Computer modelling of montane eucalypt forests

Michael James Strasser, B.Sc. (Hons)

A thesis submitted for the degree of Doctor of Philosophy
of The Australian National University

January, 1992

Declaration

The work presented in this thesis is my own. Specific contributions by others are referred to in the text and Acknowledgements.

Michael Strasser

M. J. Strasser

Ecosystem Dynamics Group
Research School of Biological Sciences
The Australian National University
Canberra

Acknowledgements

This study was undertaken while I was a postgraduate scholar in the Department of Environmental Biology, later the Ecosystem Dynamics Group, in the Research School of Biological Sciences, A.N.U., and was financially supported by a Commonwealth Postgraduate Research Award. I am grateful to Professors Barry Osmond, Ralph Slatyer and Ian Cowan, and Dr Ian Noble for allowing me to study in their department.

I am particularly indebted to my principal supervisor, Dr Ian Noble, for his guidance and support throughout the course of this project, with his ability to see through a problem with clarity. I find myself frequently reminded of the depth of his influence on my intellectual development.

I am also grateful to Dr Mike Austin who was generous with his expertise and field data. Field data were also made available by Emmett O'Loughlin, CSIRO (climatic data), and by the other contributors to the CSIRO vegetation database: L. Adams, ANU Forestry Department, D. Binns, W. Braithwaite, R. Breckwoldt, J. Briggs, S. Davey, S. Davies, L. Dovey, J. Dugin, S. Forbes, D. Gillieson, P. Gilmour, R. Good, C. Helman, J. Hibbert, F. Ingwersen, S. Kessell, J. Landsberg, D. MacKenzie, C. Margules, A. Nicholls, B. Outhred, E. Pook, R. Story, C. Wheeler and G. Yapp.

Andrew Moore deserves special mention for his willingness to read and comment on my work, and for stimulating discussions on a variety of many facets of ecology, modelling and science in general. Bruce Wellington also provided much scientific and moral support throughout the course of this project.

The Research School, and the Department in particular, provided a stimulating atmosphere in which to work. I will always remember the wide variety of topics of discussion (both scientific and not) with Ian, Bruce, Hugh, Jim, Peter, Jill, Doug, Jann, Andrew, Suzanne, Jeremy and many others.

I am also grateful to Professor Ian Ferguson and Dr Roger Sands of the Faculty of Agriculture and Forestry, University of Melbourne, who allowed me to continue work on this project while I was a member of the teaching staff in the Forestry Section. Similarly I am grateful Jones & Co and to Professor John Bain for their forbearance as I completed re-writing this thesis while in their employ in unrelated fields.

I have special thanks for my many friends in other facets of my life who enriched it while I was working on this project, particularly Pippa, Brian and Leonie.

Finally, I must thank my wife Louise, whose endurance has been tested to its limits by my necessary obsession with the re-writing of this thesis.

Abstract

BRIND is a computer model of eucalypt forests on the higher-altitude, sheltered slopes of the Brindabella Range, near Canberra. It belongs to a group of models called forest gap simulators that simulate dynamics on a small forest plot. The aims of the study were to make improvements to BRIND based on a detailed study of its structure and the predictions it makes, and to expand the range of environments in which it can make predictions.

A detailed examination of the structure of BRIND in relation to ecological theory and published data revealed unrealistic structure in some of its components and the interactions between them. A major concern with realism in BRIND is that reduction of tree growth to zero at the extremes of species' temperature (altitude) ranges is combined with limited temperature effects on establishment rates to produce a hidden assumption that radial growth is the primary determinant of species distributions along that gradient.

Field data were collected across a range of altitudes in the eastern side of the Brindabella Range and compared with predictions of BRIND. The model predicted too much biomass and too many trees at all altitudes. BRIND includes a submodel of stump sprouting which replaces most trees dying from any cause with vigorous saplings; after its removal the model was able to predict plausible biomasses and size distributions. With suitable adjustment of selected species parameters it is possible for BRIND to make plausible predictions of species distributions with altitude, but such adjustments do not address concerns about the lack of realism in some model components.

Tree-ring data from *Eucalyptus pauciflora* were used to investigate climatic and competition effects on growth. Contrary to model assumptions, variation in radial growth showed no response to year-to-year temperature variation; it was sensitive to soil moisture changes (not modelled by BRIND). Evidence for competitive effects of neighbouring trees on tree-ring width was found; these effects could be modelled by a simple one-sided index of basal area density.

A new growth submodel for BRIND was constructed according to the aims of the study: it does not reduce growth of all trees to zero at the edges of species' ranges of occurrence (as in BRIND). The new submodel includes an index of soil moisture, a simple competition index and an index of growing season length (temperature).

Regression models of species distributions along gradients of temperature and soil moisture were developed to be used for prediction of species establishment probabilities in a new submodel of establishment. It was found that regressions fitted to a large data set that encompasses most of the environmental ranges of the species were poor predictors of occurrence in the Brindabella Range. New regression models were fitted to data from only the Brindabella Range to be used in the new submodel.

BRIND predicts random, background mortality to occur at unrealistically high rates. When these rates were reduced to plausible values suppression mortality of intermediate-sized trees increased. Suppression mortality of trees in BRIND was found to be affected by the dynamic behaviour of the model: trees in even-sized cohorts grow to intermediate size before thinning takes place, but in mixed-size stands trees die from suppression at smaller sizes.

Fire response is predicted by BRIND to be only immediate or delayed mortality of trees. An alternate, three-stage submodel of response for fire-resistant eucalypt species was developed in which trees may be killed, have their stems killed, or suffer some reduction of above-ground tissue. The new submodel was parameterised using field data and predicted increased response with increasing size and decreasing vigour of trees. One stage of the submodel was compared with independent data and shown to predict plausible rates of stem death.

The new submodels were combined into a new model called NEWBRIND, whose predictions were compared with the field data from the Brindabella Range. NEWBRIND is able to predict plausible distributions of species on a range of altitudes and aspects in the Brindabella Range after adjustment of some species growth rate parameters. Plot biomass predicted by NEWBRIND is generally near to field values but does not show the same variation with altitude. NEWBRIND is able to predict dynamics of species replacement and biomass response in the Alpine Ash zone of the Brindabella Range.

Predictions by NEWBRIND are very sensitive to the values of species growth rate parameters, behaviour which revealed another hidden assumption that the outcome of competitive interactions always strongly favours faster-growing trees, even when the difference in growth rates is small. This behaviour appears to be unrealistic and is related to the treatment of suppressed trees in the model.

The new model makes predictions which are comparable to those made by BRIND but does so over a wider range of environments, and using some more realistic mechanism. The work in this study provides evidence that the gap model formulation is general at a level of qualitative prediction but each implementation needs to be carefully validated for the forests it models.

