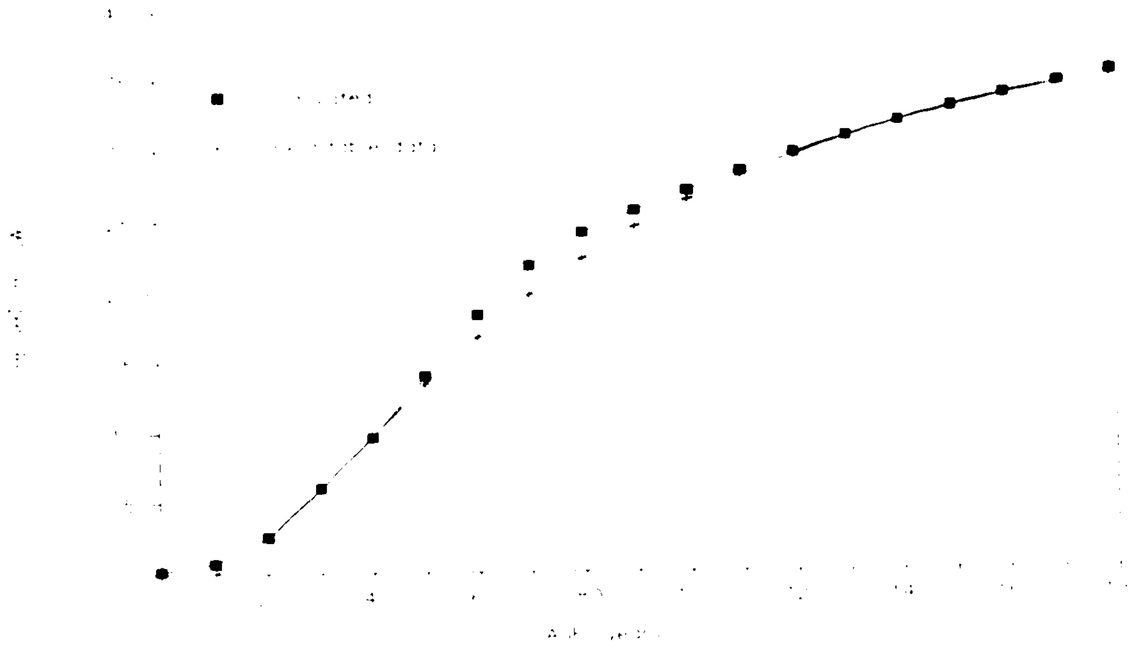


Figure C.1 Simulated and yield table site height of white spruce on good sites

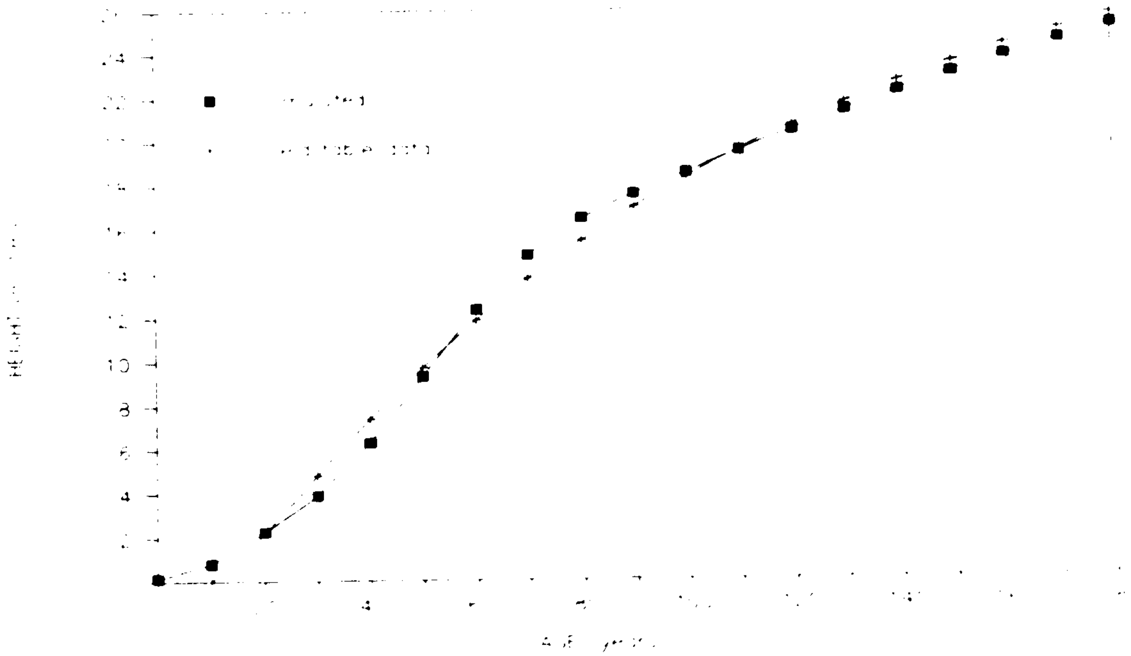


Figure C.2 Simulated and yield table site height of black spruce on good sites

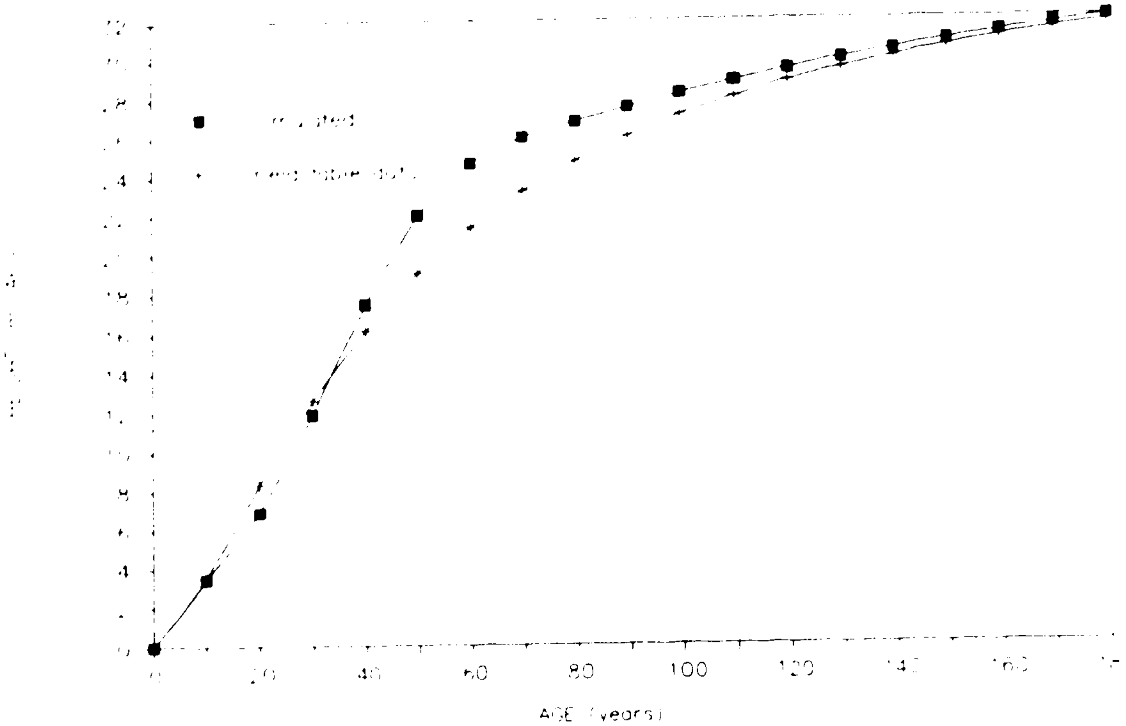


Figure C.3 Simulated and yield table site height of aspen on good sites

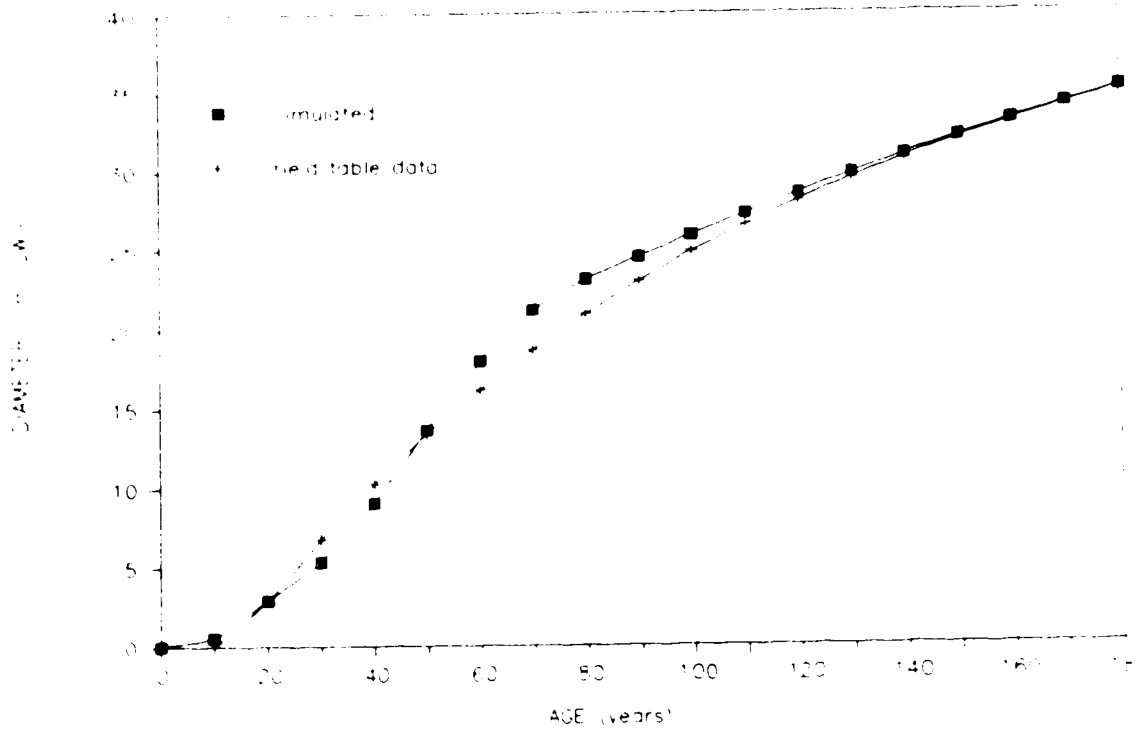


Figure C.4 Simulated and yield table average DBH of white spruce on good sites

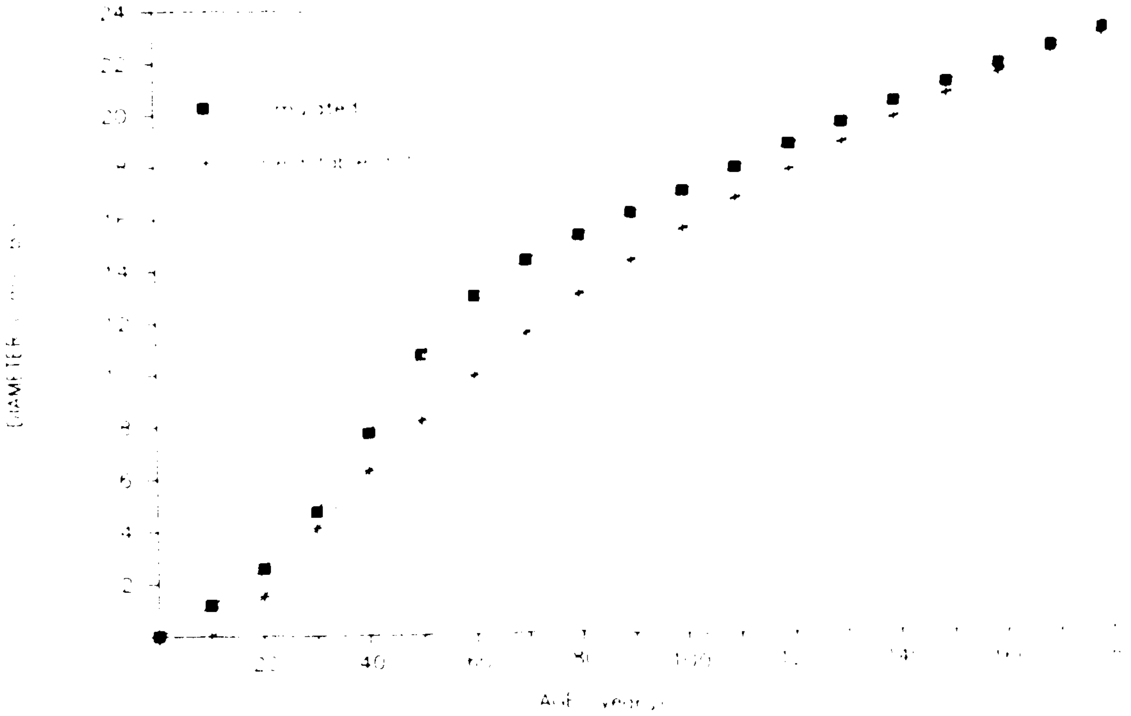


Figure C.5 Simulated and yield table average DBH of black spruce on good sites

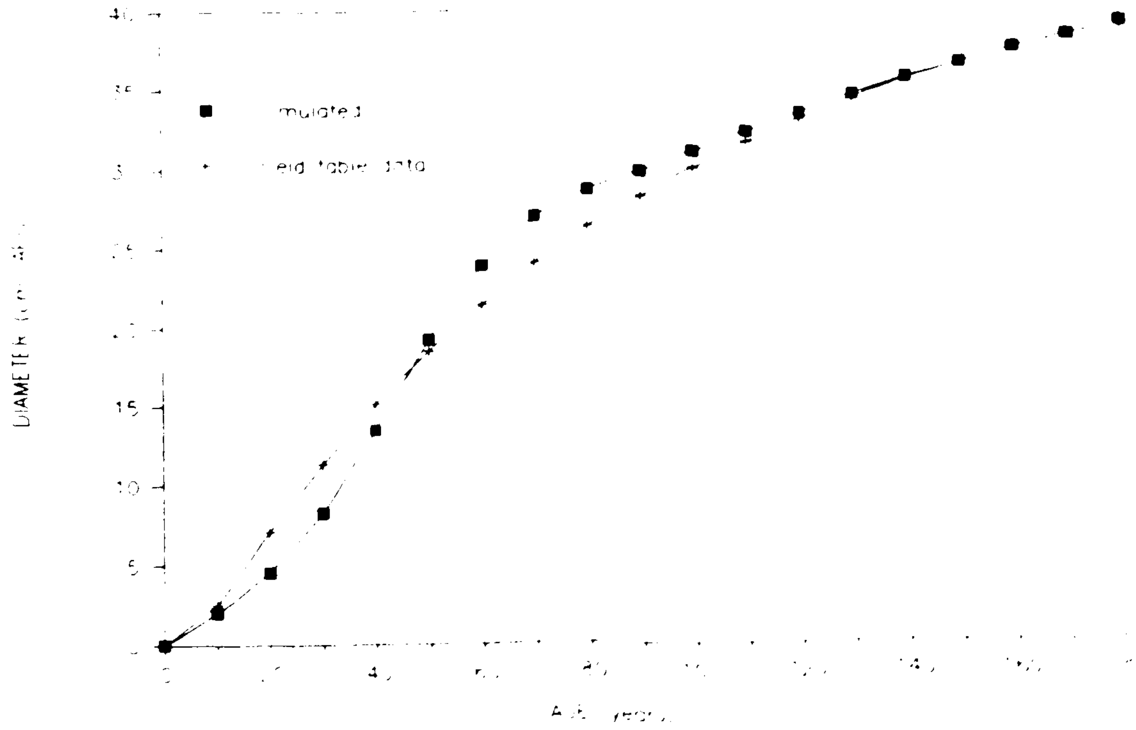


Figure C.6 Simulated and yield table average DBH of aspen on good sites

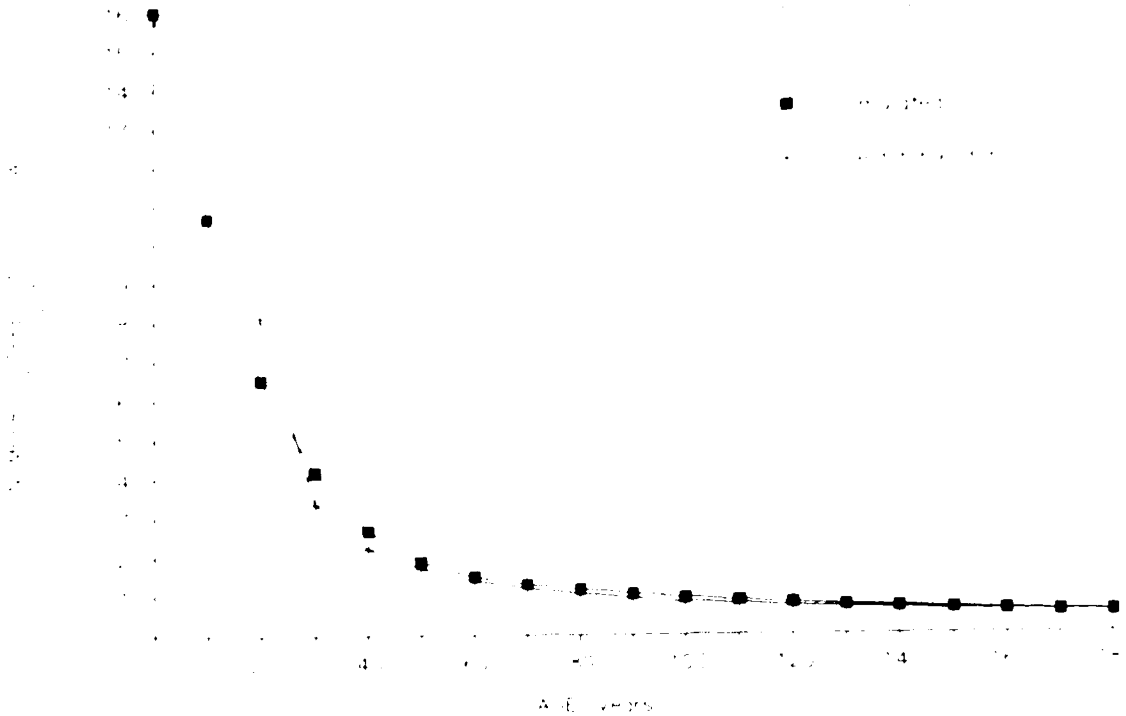


Figure C.7 Simulated and yield table number of trees of white spruce on good sites

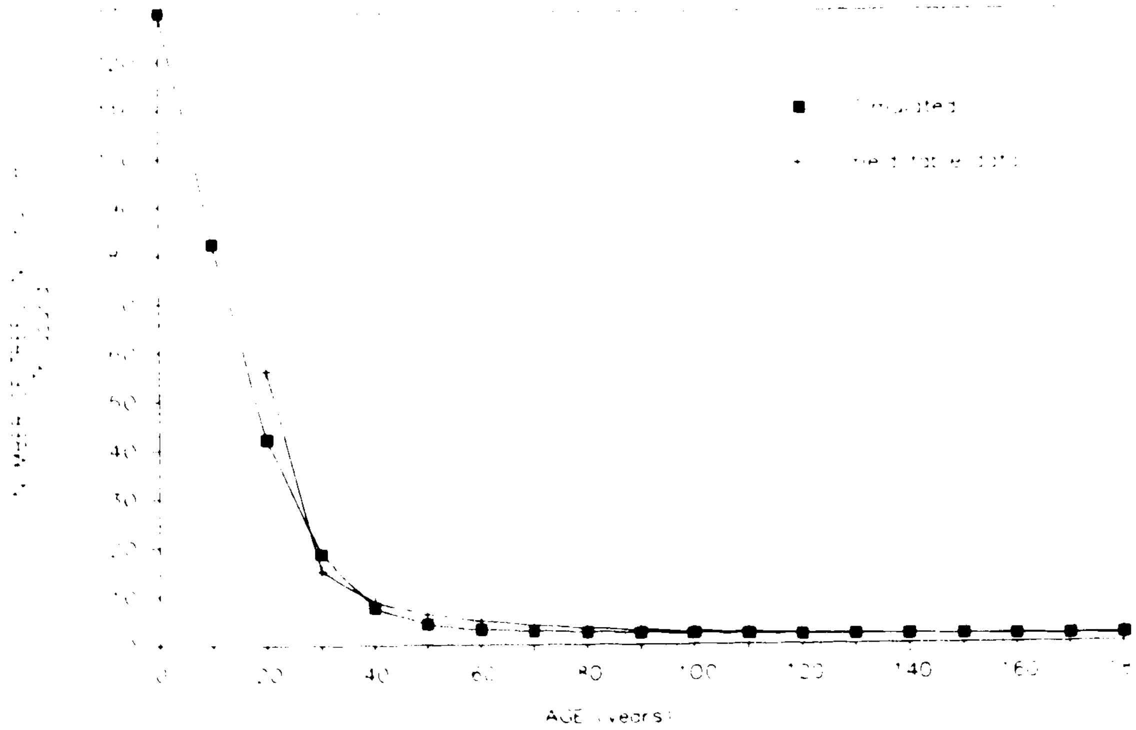


Figure C.8 Simulated and yield table number of trees of black spruce on good sites

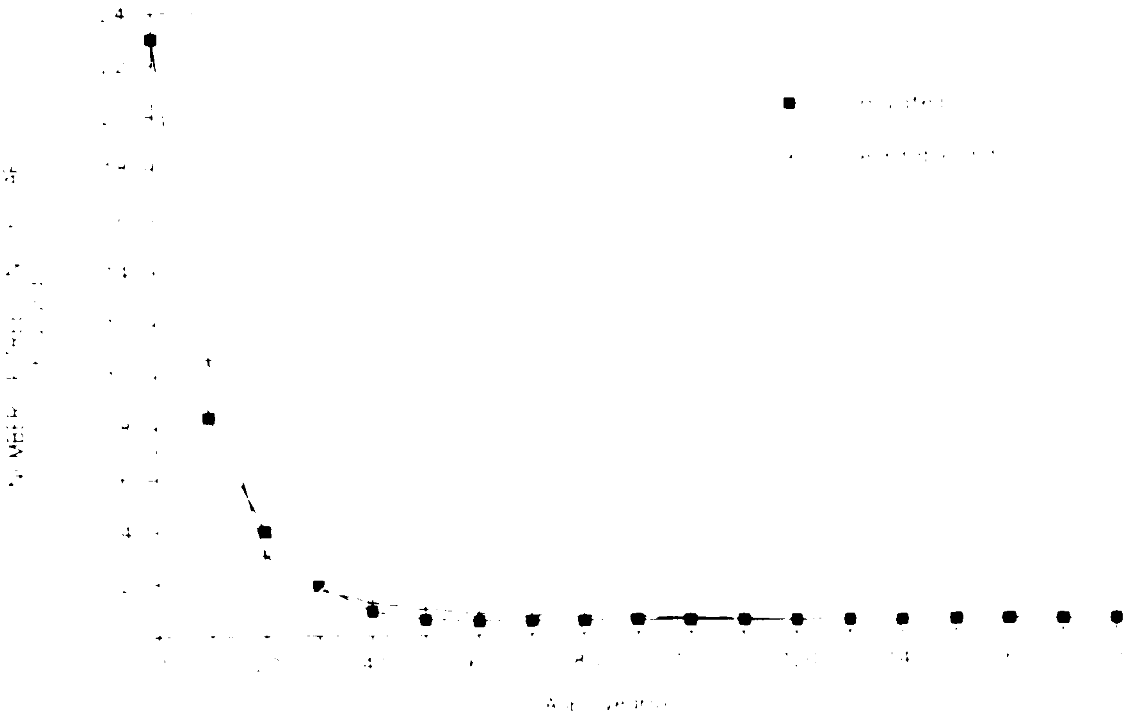


Figure C.9 Simulated and yield table number of trees of aspen on good sites

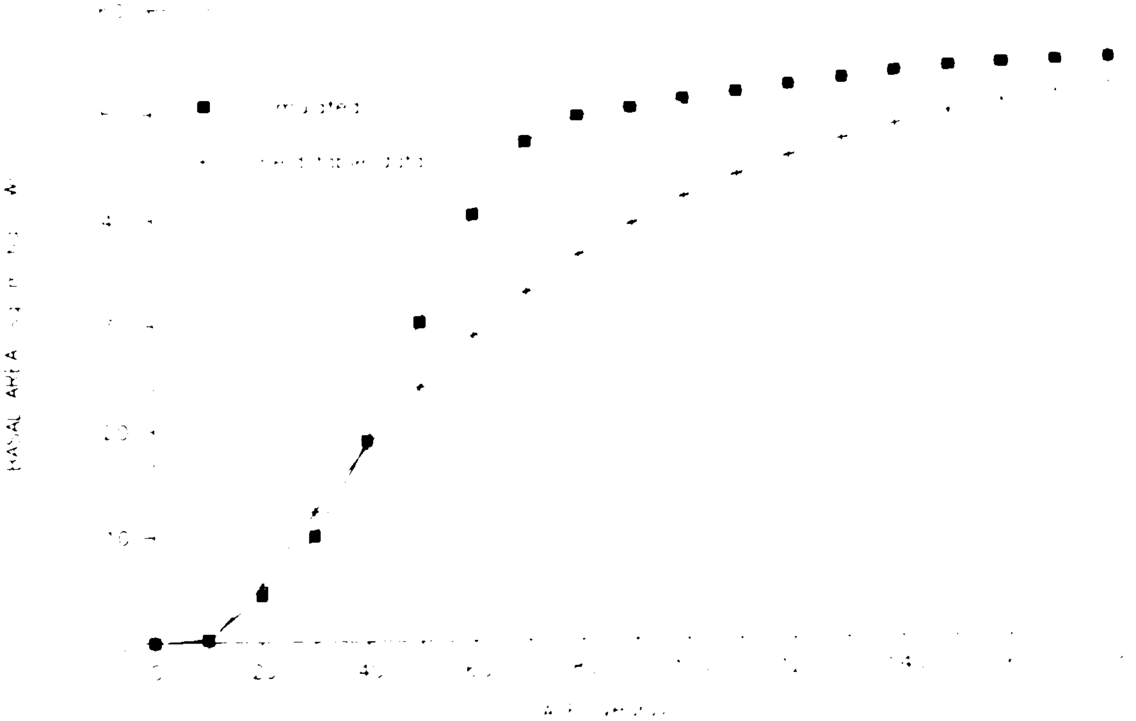


Figure C.10 Simulated and yield table basal area of white spruce on good sites

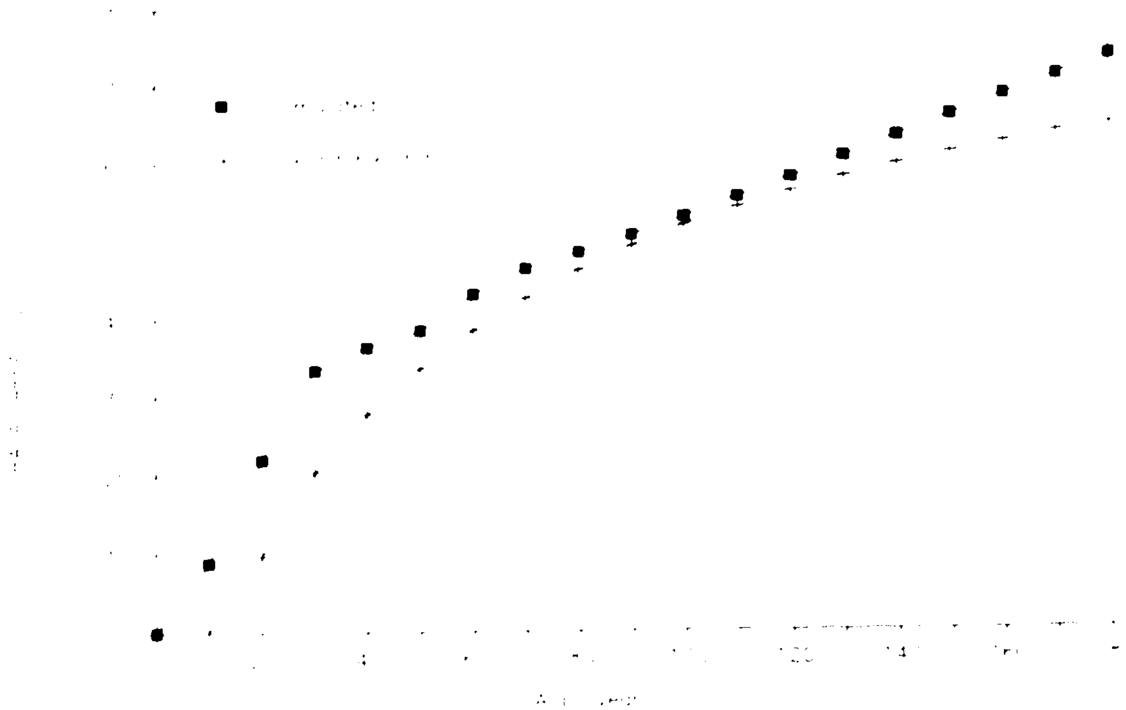


Figure C.11 Simulated and yield table basal area of black spruce on good sites

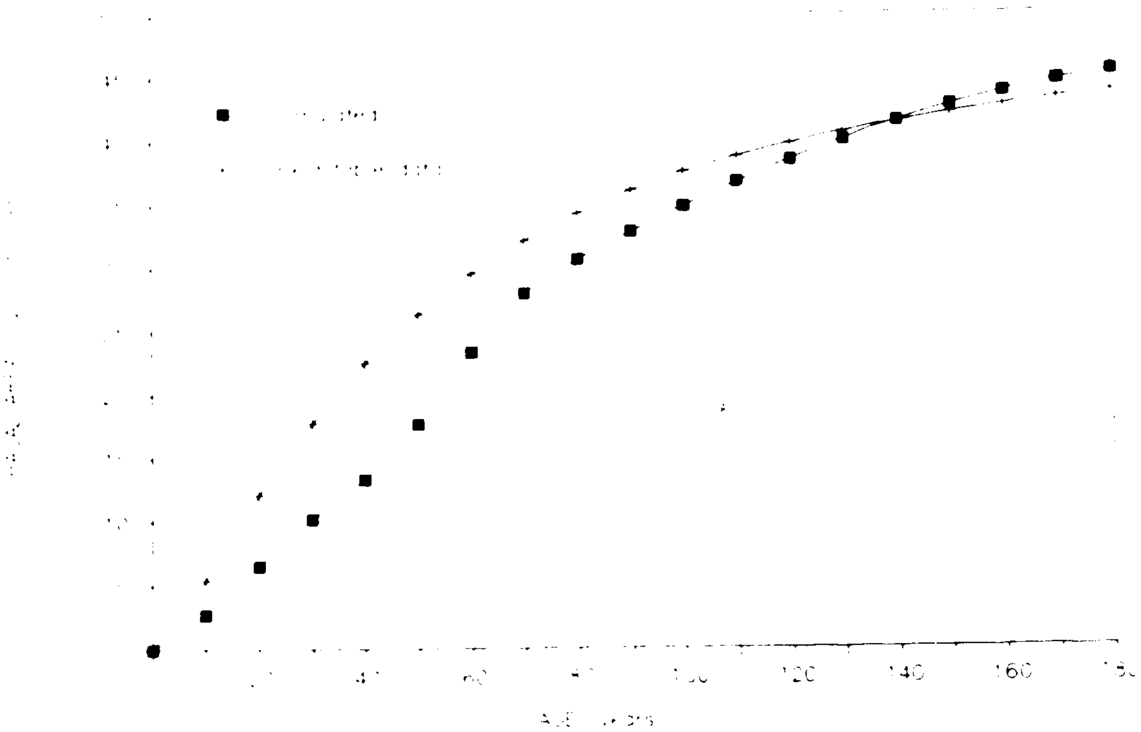


Figure C.12 Simulated and yield table basal area of aspen on good sites

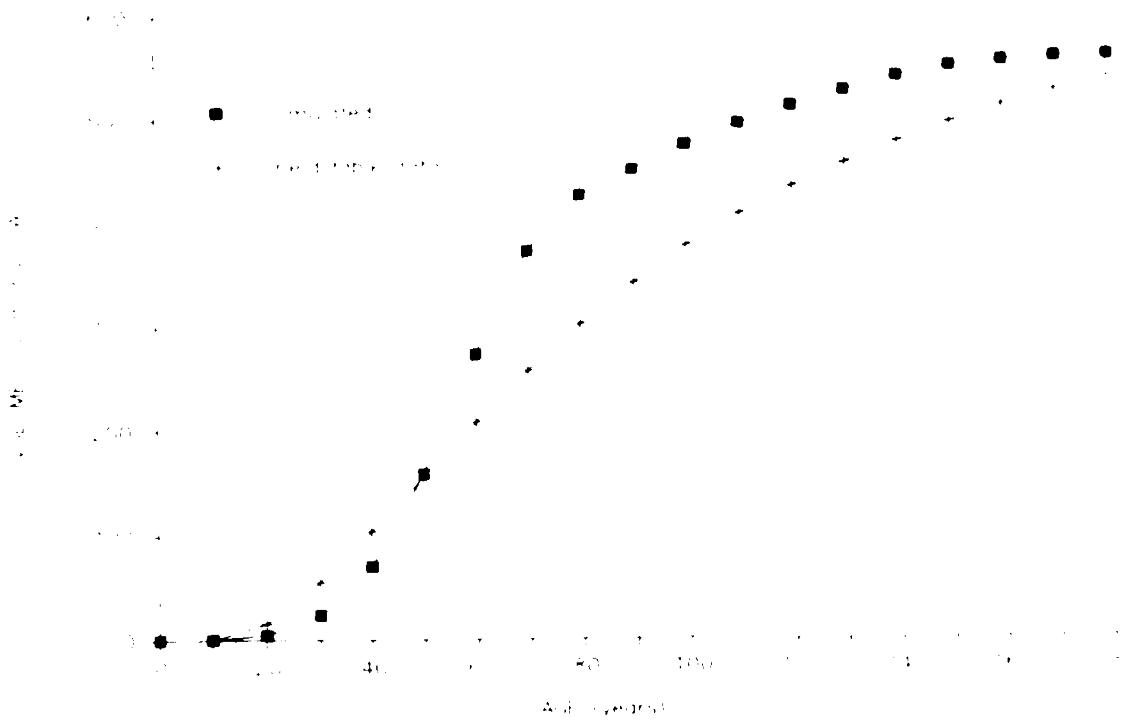


Figure C.13 Simulated and yield table gross volume of white spruce on good sites

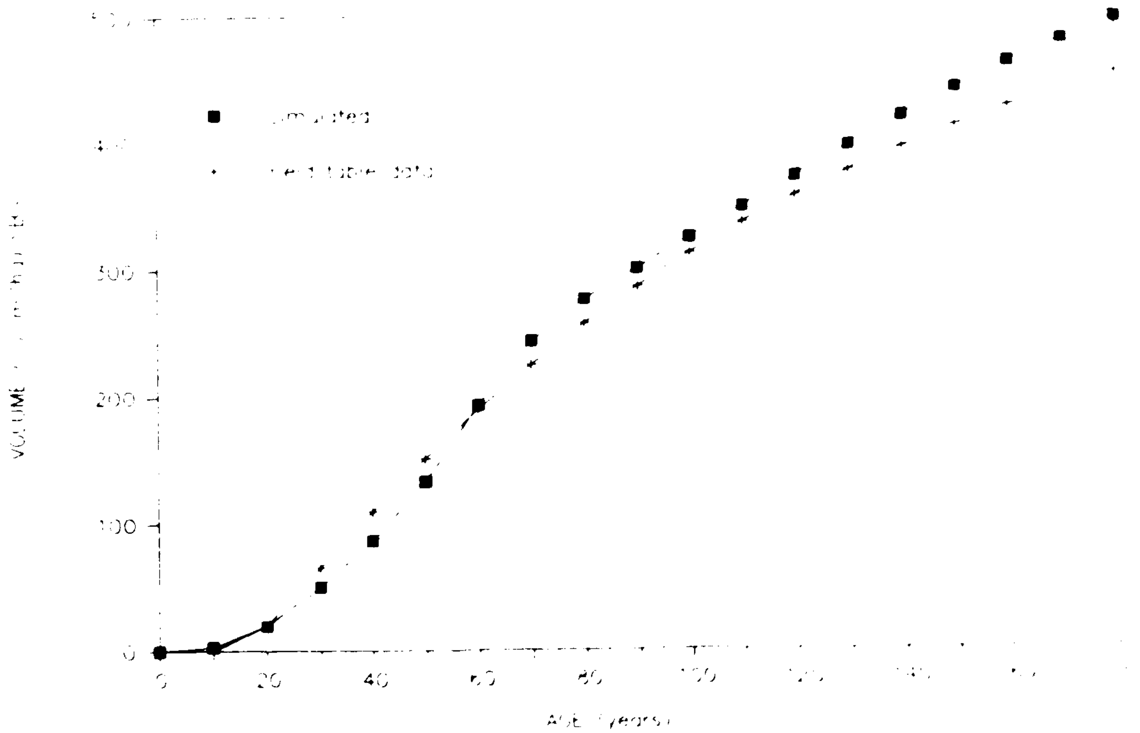


Figure C.14 Simulated and yield table gross volume of black spruce on good sites

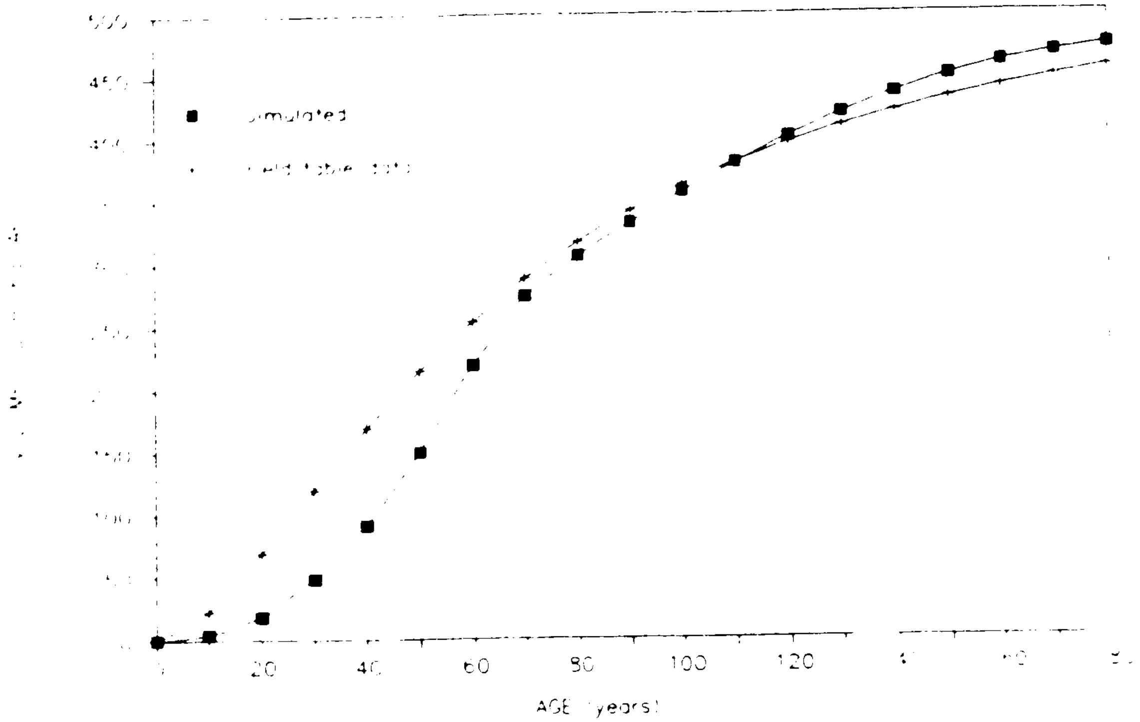


Figure C.15 Simulated and yield table gross volume of aspen on good sites

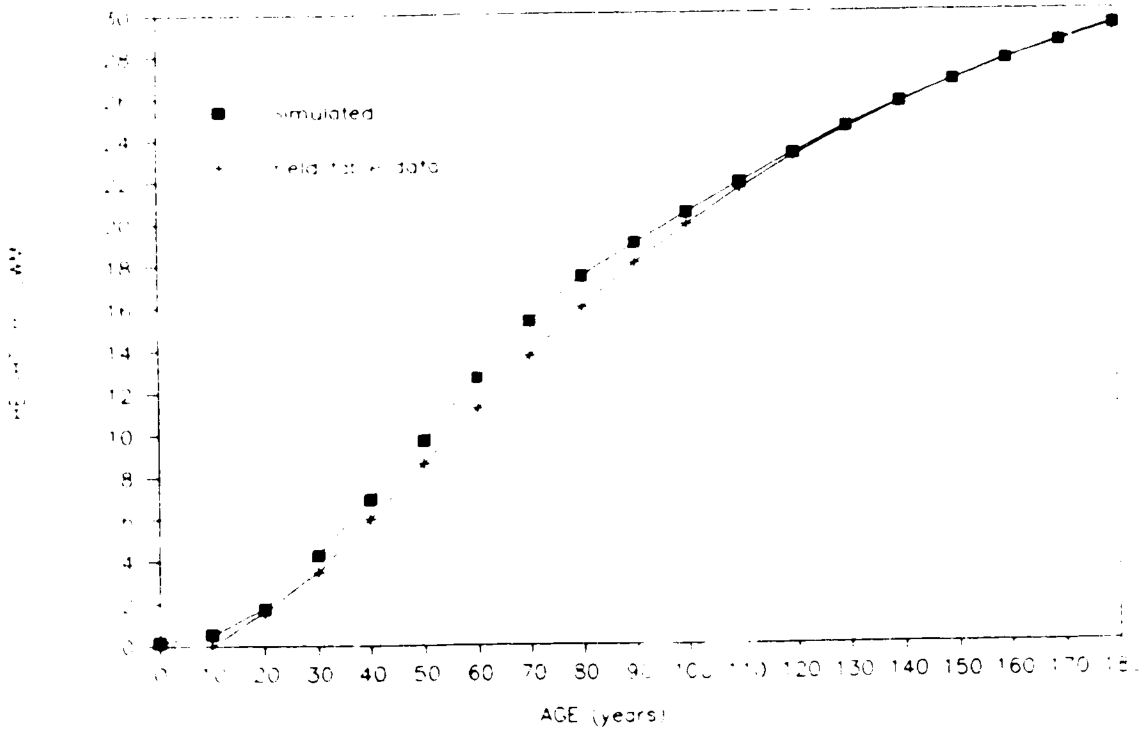


Figure C.16 Simulated and yield table site height of white spruce on medium sites

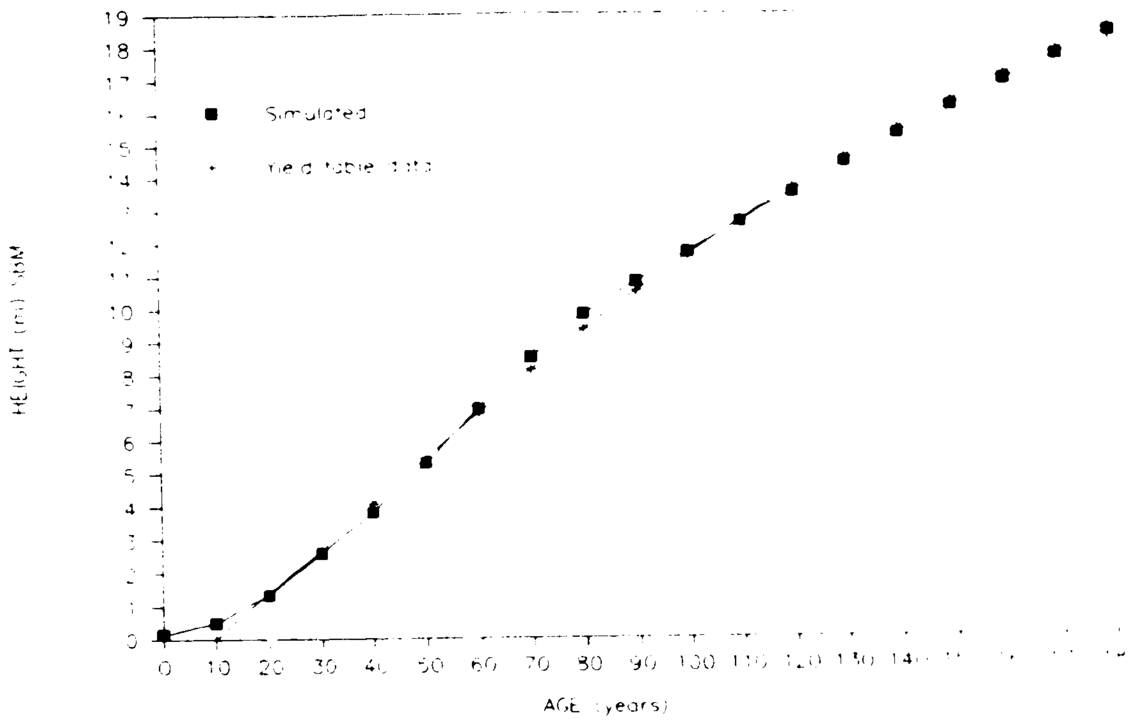


Figure C.17 Simulated and yield table site height of black spruce on medium sites