Contents

1 Introduction **1**
- 1.1 Modelling theory . . . 2
  - 1.1.1 Purposes of models . . . 2
  - 1.1.2 The modelling process . . . 4
  - 1.1.3 Model complexity . . . 10
- 1.2 Theories and models of succession . . . 11
  - 1.2.1 Succession as development . . . 12
  - 1.2.2 Successions of individuals . . . 14
  - 1.2.3 A hierarchical synthesis? . . . 15
- 1.3 Models of vegetation dynamics . . . 17
  - 1.3.1 Interactions between species . . . 17
  - 1.3.2 Species 'strategies' and attributes . . . 18
  - 1.3.3 Change-of-state models . . . 23
  - 1.3.4 Individual-tree models . . . 25
  - 1.3.5 Forest gap models . . . 26
  - 1.3.6 BRIND . . . 28
- 1.4 Response to environmental gradients . . . 29
  - 1.4.1 Environmental ordination . . . 30
  - 1.4.2 Niche and habitat . . . 31
  - 1.4.3 Physiological and ecological response . . . 32
- 1.5 Eucalypt forest dynamics . . . 33
- 1.6 The Brindabella Range . . . 36
- 1.7 This study . . . 39

2 Structure of BRIND **41**
- 2.1 Introduction . . . 41
- 2.2 Growth and competition . . . 44
  - 2.2.1 The growth equation . . . 44
  - 2.2.2 Maximum growth increments . . . 53
  - 2.2.3 Response to environmental gradients . . . 56
  - 2.2.4 Competition . . . 60
- 2.3 Establishment . . . 64
- 2.4 Mortality . . . 68
- 2.5 Fire response . . . 70

2.6 Sprouting 74
2.7 Conclusions 75

3 Comparisons with field data **79**
3.1 Field survey 79
3.2 Simulation methods 81
3.3 Total biomass 83
3.4 Numbers of trees 84
3.5 Species distributions 91
3.6 Removal of the SPROUT subroutine 95
3.7 Plot biomass 101
  3.7.1 Maximum biomass parameter (SOILQ) 101
  3.7.2 Light extinction coefficient (k) 103
3.8 Conclusions 106

4 Growth and competition **107**
4.1 Tree-ring study 108
  4.1.1 Climate and radial growth in *E. pauciflora* 108
  4.1.2 Coring methods 109
  4.1.3 Ring widths 112
  4.1.4 Effect of climate on growth 116
  4.1.5 Comparison with BRIND simulations 124
  4.1.6 Influence of other trees 129
4.2 A new growth submodel 138
  4.2.1 Soil moisture 138
  4.2.2 Temperature 146
  4.2.3 Competition 147
  4.2.4 Potential growth increments 151
  4.2.5 Some tests 156
4.3 Conclusions 159

5 Establishment **160**
5.1 Sensitivity to rate of establishment 161
5.2 Response to environmental gradients 165
  5.2.1 Species models 166
  5.2.2 Results 170
  5.2.3 Application to BRIND 171
  5.2.4 Local species models 175
5.3 Conclusions 177

6 Mortality **180**
6.1 Mortality rate curves 180
6.2 Size-independent mortality 183
6.3 Suppression mortality 187
6.4 Conclusions 193

7 Fire response 195
7.1 The Gudgenby fire 195
7.2 Three-stage response model 197
7.3 Data from Gudgenby 198
7.3.1 Scorch height 199
7.3.2 Vigour index 199
7.3.3 Summary of response 203
7.4 Statistical models 203
7.4.1 Mortality 205
7.4.2 Stem death 205
7.4.3 Growth recovery 209
7.5 Testing of new model 209
7.6 Conclusions 213

8 A new model 214
8.1 Construction of NEWBRIND 214
8.1.1 Site description 215
8.1.2 Growth 217
8.1.3 Establishment 221
8.1.4 Mortality 222
8.1.5 Fire response 223
8.2 Preliminary model tests 224
8.2.1 Species distributions 225
8.2.2 Biomass 231
8.2.3 Model dynamics 233
8.3 Conclusions 235
8.3.1 NEWBRIND 236
8.3.2 Observations about modelling 240

Bibliography 244

A Field data information 257

B Site parameters for simulations 261

C Listing of NEWBRIND 263
C.1 Header files 263
C.2 Main program 273
C.3 Biological simulation functions 278
C.4 Service functions 290
C.5 Input/output functions 307

Chapter 1

Introduction

This thesis describes a study of a computer model called BRIND that simulates dynamics of eucalypt forests in the Brindabella Range near Canberra in the Australian Capital Territory. BRIND is one of a group of models that simulate establishment, growth and death of trees on a small forest plot called a *gap* (Shugart and West 1979). BRIND was developed by Shugart and Noble (1981) from FORET, a model of the forests in the Appalachian Mountains of the eastern U. S. A. (Shugart and West 1977), which was itself developed from the first gap model JABOWA of the Hubbard Brook ecosystem in the north-eastern U. S. A. (Botkin *et al.* 1972).

The aims of this study were to make improvements to BRIND based on a detailed examination of its structure and predictions, and to expand the range of environments in which it can be used. I was interested to examine the model in detail to investigate its relationship with ecological theory and published data. I particularly wanted to examine its prediction of dynamics in eucalypt forests because much of its structure is identical to the North American forest model from which it was derived. BRIND is restricted to making predictions of dynamics on higher-altitude, sheltered slopes of the Brindabella Range (Shugart and Noble

1981). I also sought to expand both the range of altitudes in which the model predicts dynamics and the range of aspects to include more exposed slopes.

In this chapter I provide background to modelling theory, forest succession models, response to environmental gradients, dynamics in eucalypt forests and a description of the Brindabella Range. The discussion of modelling theory describes types of models and the modelling process with reference to BRIND and other gap models.

1.1 Modelling theory

A model is a representation of some part of the real world of interest to the modeller and serves to answer questions about the real world. I will use the word *system* to refer in general to the part of reality chosen by the modeller.

Kac's (1969) metaphor of models as 'caricatures of reality' is illuminating: a model emphasises parts of the system of most importance to the modeller's purpose—perhaps even distorting them—and treats some other parts in much less detail, sometimes ignoring them completely.

1.1.1 Purposes of models

The essential determinant of a model's design is the purpose for which it is to be used. In general, a model is built to make predictions about the behaviour of a system, or as an aid in explaining the mechanisms by which the system functions. Consequently two broad categories of model can be recognised: *predictive* and *mechanistic* (explanatory), but models may be used for both prediction and explanation.

A model that is required only to predict system behaviour under certain conditions or at some time in the future (or past) may be constructed with no knowledge of how the system of interest functions. Such purely predictive models (also called correlative or empirical models; Solomon 1979) are exemplified by 'curve-fitting' exercises such as regression.

In contrast, mechanistic models embody theories of how the system of interest functions. These models (also called rational models; Solomon 1979) are used to test our knowledge about the real world. The model is used to generate predictions that can be tested against data from the real world. These predictions can be treated as hypotheses to be falsified by experiment or field observation. Falsification of a generated hypothesis (model prediction) leads the modeller to reconsider theory expressed in the model (Swartzman 1979).