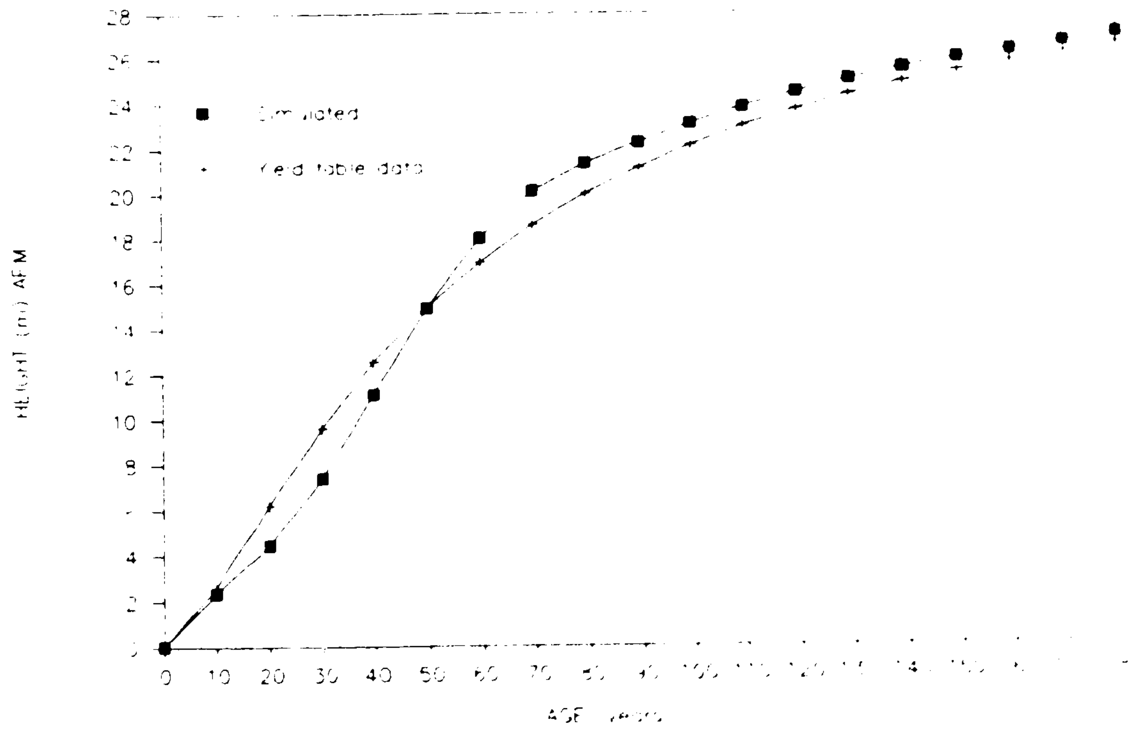


Figure C.18 Simulated and yield table site height of aspen on medium sites

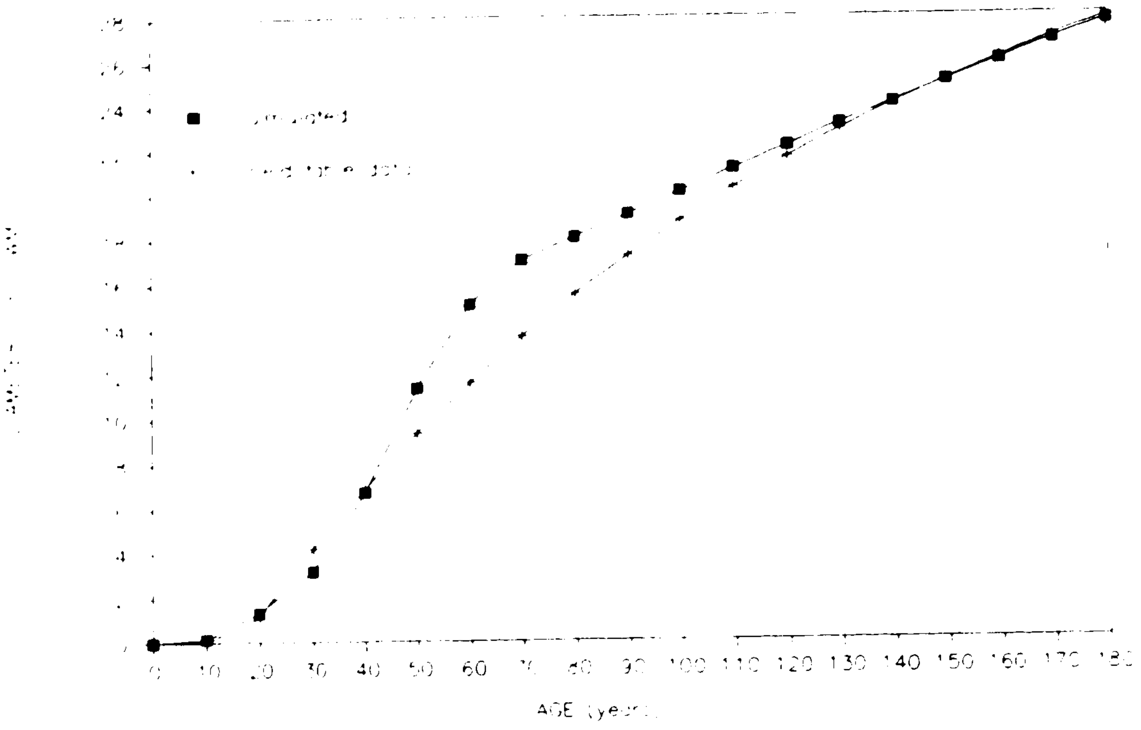


Figure C.19 Simulated and yield table average DBH of white spruce on medium sites

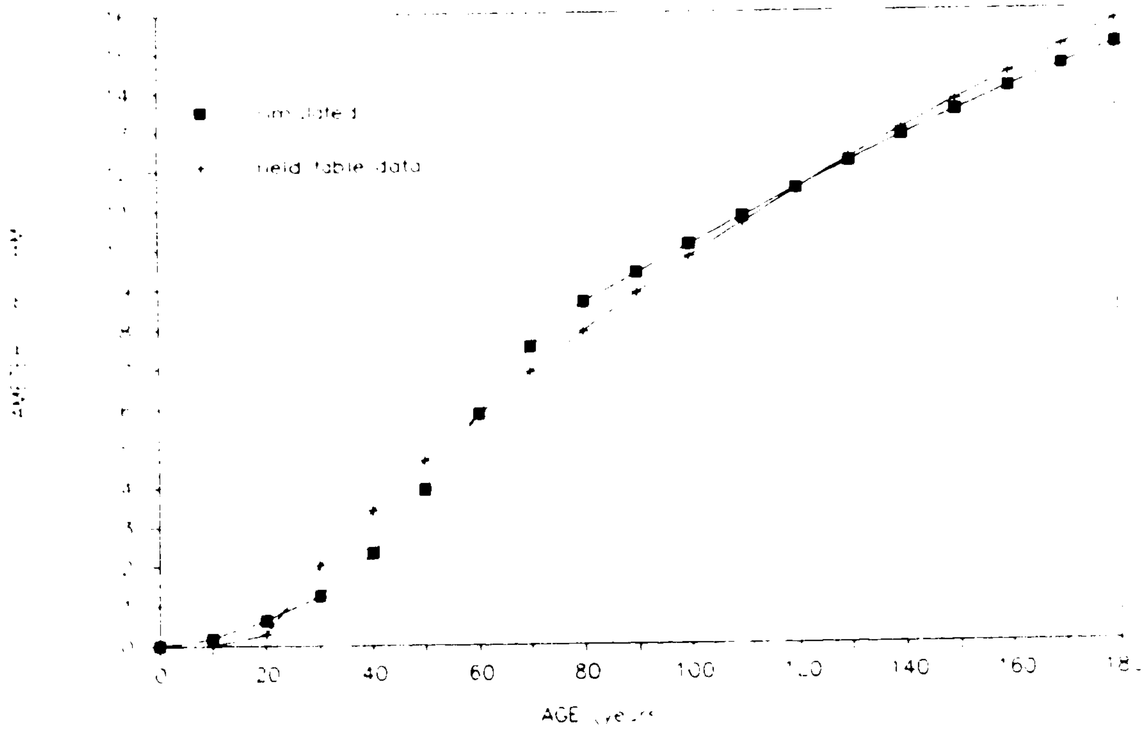


Figure C.20 Simulated and yield table average DBH of black spruce on medium sites

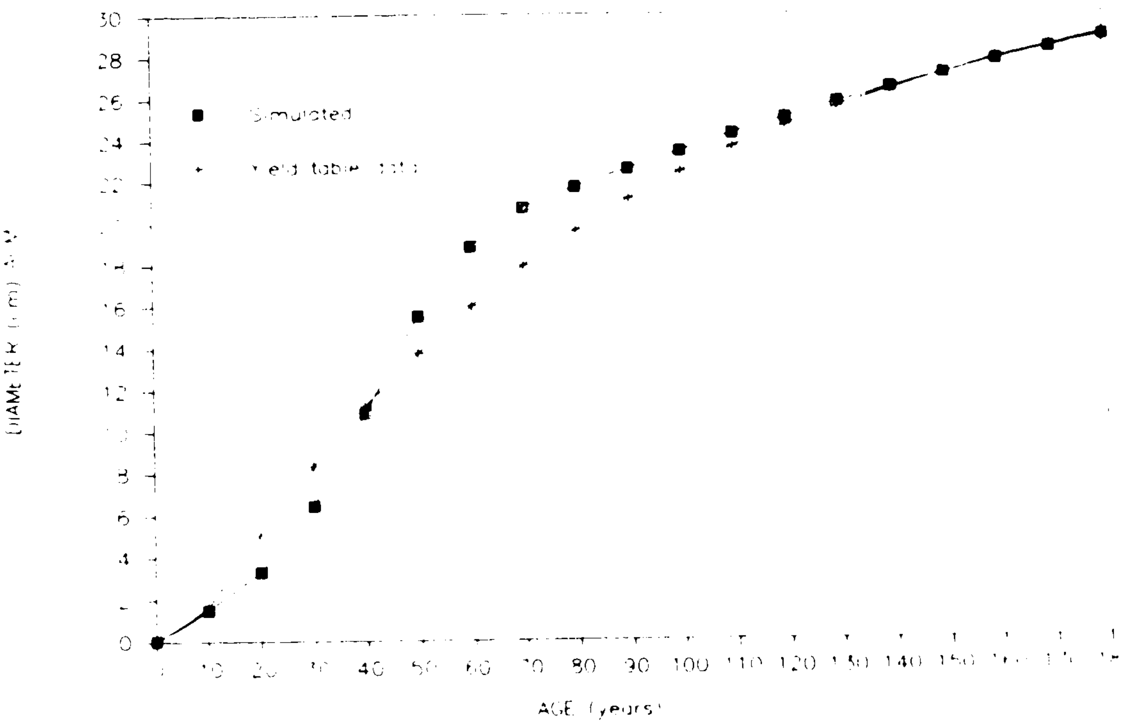


Figure C.21 Simulated and yield table average DBH of aspen on medium sites

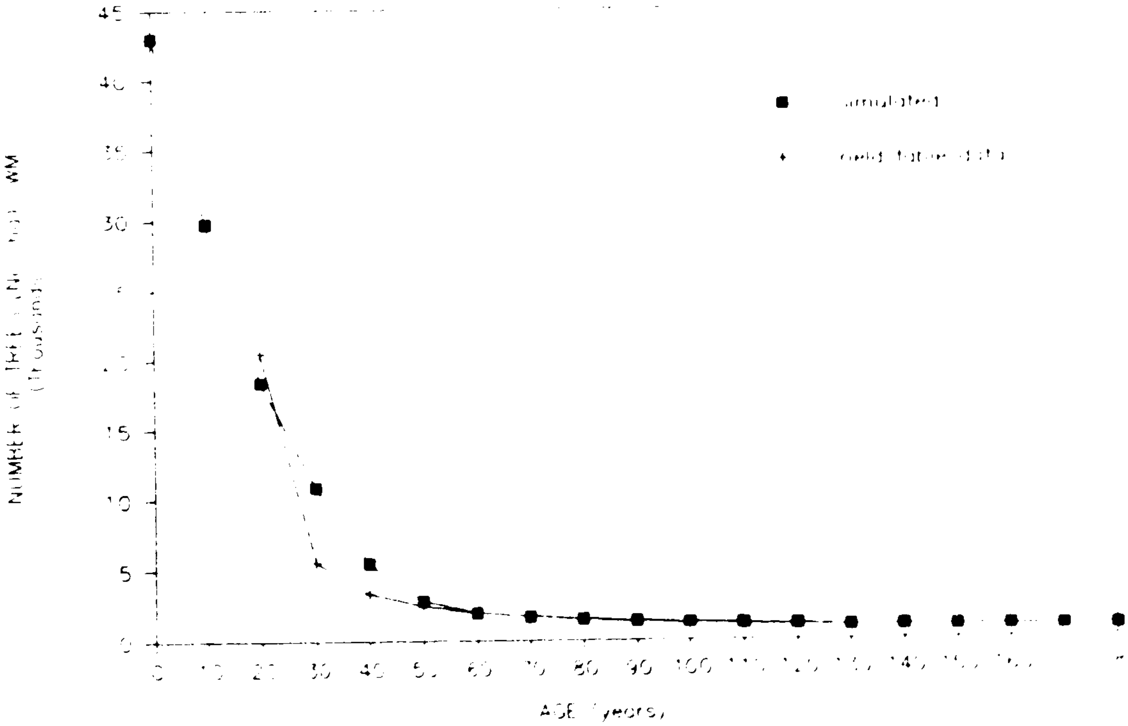


Figure C.22 Simulated and yield table number of trees of white spruce on medium sites

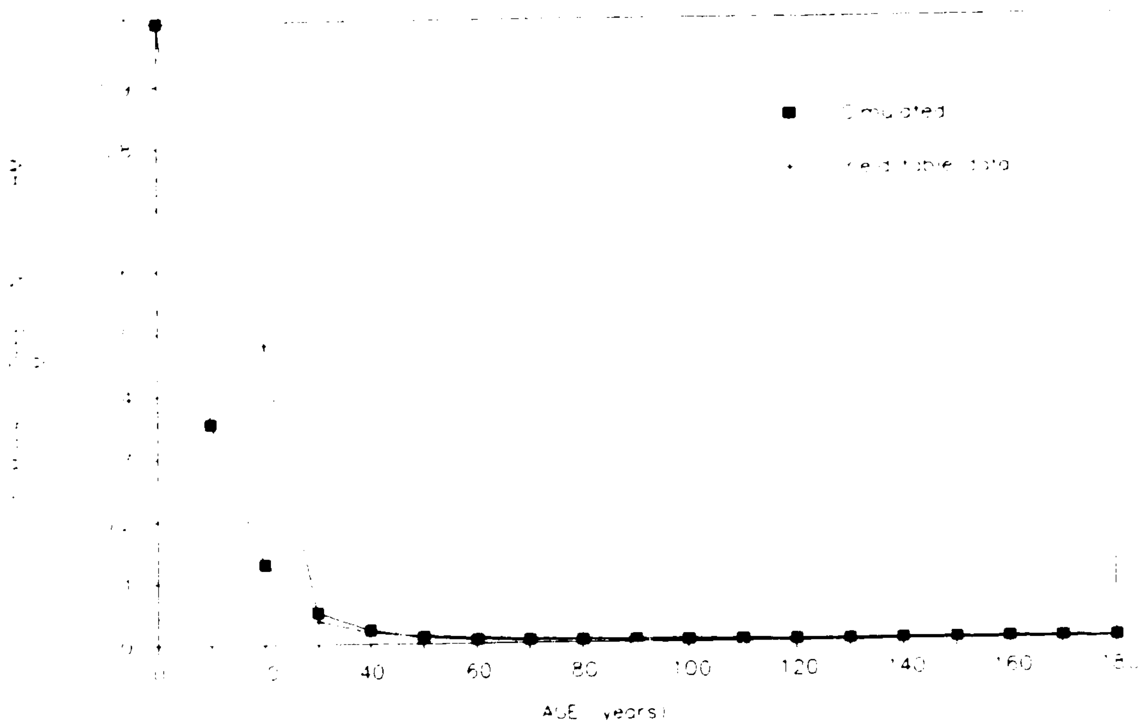


Figure C.23 Simulated and yield table number of trees of black spruce on medium sites