Dale *et al.* (1985) classified a number of forest tree growth models into 'forest growth' and 'community dynamics' models. Models in the former category simulate tree growth using equations derived from regression studies of forest growth: the parameters in their equations are chosen for 'goodness of fit', not for their biological meaning. Models in the latter category use equations that express ideas about relationships between particular biological parameters.

Similarly, Loehle (1983) classified ecological models into theoretical and predictive models (a third type—logical models—was described, but is rarely found in ecology). He further divided predictive models into application models and calculation tools: in application models, established laws and theories are applied to solve a particular problem; in calculation tools, model construction is unrelated to the structure of the system under study.

Models also serve another purpose: they facilitate communication of ideas by formalising concepts into precise (usually mathematical) terms (Swartzman 1979). A successful explanatory model of a complex system aids communication by highlighting important mechanisms in that system.

1.1.2 The modelling process

In this section I discuss the important steps in constructing a model. The steps I describe pertain particularly to explanatory models because they are more complex than calculation tools.

Modelling is an iterative process, whose major steps are model design, evaluation and refinement. Once the problem to be investigated has been defined (i.e., the model's purpose determined) a model is first designed, then evaluated and refined as necessary according to the results of evaluations. After refinement, further evaluation takes place and the model is refined again as necessary. Evaluation and refinement can continue until a satisfactory result is achieved in terms of the objectives of the modelling exercise.

It is important to decide the ***domain of applicability*** of a model as part of its definition. A model's domain is the set of conditions for which the model is intended to match reality.

Design

Broadly, model design is the process of translating what is known about the system of interest into a model, given overall objectives. The step is also referred to as abstraction (Solomon 1979), where a mathematical (often computerised) representation of some part of the real world is made.

Simplification is an important aspect of modelling—the real world is very complex—distilling the essence of a problem to answer specific questions. Thus a model must reproduce the most *important* aspects of the system of interest. It is desirable to choose a parsimonious model: one having the simplest possible structure that will answer the questions required of it.

Solomon (1979) described a number of interdependent components in the design process: supposition, idealisation, assumptions, choice of tools and choice of symbols. *Supposition* is the primary hypothesis being investigated and is a formalisation of modelling objectives. The supposition governs choices made in all steps of the modelling process. The supposition of gap models is that forest successional dynamics can be modelled using a small set of equations describing characteristics of tree growth, competition and demography (Shugart 1984).

Idealisation and *assumptions* are simplifications made in model design. It is valuable to distinguish between them: idealisations are known to be invalid, but are known (or at least expected) not to seriously affect model behaviour; but assumptions are of unknown validity. The validity of assumptions should be examined as part of model evaluation: they may need to be relaxed or altered as the model is refined. One idealisation in BRIND is that the relationship between a tree's diameter at breast height (DBH) and its height is constant for a species: this is known not to be true but expected to have little effect on predicted forest dynamics. It is assumed in BRIND that radial growth of trees of a species is severely reduced by unfavourable temperatures towards the limits of its field distribution: in Chapter 2 I question the validity of this assumption.

The *choice of tools* in a model refers to mathematical formulation: for example, analytic vs. simulative or deterministic vs. stochastic. The development of high-speed computers has facilitated the use of simulation models; such models can be used to implement analytically intractable representations of biological phenomena, such as difference equations for tree volume growth (Dale *et al.* 1985). If a system cannot be adequately represented by a deterministic model, it may be necessary to include some stochastic components to represent unexplained variation. For example, mortality of forest trees varies according to their size and status in the forest canopy but there is apparently random mortality from exogenous disturbance such as wind-throw (Shugart 1987) which is unrelated to tree size. These effectively random events are often simulated using a stochastic model component (Franklin *et al.* 1987).

The *choice of symbols* (called *representation* by Loehle 1983, 1987) is primarily governed by the specific modelling objectives. Ecosystem models like those described by Waring (1989) use symbols of matter and energy and matter content of ecosystem compartments and of rates of transfer between them and need not use symbols of individual tree species, size etc. By contrast, BRIND and other gap models predict dynamics of individual trees on a small plot and use appropriate symbols, for example tree DBH, height and growth rate.

Levins (1966) described compromises which are commonly made when designing models. He stated that while it is desirable to maximise generality, realism and precision, one of these three attributes is frequently sacrificed to the other two. *Generality* (also called robustness; Solomon 1979) is the ability of a model to make acceptable predictions over a wide range of conditions (parameter values), *realism* is the extent to which the structure of a model replicates that of the system it represents and *precision* is the detail with which a model makes its predictions.

The forest growth models described by Dale *et al.* (1985) are built to make precise predictions of growth in a particular forest stand for management purposes. These models are constructed using site-specific information and may not be very general in their application. In contrast, community dynamics models (Dale *et al.* 1985) are built for the purpose of investigating the theoretical concepts upon which they are based. These models may be less precise than forest growth models but are designed to be both more realistic and more general.

Forest gap models make precise predictions of numbers and sizes of trees on the simulated plot. In comparison the FATE model of Moore and Noble (1990) makes less precise predictions, expressing abundance on a discrete quantitative scale and the vital attributes model of Noble and Slatyer (1980) is still less precise, predicting only the presence or absence of functional groups of plants.

Evaluation

Evaluation is the process of assessing a model's *accuracy* in representing the system of interest. The choice of methods for model evaluation is dependent primarily on modelling objectives. The most common method is validation or corroboration: comparison of model predictions with data from the real world (Caswell 1976). These comparisons can be made statistically or graphically. For simple calculation tools such comparisons may be the only appropriate method of evaluation, but for mechanistic models used to explore theoretical ideas, other methods must also be used (Swartzman 1979).

Mankin *et al.* (1977) made the distinction between verification and validation of models. *Verification* is the comparison of model predictions with the primary data set used to provide parameter values for the model. It functions mainly as a test of the consistency of the model and how well its component equations express relationships found in those data. *Validation* is the comparison of model predictions with independent data within the model's domain of applicability. It is a more stringent test of how well the model represents the system of interest. Shugart and Noble (1981) were able to validate BRIND by comparing its predictions of stocking density and average size of trees with those in an independent data set.

But such a validation only shows that a model can make acceptably accurate predictions under one set of conditions. Cale *et al.* (1983) state that models should be tested at the limits of their domain of applicability. Such tests provide more information about model behaviour and indicate that the model's domain matches the range of conditions for which it was designed.

It is possible to construct alternate models of a system that make identical predictions but contain different assumptions. Solomon (1979) gives an example of two such models whose assumptions are actually contradictory. Such models need to be evaluated in more detail.

Swartzman (1979) described a number of additional evaluation methods for simulation models: model recalibration, sensitivity analysis, evaluating supporting rationale, key process identification and simulation experiments. *Model recalibration* is the adaptation of a model to a different domain of applicability. It is a test of the generality of a model formulation: if a model can be recalibrated by changing some parameter values while maintaining the same structure. it is a robust model (Solomon 1979). A feature of forest gap models is the robustness of their overall structure: the same general model structure has been used successfully to address a number of different problems in a wide range of forest types (Shugart 1984).

Sensitivity analysis and *simulation experiments* are methods of examining model behaviour under changing parameter values. In a sensitivity analysis selected parameters are varied and their effects on model predictions measured. It is valuable in pin-pointing which parameters exert most effect on model predictions: accuracy in determining their correct values is paramount in securing accurate model predictions. Simulation experiments may involve comparison of two alternate model formulations, experiments with perturbation of the model system, or just 'playing' with the model to learn about its behaviour. Dale *et al.* (1988) describe procedures for carrying out sensitivity analyses on a forest gap model like BRIND.

Key process identification is similar to sensitivity analysis, but whole processes within the model (rather than just parameter values) are assessed for their influence on model predictions. In both cases and especially with complex models having many parameters, it is necessary to decide where to look for parameters and processes that might be important.

Evaluating supporting rationale is perhaps the most important evaluation technique: it is a thorough review of the completed model with reference to the state of knowledge about the system of interest. Such examination, in the light of model objectives, should guide the modeller as to which parameters and processes in the model are likely to be the most important to model behaviour (their importance can then be tested using sensitivity analysis and key process identification).

It is desirable to subject a model to all these evaluation methods, which are interdependent. It might appear that if a model makes inaccurate predictions, other evaluation methods are of little relevance, but Swartzman (1979) argues that they can still provide important insights into model behaviour and may reveal the cause of the inaccuracy.

Refinement

Model refinement may be simple adjustment of parameters, or reformulation of some or all of a model. Model evaluation results may indicate that a model needs more complexity to make accurate predictions or alternatively that a model can be simplified by removing unnecessary components.

Loehle (1983) provided a valuable warning against model 'tuning': the adjustment of parameter values in a model so that its predictions match more closely a particular set of observations of the real world. If the parameters concerned have a theoretical basis, their relationship with theory will be broken by the change in parameter values. Loehle states that any tuning of a theoretical model, or the law content of an application model, results in its conversion to a calculation tool. In essence, this means that a model that is designed to be mechanistic but is tuned during its construction may not actually represent that mechanism when used for prediction. Under these circumstances, inferences made about the power of the model to explain system behaviour (more than just predicting it) become invalid.

1.1.3 Model complexity

Mechanistic models of complex systems are complex themselves. Components of such models are submodels that represent structures or processes in the system.

Complex models may have a hierarchical structure, in which each submodel represents a 'subsystem' that functions at smaller temporal and spatial scales (hierarchical structure is not necessary: a modeller may divide a system into subsystems that operate at the same spatial or temporal scales). Hierarchical structure is appropriate to ecological models, given the value of a hierarchical approach to ecosystem structure and function (Shugart and Urban 1988). In a complex model, predictions of each of the submodels become part of the mechanism of the 'parent' model, whose predictions may be of ultimate interest to the modeller.

The choice of components to use in a complex model is part of the design process (choice of symbols or representation, Section 1.1.2) and reflects the modeller's understanding of system function. Simplification is often achieved by not modelling structures or processes whose effect on overall model behaviour is minimal. For example, a model predicting primary production of forests will not need to include submodels of small mammal populations unless they are believed to affect tree population dynamics.

Realism of separate model components will vary, as it does for whole models. Calculation tools may be used to simulate the behaviour of some components while others are modelled in some mechanistic detail. Also, a model that has been transferred from one domain to another may retain a submodel that makes plausible predictions, but for the 'wrong reasons' (the mechanism it represents is not appropriate to or important in the new domain); this results in a loss of realism.

Gap models are complex models with hierarchical structure: they predict forest dynamics at the level of a forest gap from tree-by-tree mechanisms that operate at lower levels. They also have non-hierarchical structure where individual tree processes are modelled with some independence; these submodels can be varied or replaced as necessary (Shugart 1984).

1.2 Theories and models of succession

In this section I turn to a discussion of successional concepts, particularly as they apply to forests. In Section 1.3 I discuss important models of vegetation dynamics, linking them to successional concepts and modelling theory.

Observed patterns of temporal change in natural systems have given rise to a number of concepts of succession. The word *succession* generally refers to sequential changes that take place after some disturbance, or between disturbances. Primary succession refers to sequences of changes after massive disturbance that disrupts soil structure and secondary succession refers to sequences that occur after disturbance of biotic components of ecosystems.

Succession is a central idea in ecology and much study has been aimed at finding universal laws to explain the phenomenon. Differences in approaches to succession reflect differences of ideas about where such universal laws will be found. There has been a vigorous debate about the true nature of succession for many years and ecologists continue to strive to reliably predict successional phenomena, at least partly because the ability to predict is seen as a hallmark of a mature science (McIntosh 1980).

In general, theories about the nature and causes of succession have developed from two divergent points of view: a *holistic* view of succession as the development of a large-scale entity; and a *reductionistic* view of succession as the effects of changes in individual plants.

1.2.1 Succession as development

> The developmental study of vegetation necessarily rests upon the assumption that the unit of climax formation is an organic entity ... Succession is the universal process of formation development. Clements (1916, p. 3)

The concept of succession advanced by F. E. Clements (1916) early this century has dominated discussion of the subject for many decades. Its essential components are that succession is the development of a 'supra-organism', and that succession is a universal process that is basically deterministic. On the former point, active discussion still continues; on the latter, there is perhaps universal agreement that succession is very complex and difficult to predict (see, for example, McIntosh 1980).

To Clements, a disturbance is followed by an orderly sequence of communities (a sere) that replace one another by the principal processes of competition and reaction. Reaction occurs when one community type, by its presence, modifies the site (through shading and the medium of the soil) and makes it less suitable for itself, but more suitable for the next in the sere. The composition of a particular climax formation is determined entirely by climate; once reached, a climax can persist until disturbed, when a new succession begins.

Clements (1916) recognised the complexity in vegetation structure, seeing pattern as small-scale successions set in train by recurrent disturbance. But he believed that the "development of the formation as an organism" is "the motive force in succession" (p. 7), and that by rigourous and conscientious study the true successional nature of any community could be elucidated.

Whittaker (1953) assessed the climax concept and found it to be a 'rule' proved by a large number of exceptions, and argued that Clements' complex nomenclature was a way of forcing the variety of patterns found in nature into his scheme. Whittaker rejected the idea of a strict monoclimax determined by climate. He re-defined the climax as a useful but essentially arbitrary concept, referring to a dynamic steady-state of vegetation whose composition is determined by species' responses to environmental gradients.