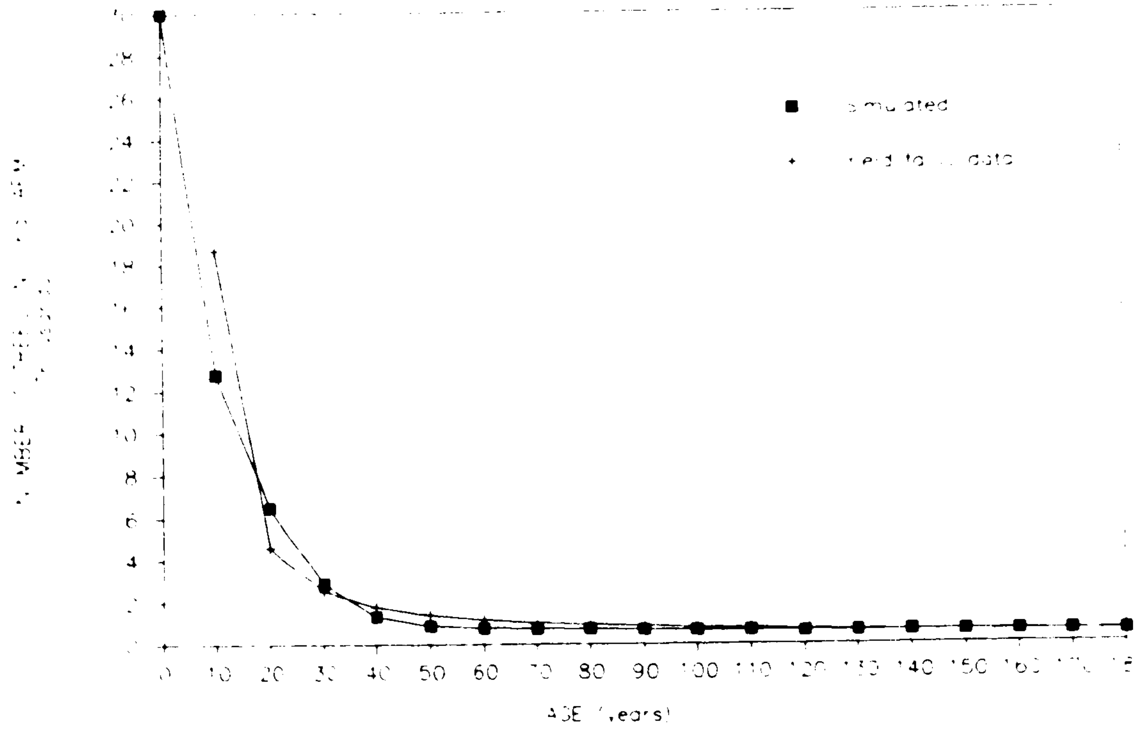


Figure C.24 Simulated and yield table number of trees of aspen on medium sites

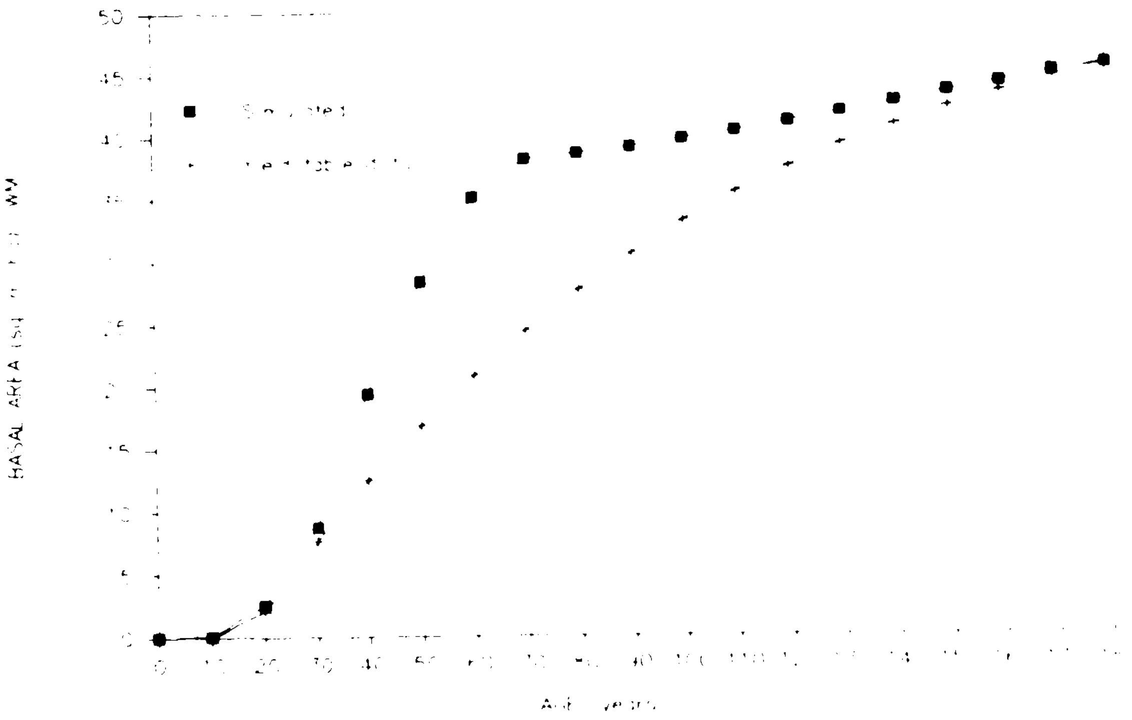


Figure C.25 Simulated and yield table basal area of white spruce on medium sites

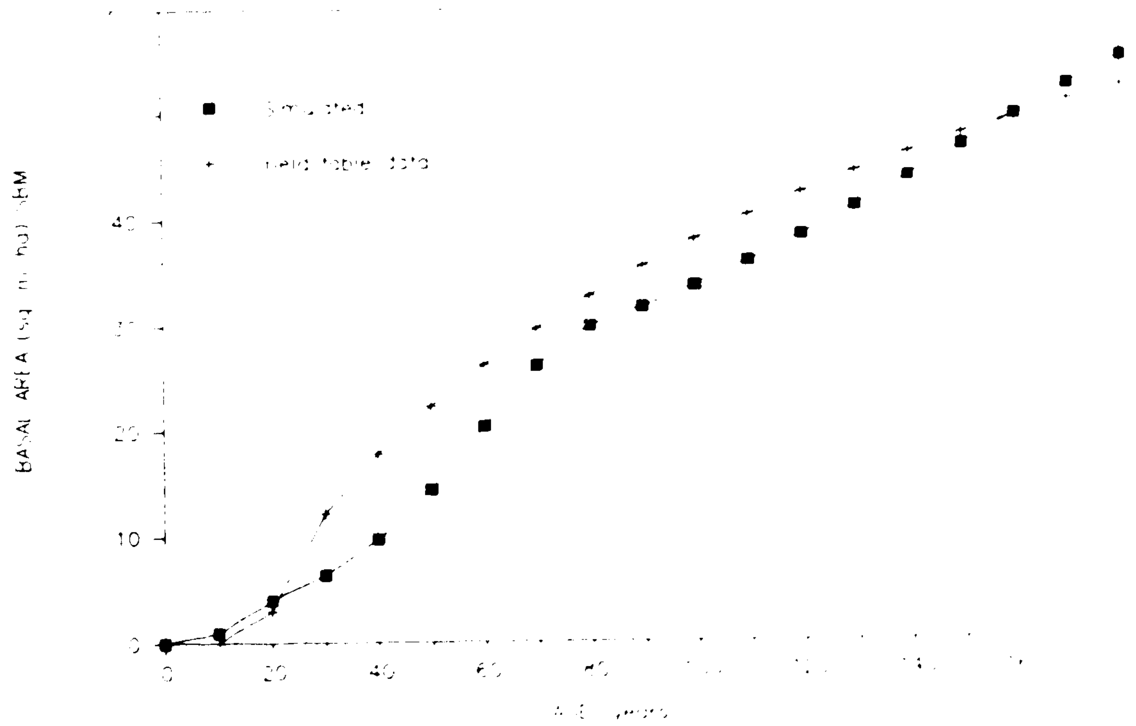


Figure C.26 Simulated and yield table basal area of black spruce on medium sites

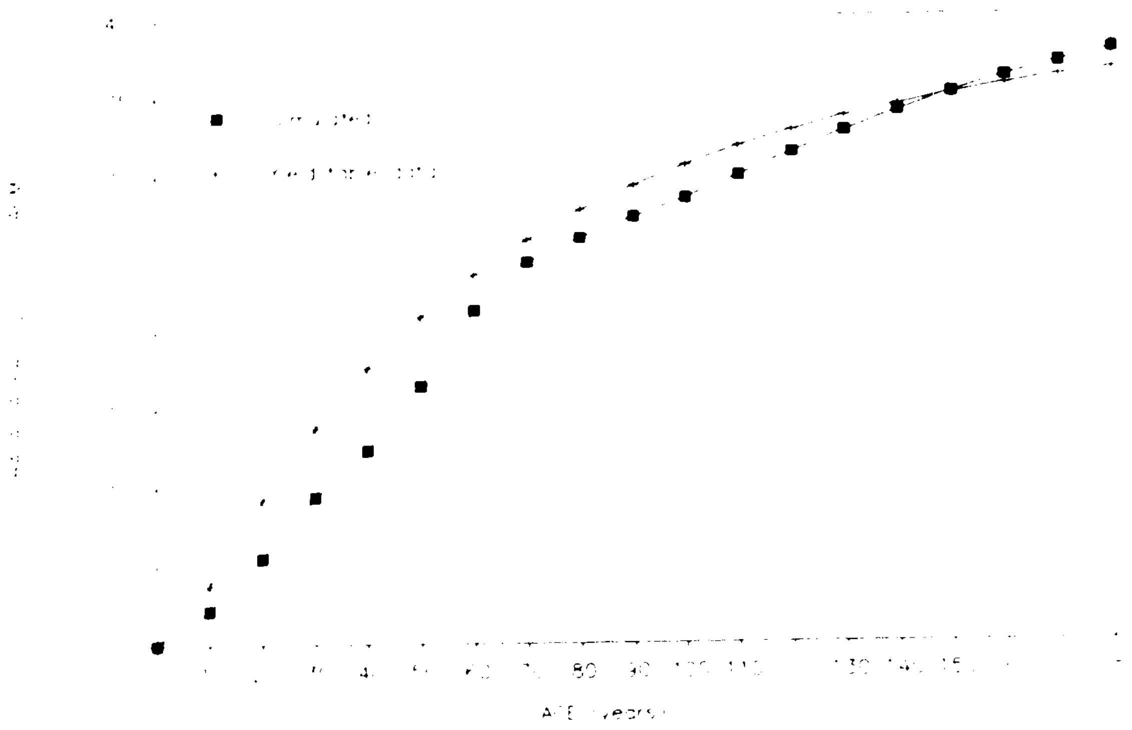


Figure C.27 Simulated and yield table basal area of aspen on medium sites

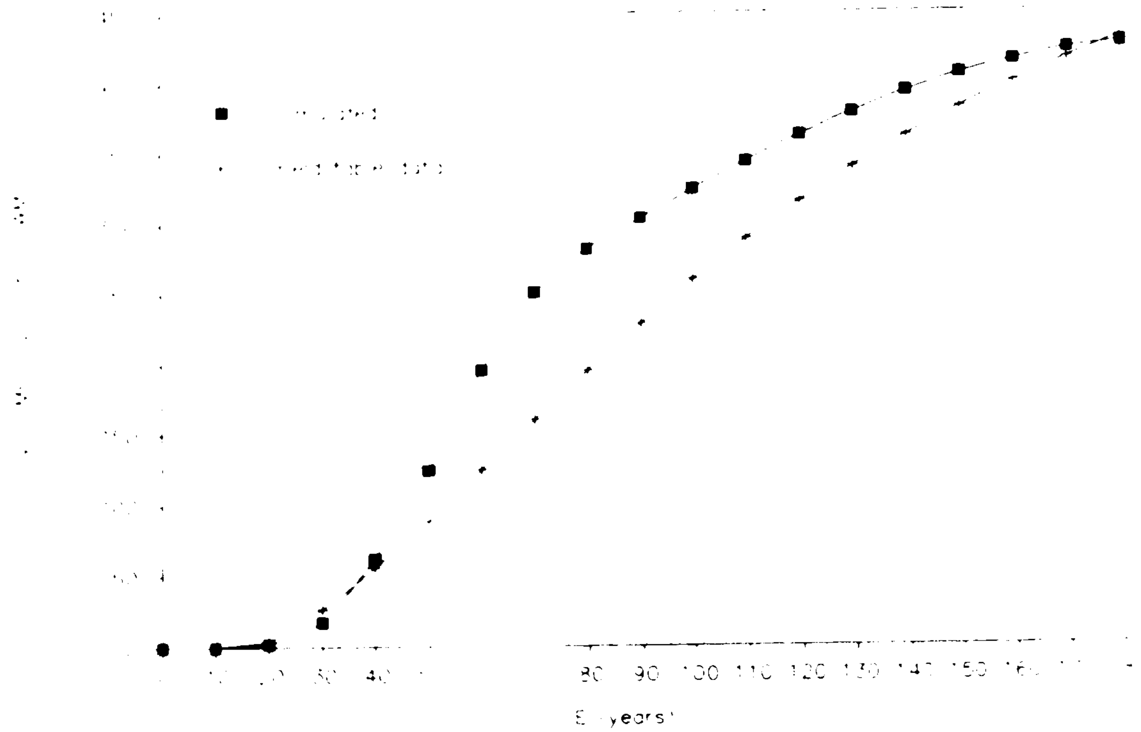


Figure C.28 Simulated and yield table gross volume of white spruce on medium sites