Egler (1954) believed that succession is a phenomenon of community development, but he criticised the assertion by Clements that successions comprise discrete stages of community types, arguing instead that changes in communities are gradual. He found for old-field successions that the initial species composition appeared to have an important influence on the path of succession. His 'initial floristics', where most species are present at the beginning of a sere, was proposed as an alternative to the Clementsian 'relay floristics', where species are not present initially, but invade as the succession proceeds.

The 'reality' of climax formations was accepted by many ecologists, including Tansley (1935), who was nevertheless critical of some of Clements' terminology. Tansley emphasised the importance of considering not only the community but its abiotic environment with which it interacts and who coined the term 'ecosystem' to describe them together.

Lindeman (1942) stressed that the ecosystem (rather than the community) is the fundamental ecological unit because biotic components cannot be separated from abiotic ones. He was the first to show the ecosystem as a thermodynamic entity, with trophic levels through which energy and matter pass. To Lindeman, succession is ecosystem development towards a "relatively stable condition of equilibrium." (p. 409).

Lindeman's thermodynamic ecosystem was the seed of *systems ecology*, the study of behaviour and development of ecosystems. Systems ecologists study fluxes of matter and energy between ecosystem components that do not necessarily correspond to tangible entities such as individual organisms or communities. Ecosystems have been described as having *emergent* properties that cannot be predicted from knowledge of separate components, but are a result of their organisation (Odum 1971).

To systems ecologists, succession is the development of an ecosystem towards a 'mature' state, measured by such attributes as rates of biomass accumulation and mineral exchange and by structural complexity (Odum 1971). Systems ecologists have rejected the rigid determinism of Clements, but describe succession as reasonably directional—and hence predictable—with trends to be expected in ecosystem attributes as development proceeds (Odum 1969). Recent reviews of systems ecology have placed less emphasis on the holistic aspects such as ecosystem development and emergent properties (e.g., Pomeroy *et al.* 1988).

1.2.2 Successions of individuals

An alternative view of succession, often cited as being in opposition (e.g., by McIntosh 1980, 1981) to the developmental view described above is that which sees succession as the massed effect of growth and interactions of many individual organisms. The first proponent of the individualistic view was Gleason (1917, 1926, 1927), who objected to Clements' rigid determinism and saw succession as being highly variable and unpredictable.

Whittaker (1953, p. 44) described succession as "lacking orderliness or uniformity in detail though marked by fairly uniform overall tendencies." Such views of succession concentrate on the great variability to be found in vegetation when viewed at a small scale.

Many subsequent theories of succession take an explicitly or implicitly Gleasonian viewpoint. Drury and Nisbet (1973, p. 362) were explicit in stating that:

> ... a complete theory of vegetational succession should be sought at the organismic, physiological or cellular level, and not in emergent properties of populations or communities.

Connell and Slatyer (1977) concluded that an individualistic concept of succession is supported better by field data than a holistic one.

The vegetation dynamics models described in Section 1.3 are mostly based on Gleasonian principles.

1.2.3 A hierarchical synthesis?

It is clear that, in general, 'holists' and 'reductionists' are concerned with different aspects of succession that cannot be related to each other in any simple way (O'Neill *et al.* 1986). Proponents of each point of view accuse the other of ignoring important components but they are asking different questions. More recently, a hierarchical view of ecosystem structure and function has been promoted that attempts to tie these disparate views together into a coherent body of theory (Allen and Starr 1982; O'Neill *et al.* 1986).

O'Neill *et al.* (1986) referred to these two approaches as the 'population–community' and the 'process–functional' approaches to ecosystems. They warned that neither approach is fundamental nor the best way to view ecosystems and their function but are two different views of different parts of the same reality. Given this, McIntosh's (1981, p. 16) concern:

> The key distinction between species attributes and biogeochemical processes as the basis of succession requires resolution.

becomes irrelevant, because neither approach is fundamental.

Viewed hierarchically, ecosystems comprise many different processes proceeding at different rates with a range of degrees of interaction between them (Webster 1979). Ecosystems are hierarchical because higher-level processes operate at slower rates over larger areas and provide constraints to lower-level, smaller-scale processes. Differences in rates determine ecosystem structure: there are greater differences in rates between organisational levels than within them. This structure makes ecosystems inherently stable because of the asymmetry of the relations: changes at higher levels greatly affect lower levels but changes at lower levels have minimal effect on higher levels (O'Neill *et al.* 1986).

Urban *et al.* (1987) and Shugart and Urban (1988) presented a hierarchical view of forest process and function in which phenomena emerge at the scale of landscapes as a result of lower-level phenomena. Processes operating at the small scale of a forest gap are seen as the mechanisms of processes that operate at the higher level of the forest stand, which are in turn mechanisms of watershed-level processes, themselves seen as mechanisms at the landscape level. Viewing the hierarchy from the other direction, higher-level processes operating at slower rates act as constraints on lower-level ones (Shugart and Urban 1988).

Pickett *et al.* (1987a, 1987b) presented a hierarchical scheme of the causes of vegetation succession at lower levels than that of Shugart and Urban (1988). They showed a three-level hierarchy in which phenomena at the highest level such as differential species performance are explained by middle-level mechanisms. Middle-level mechanisms in their hierarchy include plant life-history characteristics and are themselves the product of lower-level mechanisms such as within-plant resource allocation.

While a hierarchical view of processes has become emphasised recently, it is not new, having been advocated by Watt (1947) in his studies of plant pattern and process and by Bormann and Likens (1979) in the Hubbard Brook Ecosystem Study. Its value is that it encourages ecologists to consider the temporal and spatial scales at which different phenomena occur and how differences in scale affect organisms and their interactions.

1.3 Models of vegetation dynamics

This section describes a number of models of vegetation dynamics that express particular theoretical positions about the nature of successional change. These models are mostly Gleasonian, non-equilibrium models. Some are conceptual models whose purpose is to explain successional phenomena from particular mechanisms, some seek to predict dynamic phenomena using particular mechanisms and some are primarily predictive (with little mechanism).

1.3.1 Interactions between species

As a logical development of a Gleasonian view of succession, ecologists have focussed their attention on the nature of interactions between plants and how they influence succession. Competition has been designated as the interaction of overriding importance in succession (e.g., Drury and Nisbet 1973) but Connell and others (Connell 1975; Connell and Slatyer 1977) have suggested that interactions with predators, herbivores and pathogens may be critical to some successions.

Connell and Slatyer (1977) described three possible outcomes of interactions between species during succession. *Facilitation* occurs when a species makes site conditions favourable for the next species in the sere, at the expense of itself (this interaction is most like Clements' reaction). *Tolerance* occurs when species tolerate each other, with no particular effect on each other. *Inhibition* occurs when a species makes conditions less suitable for succeeding species. Connell and Slatyer (1977) concluded that inhibition resulting from competition is the most common interaction between species.

There has been much discussion of the utility of the facilitation–tolerance–inhibition model (hereafter FTI) and some disagreement as to how it should be applied (Connell and Slatyer were somewhat ambiguous on this point). Huston and Smith (1987) asserted that FTI should be applied only to direct interactions between individual plants, while Walker and Chapin (1987) apply them to population interactions. Pickett *et al.* (1987b) use FTI for both individuals and populations as different levels of a hierarchy, specifying the level at which they mean the terms to apply.

The overriding value of Connell and Slatyer's (1977) contribution has been to focus debate on the importance of plant interactions in determining the nature of vegetation change.

1.3.2 Species 'strategies' and attributes

Another development from Gleasonian principles of succession is the study of patterns of plant form, life-history, physiology etc. and their association with successional behaviour. This work has either sought to describe 'strategies' adopted by plant species or to elucidate particular species attributes that are important to their behaviour.

Pickett (1976) sought to place succession theory into an evolutionary framework. He described succession as a temporal gradient of decreasing physical stress, along which there is evolutionary differentiation of species, analogous to habitat and niche differentiation along spatial gradients. Because landscapes consist of continually-changing patches of different successional environments, there is a complex pattern of selection pressures on populations.

Plants have evolved a wide variety of mechanisms for exploiting resources with different species being competitively superior under different conditions. Trade-offs in species attributes arise because it is not possible for a species to be a superior competitor under all conditions—there are no 'super-species' (Pickett 1976).

Grime (1977, 1979) sought to explain vegetation processes in general in terms of plant 'strategies' and described three primary strategies that have evolved in plants. The primary strategies are those of ruderals, competitors and stress-tolerators. Within a triangle having a primary strategy at each apex fall many secondary strategies that are combinations of the three. There are plant physiological, anatomical and life-history attributes associated with each strategy.

Grime (1977) defined these strategies on the basis of two general environmental gradients: stress (which reduces photosynthetic production) and disturbance (which reduces existing plant biomass). During the course of succession, stress increases as standing biomass uses more of the available light and soil resources. Note that Grime defines stress as autogenic, where Pickett's *decreasing* stress during succession refers to the amelioration of environmental fluctuations by standing biomass.

The major criticism of Grime's model is that it is too general and over-simplifies complex relationships between plants and their environment. For example Grime's single gradient of disturbance includes disturbances of varying frequency, to which quite different plant characteristics are adapted (Grubb 1985). Similarly, Grime's category of stress-tolerating plants includes such diverse taxa as lichens and forest trees (Grime 1977): such a category is of limited utility because they are tolerant of different types of stress. Grime also contends that competition is only important under conditions of low stress and disturbance but competition can be important under conditions of low productivity and shortly after disturbance (Grubb 1985).

I agree with Grubb (1985) that it is more productive to seek patterns of 'key characters' (Grubb's terminology) in plant species than to classify them according to 'strategies'. The ALLOCATE model of Tilman (1988) and Shugart's (1984) definition of forest tree 'roles' describe plant 'strategies' but define them precisely in terms of species attributes, either continuous or discrete in nature.

Tilman (1982, 1985, 1988) has sought to understand plant dynamics on the basis of plant strategies in acquiring resources. In his ALLOCATE model (1988) the strategy of a species is defined by its pattern of allocation of photosynthate to stems, roots and leaves.

Resources are of two general types: above-ground (light) and below-ground (water and nutrients). Allocation to leaves increases a plant's ability to absorb light; allocation to stems increases its ability to collect available light by growing taller than other plants; and allocation to roots increases its ability to collect soil resources (water and nutrients). According to Tilman's theory (1985) and his computer model (1988) it is the ratio of above-ground to below-ground resources that is important in determining which strategy (allocation pattern) will be successful.

Tilman (1988) addressed the issue of phenotypic plasticity in plants but found that its effects on predictions by his model were few. He also argued that plasticity itself has an energetic cost that reduces a plant's ability to acquire resources.

A serious criticism of ALLOCATE concerns its simplified treatment of plant mortality. All plants are subject to a constant, density-independent rate of loss. Plants that fail to acquire sufficient resources for respiratory requirements become dormant but do not suffer the increased mortality observed in suppressed plants in the field (Harcombe 1987; Peet and Christensen 1987).

Tilman (1988) viewed succession as comprising a phase of transient dynamics followed by a much longer process of long-term soil change (such as accumulation of soil nitrogen during succession on poor soils). His model is really concerned with vegetation equilibria tracking changes in the soil, with transient dynamics considered to be a temporary nuisance and an impediment to scientific progress:

> Transient population dynamics may be a major factor slowing down the rate at which ecologists gain an understanding of the workings of nature. (Tilman 1988, p. 204)

But studying transient dynamics may be more appropriate for a better understanding of vegetation dynamics, given that most natural and managed systems will undergo some disturbance before they have time to reach equilibrium (Connell and Slatyer 1977). Moore (1989a) argued that transient dynamics would become more important in the ALLOCATE model if it included a more realistic model of mortality.

The ALLOCATE model also assumes that the maximum relative growth rate of a species is correlated with the ratio of photosynthetic to non-photosynthetic tissue in plants (Tilman 1988). Shipley and Peters (1990) tested this assumption on 68 species of angiosperm seedlings and found it invalid.

Huston and Smith (1987) argued that dependence on ratios of resources (Tilman 1985) is unnecessary to explain the range of successional behaviour of plant species. They argued that one resource is limiting at any time and that, for secondary succession in forests, light is the fundamental limiting resource. Secondary succession as defined by Huston and Smith (1987) is a non-equilibrium process and is equivalent to Tilman's (1988) transient dynamics.

Shugart's description of four ecological 'roles' assumed by forest tree species

(Shugart 1984) defines those roles from the intersection of two demographic dichotomies: need for a canopy gap for establishment of a species; and production of a gap when a mature tree dies. While there are other definitions of roles that could be made, a wide variety of patterns of forest dynamics can be predicted from these four categories (Shugart 1984, 1987).

Shugart's recognition of tree 'roles' is an example of the general proposition that:

> The key to the complexity of successional patterns is not a variety of different mechanisms but rather a variety of patterns in relative characteristics of the species. (Huston *et al.* 1988, p. 688)

and is the particular example of importance to forest succession, where light is the most important resource (Huston and Smith 1987).

The vital attributes model of Noble and Slatyer (1977, 1980) was constructed with the aim of finding the smallest set of attributes necessary to model changes in vegetation composition subject to recurrent disturbance. The attributes describe: method of species arrival after, or persistence through a disturbance; the ability to establish and grow to maturity in the developing community; and the time taken to reach critical life stages. Each unit modelled is a population of functionally-identical taxa.

The vital attributes model predicts the presence or absence of populations of taxa with like attributes in a community of unspecified size, having a homogeneous environment. Disturbances affect the whole community. It is a general but imprecise model, being unable to predict how much of each vital attributes group is present.