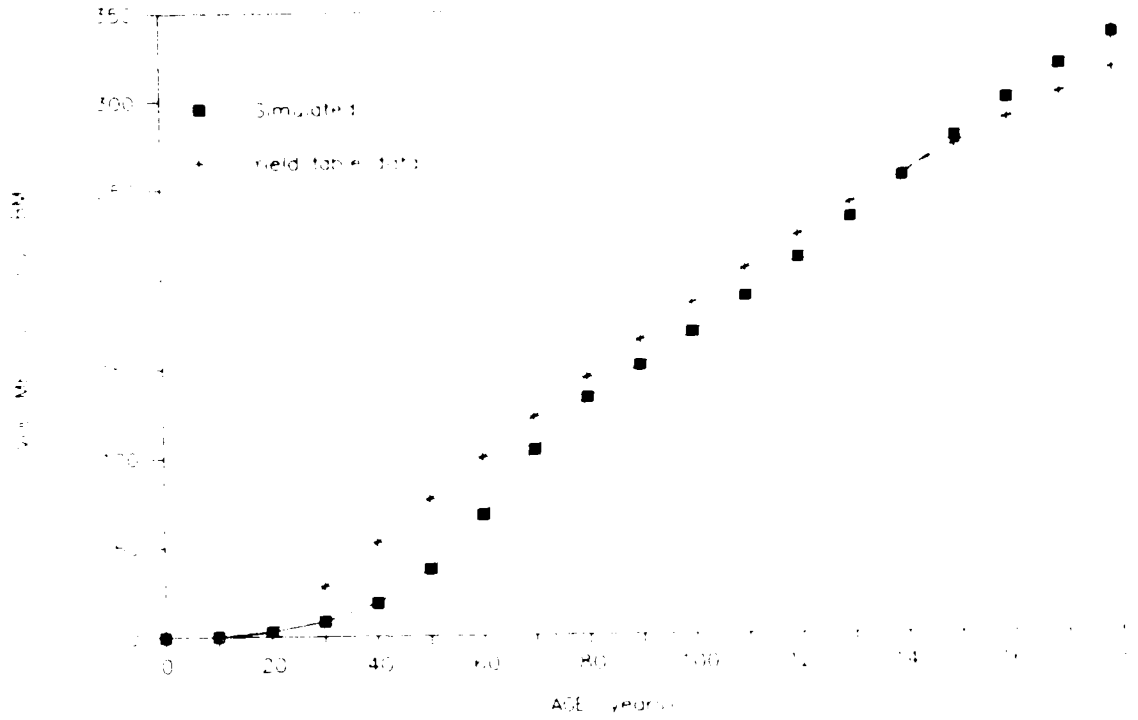


Figure C.29 Simulated and yield table gross volume of black spruce on medium sites