A recent development of the vital attributes concept is the FATE (Functional Attributes in Terrestrial Ecosystems) model of Moore and Noble (1990). FATE models vegetation covering a landscape unit (*sensu* Naveh and Liebermann 1985), simulating the dynamics of functional groups of plants (Botkin 1975). It makes predictions with 'discrete quantitative' resolution (using a ranked scale such as low, medium, high) of abundance. Functional groups are sets of plants that share life-history characteristics and which behave in a similar, ecologically predictive way. Grime's (1977) model of plant strategies and the vital attributes model also use functional classifications of plants.

FATE makes predictions with a greater degree of precision than vital attributes but does so at the expense of requiring more detailed parameters to describe the functional groups. The use of discrete scales of measurement make parameterising the model simpler than if precise numerical values are required (Moore and Noble 1990).

FATE was also designed to be mathematically tractable so that its behaviour can be analysed under non-equilibrium conditions. It can be used to generate replacement sequences for a particular stand and calculate extinction probabilities and mean times to extinction of functional groups (Moore and Noble 1990).

1.3.3 Change-of-state models

These models simulate forest dynamics as sequences of changes from one state to another. Change-of-state models can be deterministic, with rates of transition expressed as differential equations, or stochastic and use a matrix of probabilities of transitions from state to state (Markov models).

An example differential-equation model is that of Shugart *et al.* (1973) for large areas of the Great Lakes region of North America. States were defined from fifteen predefined forest types and three age classes within each type. Shugart *et al.* (1973) estimated parameters for the 45 differential equations of annual transitions and predicted changes in area of the forest types over 250 years in the absence of disturbance.

Markov processes have been explored by a number of authors as models of succession (e.g., Stephens and Waggoner 1970; Horn 1974, 1975, 1981; Usher 1979, 1981; van Hulst 1979). They have the property of convergence to the same stationary set of states irrespective of starting point (Horn 1975) which makes them obvious analogues of convergence in succession. Markov models can also be constructed to mimic other patterns of change found in succession (van Hulst 1980; Horn 1981).

State definitions vary between Markov models. For example, Horn (1974, 1975, 1981) modelled succession as a tree-by-tree replacement process with each state being a vector of probabilities of occupation by species but Stephens and Waggoner (1970) used a vector of species composition of forest stands.

Markov models are based on the assumption that the future state of the system can be predicted solely on the basis of the current state—there is no 'memory' of past states. Such an assumption is unlikely to hold in forests, where the soil is an important repository of information about past forest states (van Hulst 1980). A Markov model can be extended to include some 'memory' information by defining additional states but such an expansion necessitates estimation of more transition probabilities: for n states there are n^2 probabilities. Van Hulst (1980) observed that the addition of information about soil states to a Markov model turns it into a model of ecosystem dynamics rather than only vegetation dynamics.

Markov models are also based on the assumption that transition probabilities remain constant with time—a property called stationarity (Horn 1975, 1981). Horn (1975) found this assumption justified for his model of forests near Princeton. Periodic small-scale disturbance will alter transition probabilities. Horn (1981) suggested the use of additional 'disturbance' matrices with probabilities in disturbance years but these, like additional states to include system 'memory', increase the numbers of parameters to be estimated for the models.

Moore (1990) has shown that the semi-Markov process can be used for modelling vegetation dynamics and that it overcomes the problems of Markov processes. The semi-Markov process is an extension of the Markov process where it can remain in one state for a number of time-steps; after a certain number it will change to another state (succession) but if a disturbance occurs it will change to a different state. The probability that a disturbance will occur can vary with the current state and the time since the vegetation entered that state.

Moore (1990) showed that the required probability information can be provided by replacement-sequence information such as that provided by the vital attributes (Noble and Slatyer 1980) and FATE models (Moore and Noble 1990).

Kessell and Potter (1980) constructed a succession model for forests in Montana that has converged to a similar structure to Moore's (1990) semi-Markov model, although coming from a different direction (i.e., a specific modelling need rather than an application of a mathematical construct to a general modelling problem). The state definitions in Kessell and Potter's model include quantities of component species on a seven-point importance scale and different transitions occur as a result of different intensities of fire as well as succession (no disturbance after a specified period of time).

1.3.4 Individual-tree models

Models with individual trees as their fundamental unit (often called 'tree models') are explicitly Gleasonian in their formulation: they simulate responses of individuals to their local environment that includes the effect of neighbouring individuals. Information about extent and distribution of variation in populations is modelled explicitly in such models, rather than being subsumed into single measures, as in state-variable models (Huston and Smith 1987; Huston *et al.* 1988). Tree models make predictions about variation in size and age structure as well as individual tree attributes within and between forest stands (Dale *et al.* 1985).

Individual-based models also avoid problems with aggregation error (Gardner *et al.* 1982; Loehle 1987)—loss of modelling accuracy through inappropriate aggregation of real-world objects into model parameters.

Shugart and West (1980) and Shugart (1984) provide surveys of tree models used in forestry and ecology. The most detailed tree models maintain information of horizontal tree locations and use it to calculate competitive interactions between trees. Examples are FOREST of Ek and Monserud (1974) and RESCOMP of Penridge *et al.* (1987).

1.3.5 Forest gap models

A special case of individual-tree models are forest gap models. Growth, establishment and mortality of individual trees are simulated on a small plot, approximately the size of a gap left by the loss of a mature tree (Shugart and West 1979). Instead of horizontal spatial structure, gap models contain explicit vertical structure determined by tree heights that determine the competitive status of trees (Shugart 1984).

The underlying philosophy of gap models is to simulate dynamic forest processes using general equations derived from basic knowledge of plant physiology, morphology and life history (Shugart 1984). Parameter values for the equations can be obtained from widely-available forestry texts, or calculated from field data when available. Ideally, species parameters are those that describe the whole species:

> However, the underlying concepts of the simulation are general. The properties of each species are derived from its entire geographic range and in theory any non-hydrophytic species whose relevant characteristics are known can be entered into the simulation. (Botkin *et al.* 1972, p. 849)

The first gap model was JABOWA, developed by Botkin *et al.* (1972) for forests in the Hubbard Brook ecosystem (Bormann and Likens 1979). Most subsequent implementations have been adapted from FORET, constructed by Shugart and West (1977) for forests in eastern Tennessee. Shugart (1984) provides a thorough review of gap model implementations to that date. Since then, implementations have included those by Dale *et al.* (1986), Pastor and Post (1986), Waldrop *et al.* (1986), Busing and Clebsch (1987), Leemans and Prentice (1987) and Smith and Urban (1988).

That the gap model framework has been applied successfully for two decades is an indication of the robustness (and hence generality) of the approach. In the final chapter of his book, Shugart (1984) argues that gap models have been successful because they replicate the most important processes operating in forest successional dynamics and that these models constitute a theory of forest dynamics. Huston *et al.* (1988) propose that simulation models like gap models will play a valuable part in exploration of theoretical issues in ecology—a position traditionally held by analytical models (Horn *et al.* 1989).