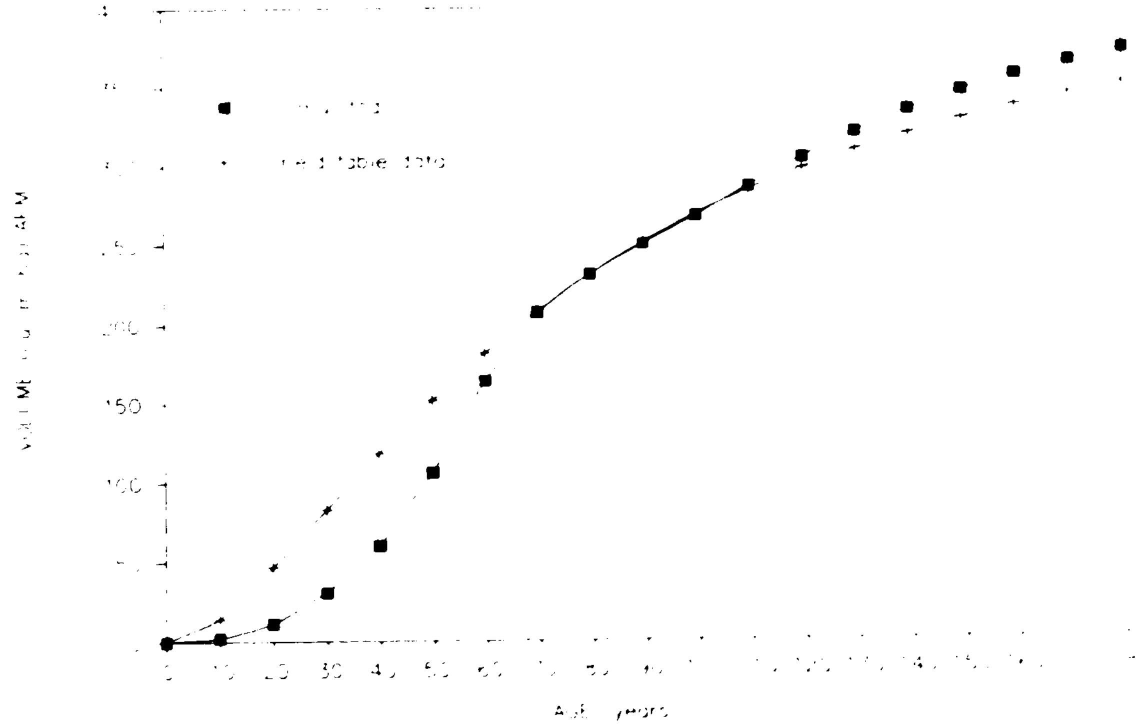


Figure C.30 Simulated and yield table gross volume of aspen on medium sites

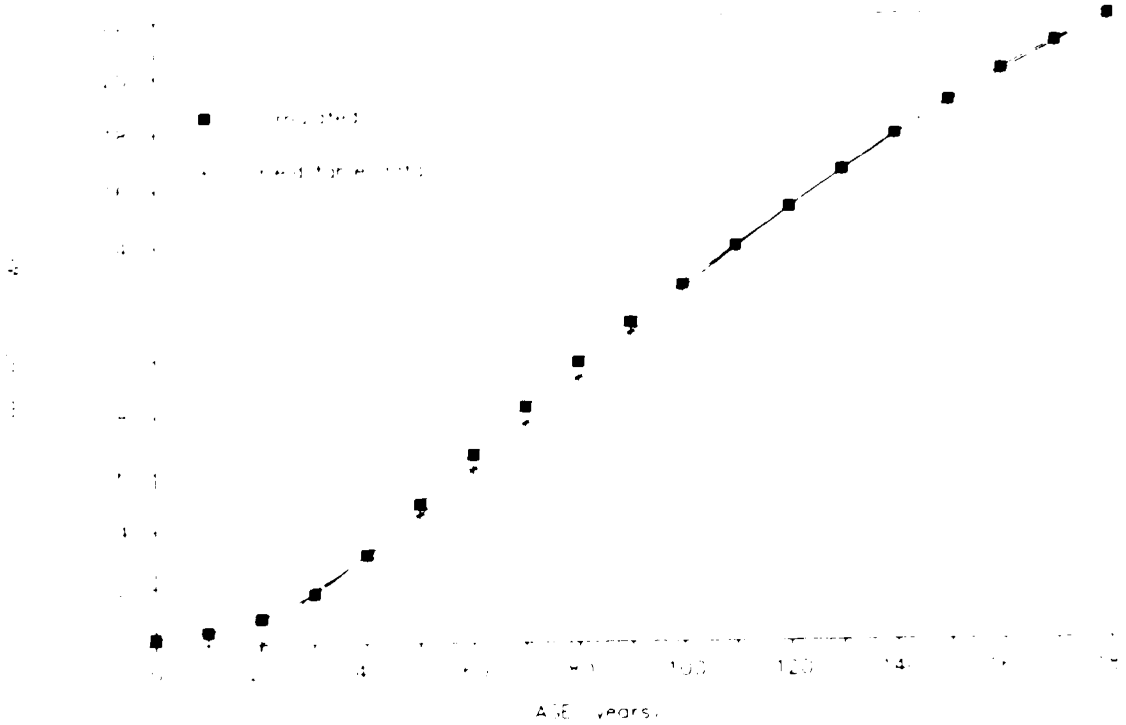


Figure C.31 Simulated and yield table site height of white spruce on fair sites

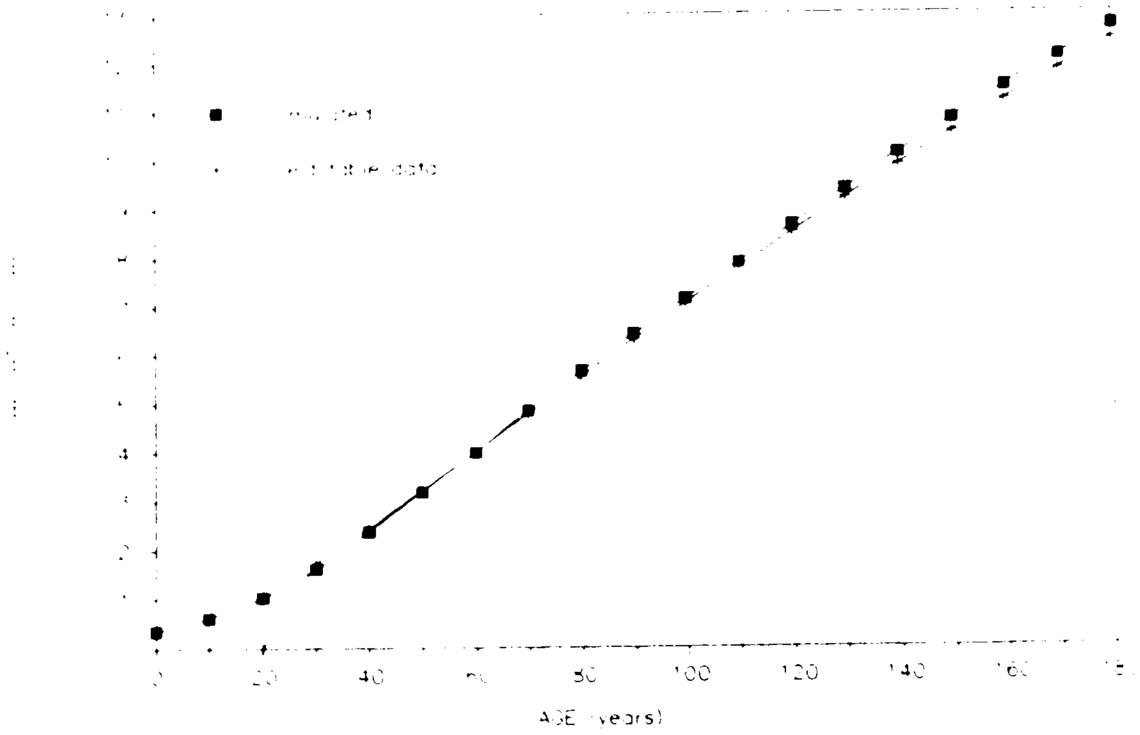


Figure C.32 Simulated and yield table site height of black spruce on fair sites

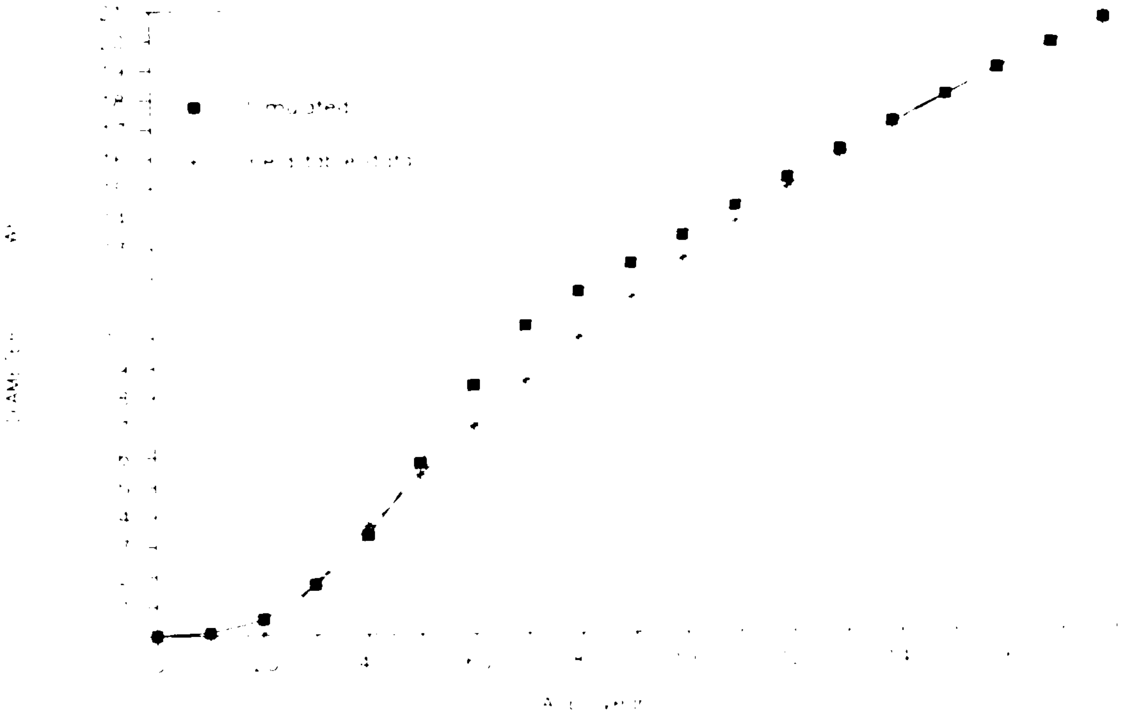


Figure C.33 Simulated and yield table site height of aspen on fair sites

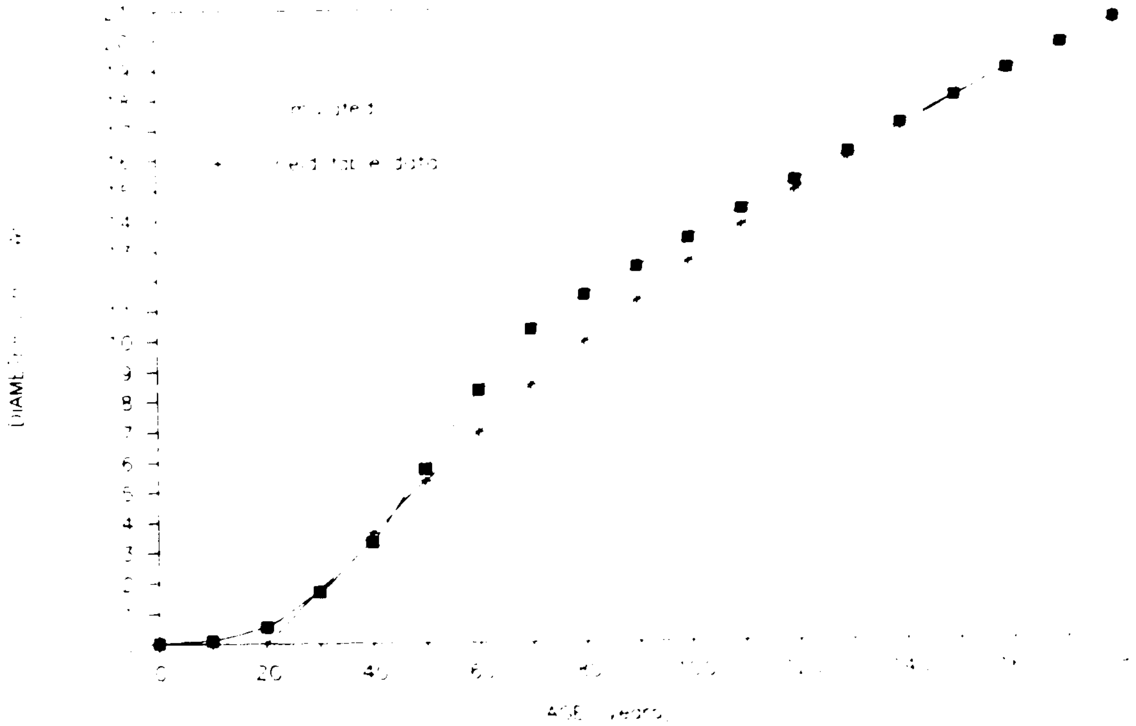


Figure C.34 Simulated and yield table average DBH of white spruce on fair sites

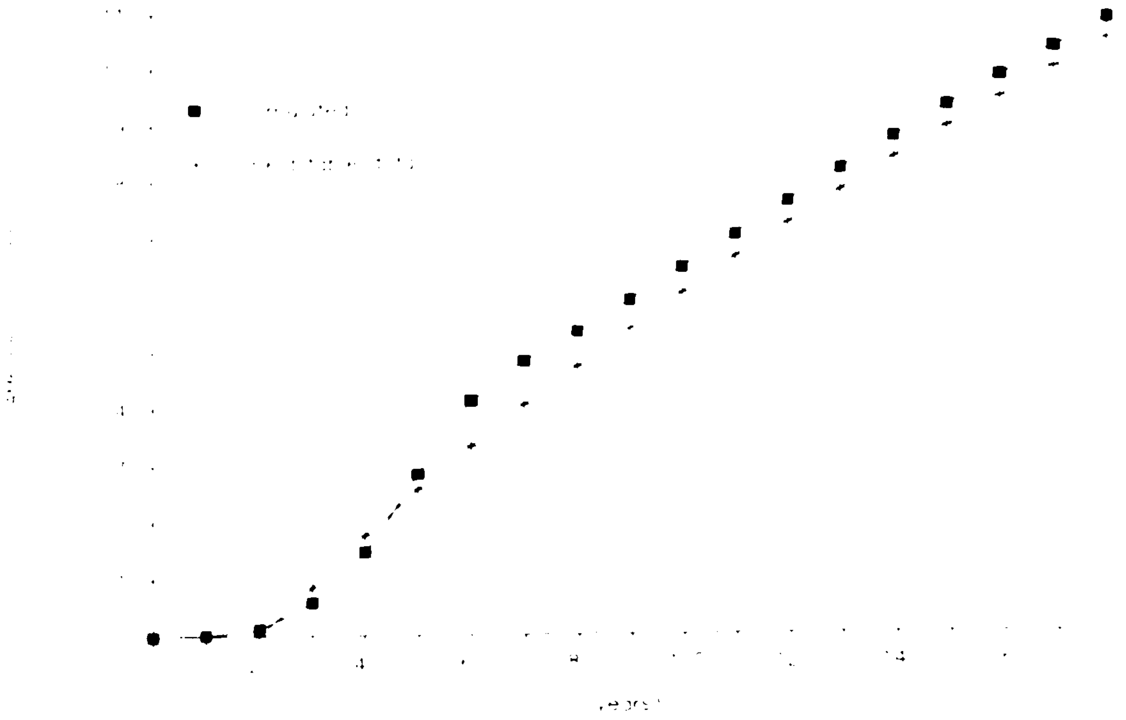


Figure C.35 Simulated and yield table average DBH of black spruce on fair sites

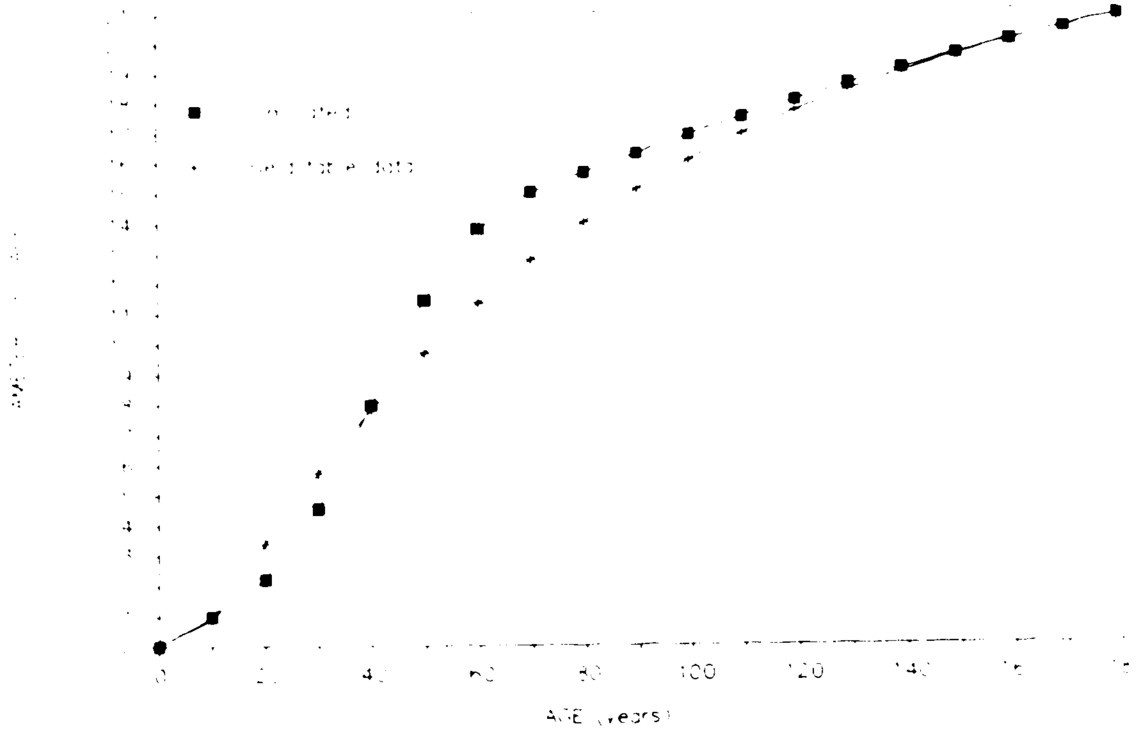


Figure C.36 Simulated and yield table average DBH of aspen on fair sites

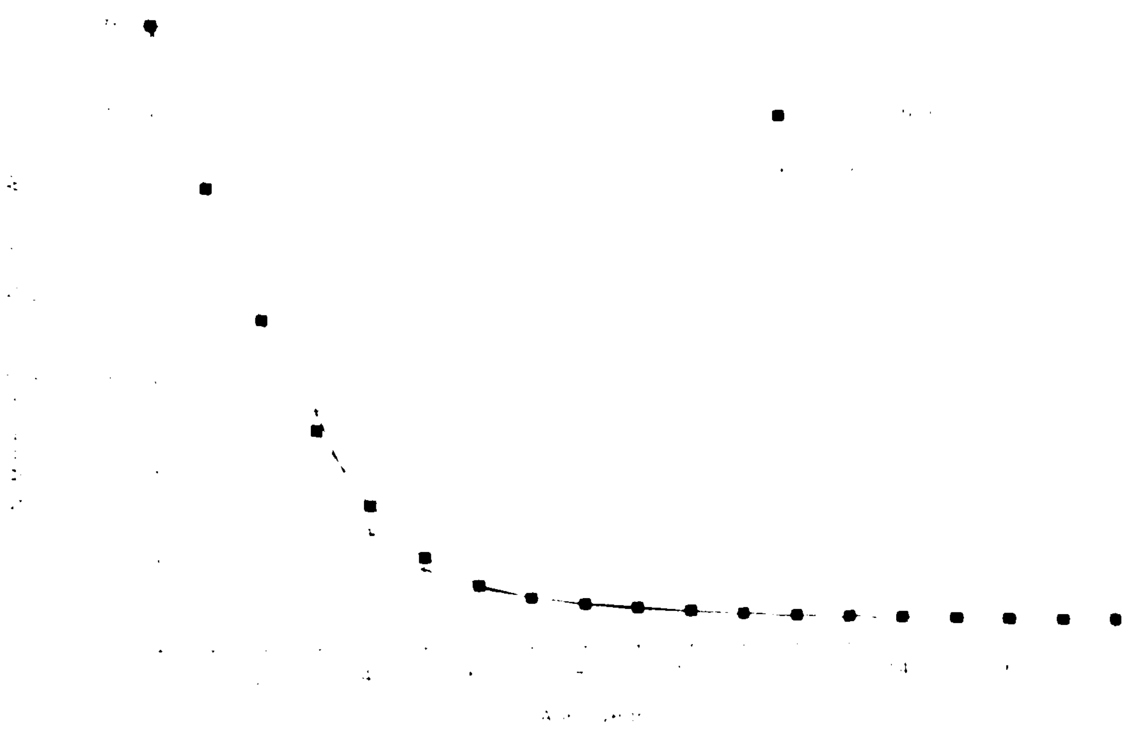


Figure C.37 Simulated and yield table number [illegible] of white spruce on fair sites

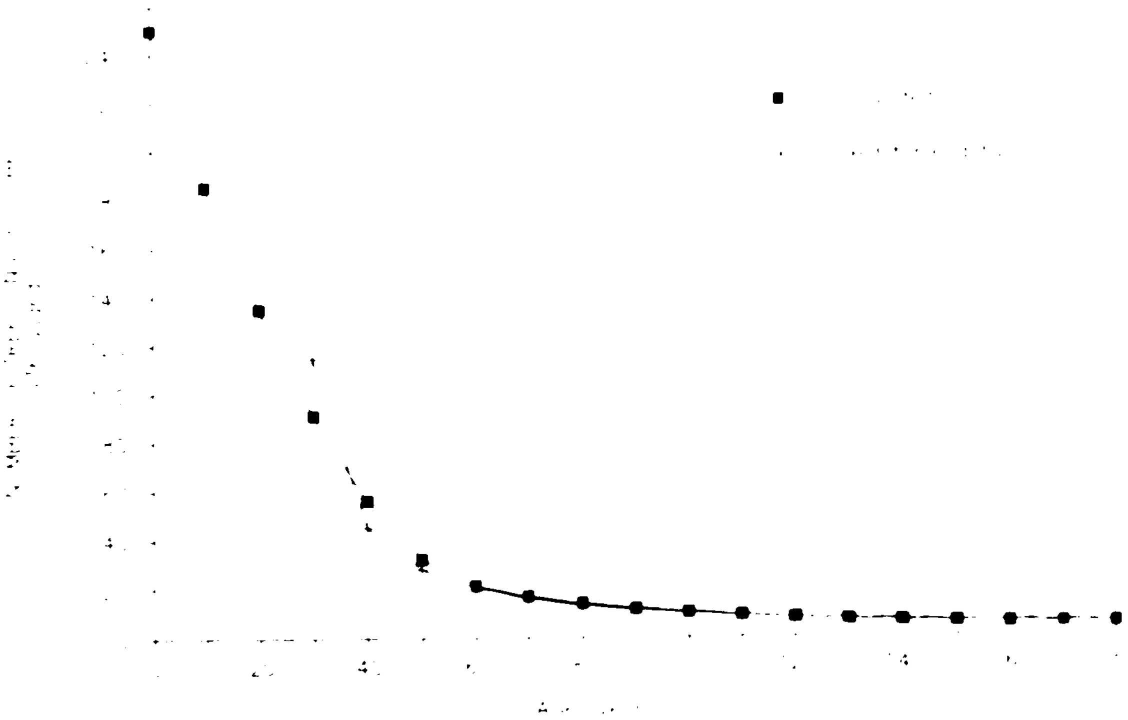


Figure C.38 Simulated and yield table number of trees of black spruce on fair sites

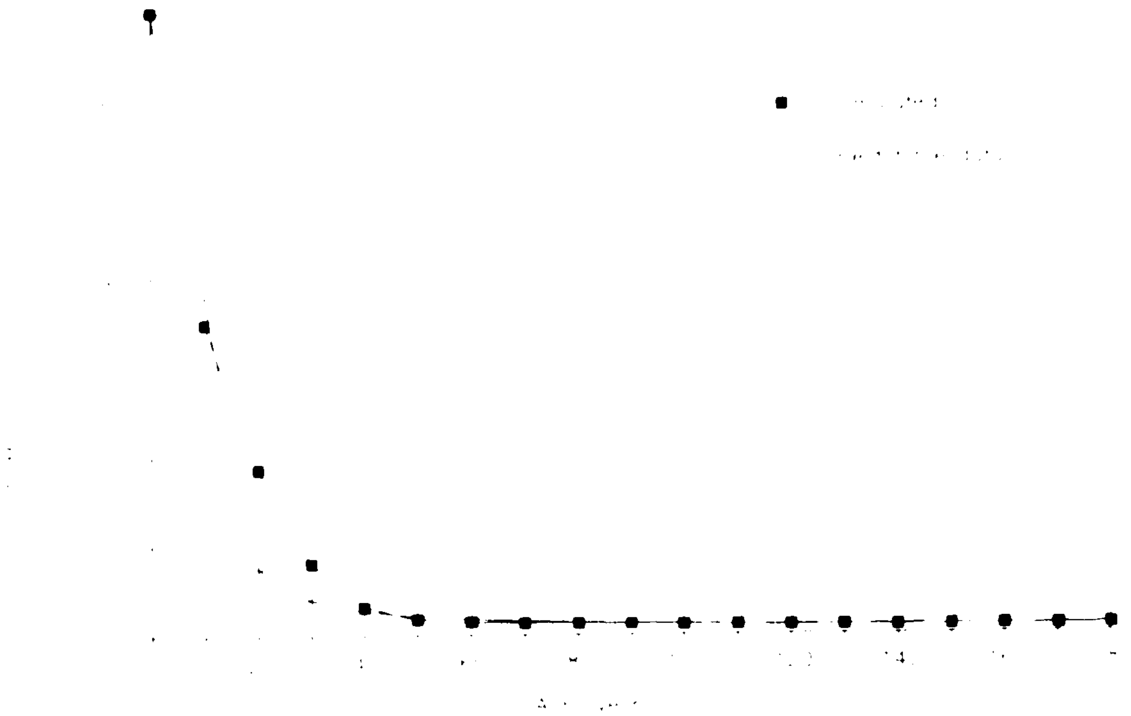


Figure C.39 Simulated and yield table number of trees of aspen on fair sites

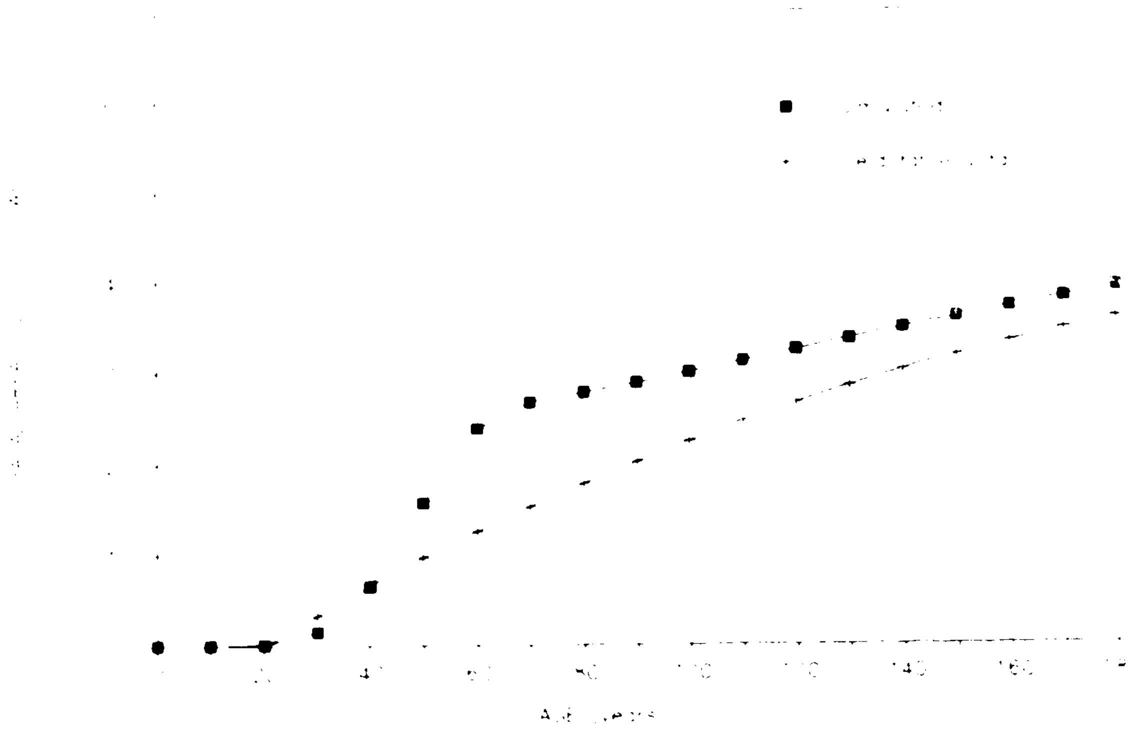


Figure C.40 Simulated and yield table basal area of white spruce on fair sites

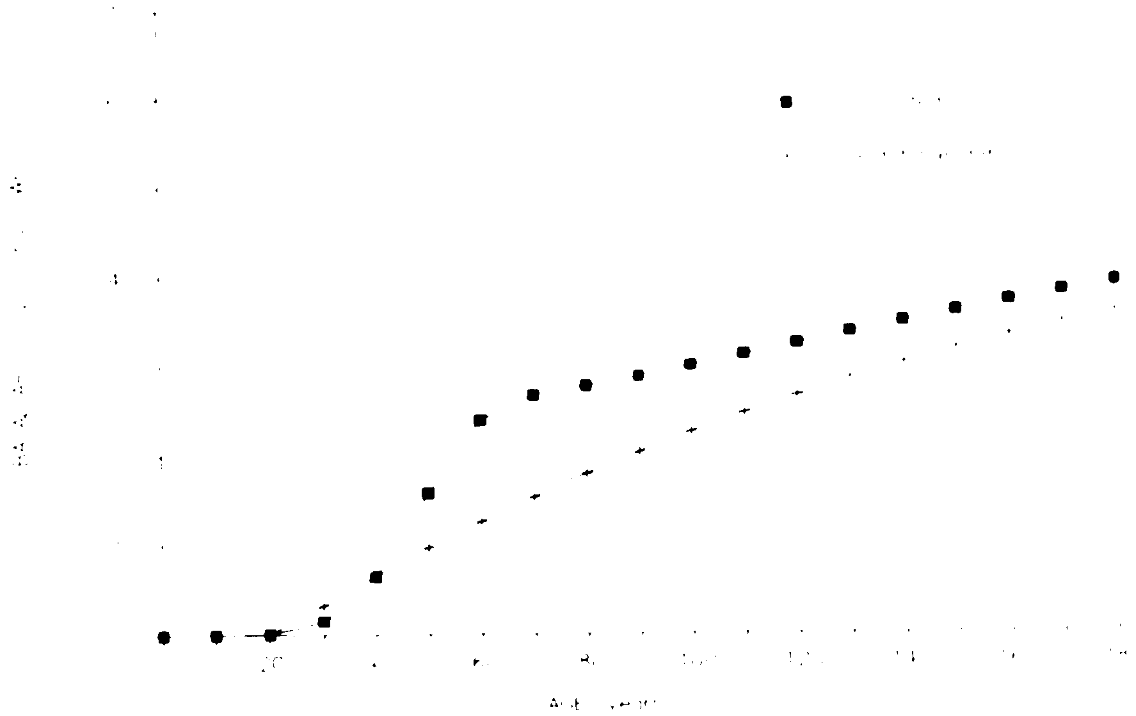


Figure C.41 Simulated and yield table basal area of black spruce on fair sites

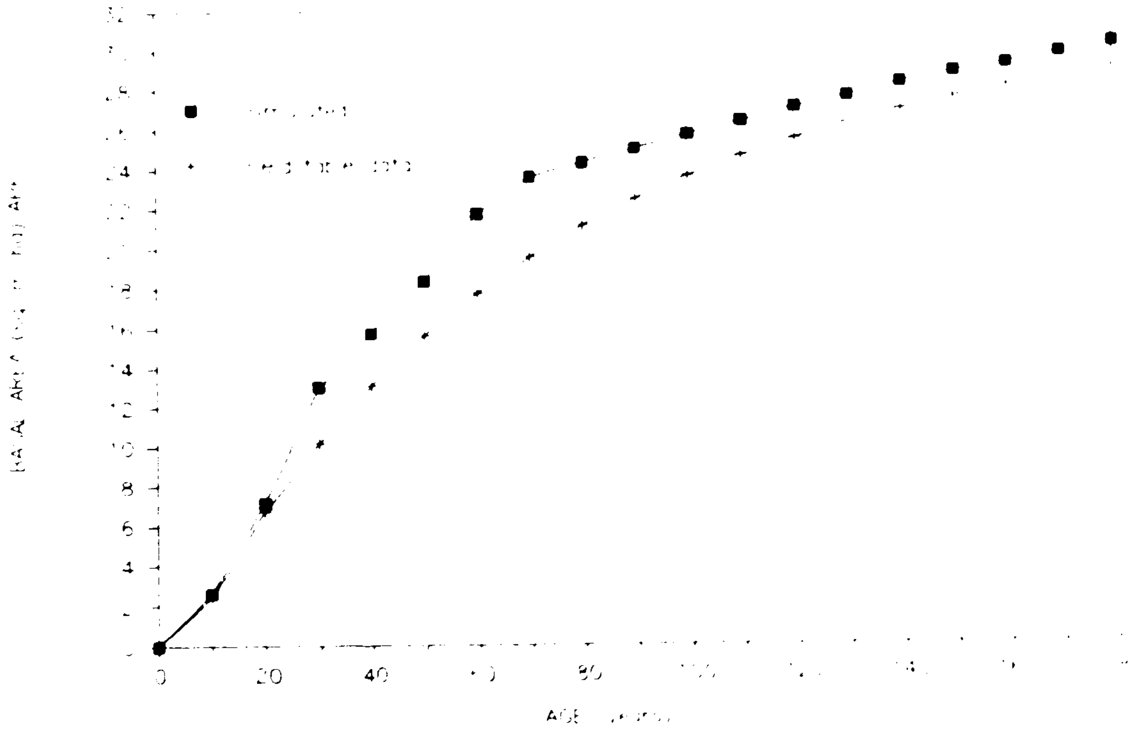


Figure C.42 Simulated and yield table basal area of aspen on fair sites

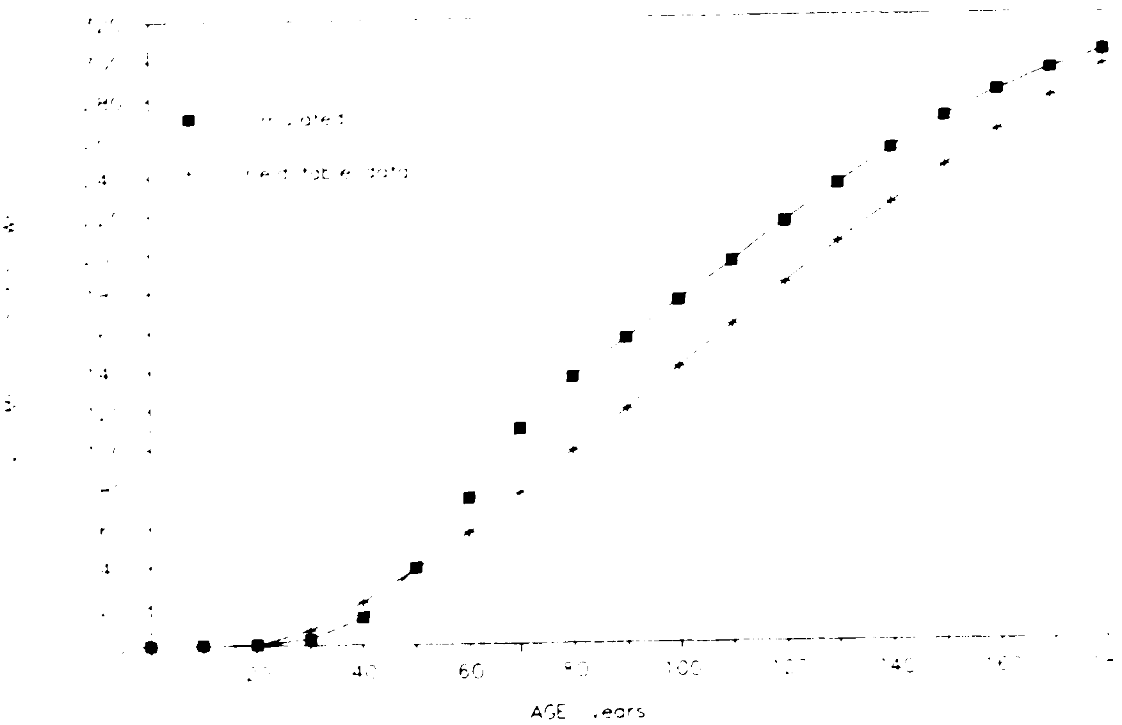


Figure C.43 Simulated and yield table gross volume of white spruce on fair sites

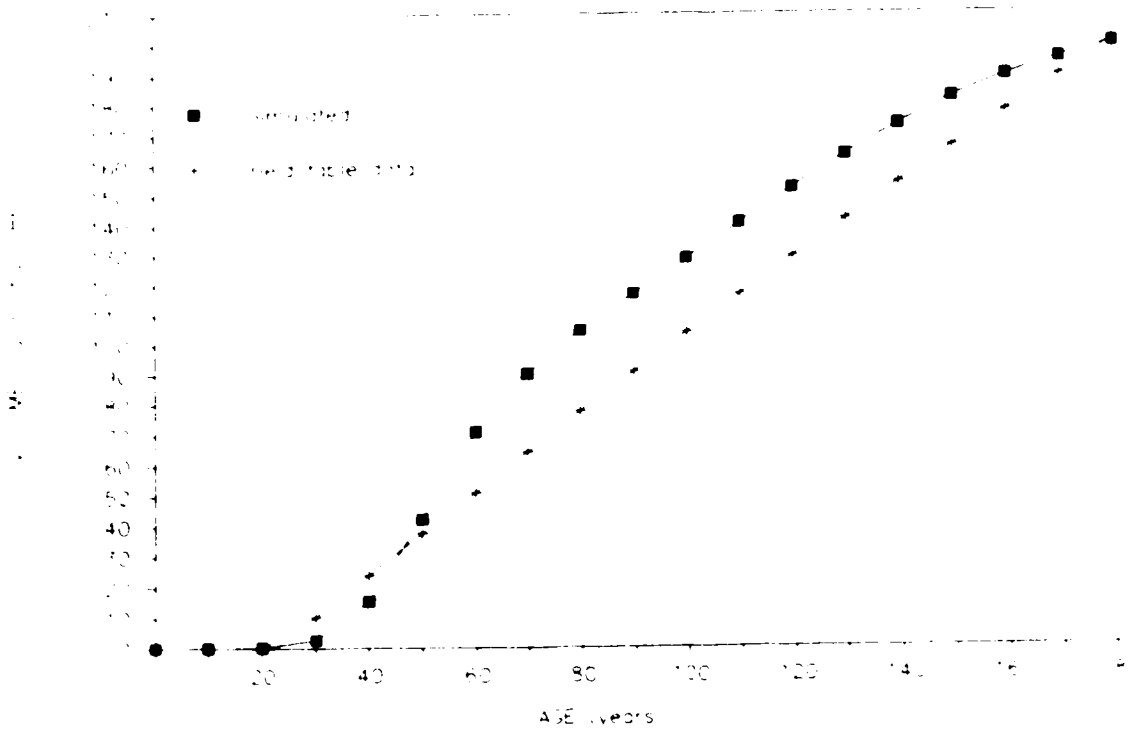


Figure C.44 Simulated and yield table gross volume of black spruce on fair sites

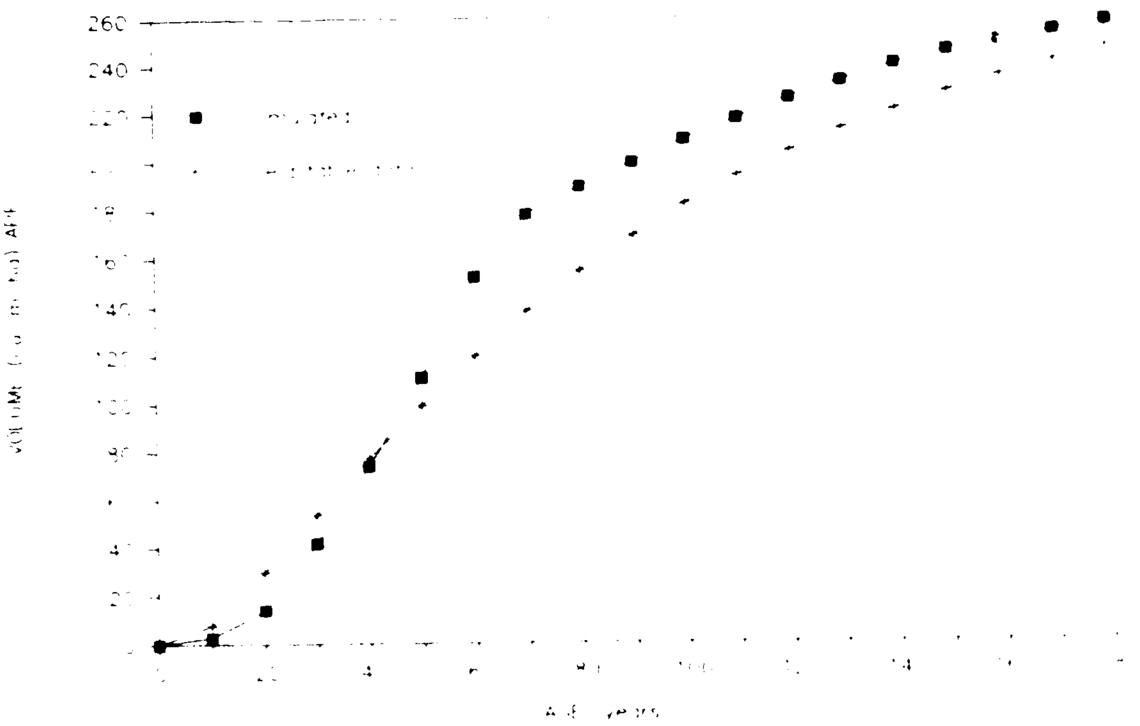


Figure C.45 Simulated and yield table gross volume of aspen on fair sites