Even though gap models are Gleasonian in their formulation they have always had links with systems ecology: the first gap model (JABOWA) was developed as part of the larger study of Hubbard Brook ecosystem; and Shugart and O'Neill (1979) reproduced the description of JABOWA by Botkin *et al.* (1972) in a volume describing significant advances in that field. Some gap models directly simulate exchange of matter between biotic and abiotic components of forests ecosystems.

For example, Pastor and Post's (1986) gap model implementation simulates cycling of carbon, soil nutrients and water between the biotic and abiotic components of the forest. They used the model to make accurate predictions of primary production, biomass accumulation, nitrogen availability and soil organic matter.

A number of features of other models of vegetation dynamics that I described above are evident in gap models. Interactions between species are modelled explicitly, with the exact nature of interactions being determined largely by differences in species characteristics (Huston and Smith 1987). Species 'strategies' can be recognised but they are defined on the basis of physiological and life-history characteristics, rather than being defined *a priori*.

Gap models have been described as Markovian in character by Horn (1975, 1981), implying that the description of the simulated plot at any time is sufficient to determine its future states. I think the connection is a tenuous one because the plot description includes 'memory' such as time since disturbance and sizes of trees, and there is an infinite variety of 'transition matrices' to apply according to extrinsic variables such as temperature and rainfall. A more recent connection with Markov models is the use of a gap model by Horn *et al.* (1989) to generate a transition matrix for a simpler Markov model.

Moore and Noble (1990) criticised gap models as being too detailed for accurate determination of truly general parameter values. As a result, while the model structure is quite general, each implementation requires some 'tuning' of parameters to make accurate predictions.

1.3.6 BRIND

BRIND is a gap model implementation for the eucalypt forests of the Brindabella Range. Shugart and Noble (1981) restricted its application to forests occurring on moist, sheltered slopes at higher altitudes (the Brindabella Range is described in Section 1.6). It was constructed to investigate the applicability of a North American modelling framework to eucalypt forests. In Chapter 2 I describe the structure of BRIND in detail and discuss tests of it by Shugart and Noble (1981).

BRIND is notable as a gap model for two reasons. Firstly, in gap models light is the most important resource for which trees compete (Shugart 1984; Huston and Smith 1987) but the eucalypt forests simulated by BRIND are not all light-limited (see Section 1.5). I will argue in Chapter 2 that BRIND's model of competition for light works as a general one-sided competition in open forest stands.

Secondly, eucalypt forests (including those in the Brindabella Range) do not in general show classical patterns of successional species replacement after disturbance; changes in composition are often associated with changes in fire regime (Noble and Slatyer 1981; see Section 1.5). Thus BRIND shows that the gap model formulation can predict effects of variation in disturbance regime.

1.4 Response to environmental gradients

BRIND simulates forest dynamics over a range of altitudes. Its structure is based on a set of assumptions about the nature of species' responses to a gradient of temperature. This section discusses issues relevant to the modelling of response to environmental gradients.

Historically there has been disagreement about whether observed assemblages of species have some special existence as communities or whether they occur together because they respond in similar ways to environmental gradients. This community–continuum debate has its origin in the Clements–Gleason dichotomy of ideas about succession: to Clements it was a sere of community types; to Gleason it was the outcome of the responses of individual plants to their environment.

However an individualistic (Gleasonian) view of vegetation can include coincident species optima and boundaries (Goodall 1963).

Austin (1985, 1986) states that by general consensus the continuum concept has become the prevailing paradigm in the analysis of vegetation patterns.

1.4.1 Environmental ordination

Exploration of plant species responses to environmental gradients includes vegetational ordination and environmental ordination (Austin 1985). In vegetational ordination, patterns of vegetation composition are examined in an attempt to reveal underlying environmental gradients. In environmental ordination, species or community distributions are examined in relation to selected environmental gradients.

The technique of Direct Gradient Analysis (DGA) was developed by Whittaker (1967, 1978) to explore patterns of species response to environmental gradients. He showed that complex patterns of species responses can be reproduced by combinations of simple response shapes. This is a spatial analogue of the argument (e.g., by Huston *et al.* 1988) that successional patterns can be predicted from combinations of simple species attributes.

DGA is a graphical technique where species distributions are displayed on environmental axes of topography–moisture and altitude. Mosaic diagrams show distributions of forest types against these two axes. The data can also be used to construct nomograms that display frequency distributions of individual species and community measures such as diversity and productivity (Whittaker 1978).

Kessell (1979) developed DGA as the basis of a land management system for extensive forested areas. He found that additional gradients were necessary to predict vegetation found in the Glacier National Park in Montana. The added gradients are: time since fire, drainage, primary succession and alpine wind–snow. Successional changes are predicted using the vital attributes model of Noble and Slatyer (1977, 1980). The gradient modelling system also includes models that predict build-up of fuel and behaviour of fires.

Two conclusions that have been drawn from DGA work are that species' response curves are generally gaussian, or bell-shaped, and that modes of major species are uniformly distributed along gradients while modes of minor species are randomly distributed (Gauch and Whittaker 1972). These assumptions have been questioned by Austin and co-workers, who have subsequently found skewed and occasional bimodal response curves without regularity of distribution (Austin 1985; Austin *et al.* 1983, 1984, 1990). The formulation of many gap models including BRIND (see Chapter 2) is based on the assumption that tree growth shows a symmetrical (parabolic, effectively gaussian) response to temperature.

1.4.2 Niche and habitat

Some ecologists have attempted to describe plant species distributions in terms of niche theory, but there has been disagreement about how the *niche* of a species is defined. In an effort to reduce confusion, Whittaker *et al.* (1973) suggested restricting the word 'niche' to Elton's (1927) functional definition (the 'occupation' of a species) and using the word *habitat* to refer to its position in the broader environment (its 'address'). The advice of Whittaker *et al.* appears to have gone largely unheeded and the word 'niche' is used in both senses by various authors (Schoener 1989).

In plant ecology, the word 'niche' is often used to refer to the location of a species on a number of environmental gradients (for example, Austin *et al.* 1990)—the 'habitat' of Whittaker *et al.* (1973). I think this confusion stems from difficulties in translating a definition (the Eltonian niche) from animal ecology into plant ecology with gradients of different types: "A gradient of nitrate-nitrogen supply cannot be equated with a food-particle size gradient" (Austin 1985, p. 55). The Eltonian niche describes how a species functions within a community, but the approach of plant ecologists like Austin *et al.* (1990) has been to examine species distributions without assuming any *a priori* existence of communities. Silvertown and Law (1987) questioned the necessity of defining niches for plant species because large numbers of them can co-exist with apparently identical niches (in apparent contravention of niche theory, which states that no two species can occupy the same niche).

1.4.3 Physiological and ecological response

Hutchinson (1958) made an important contribution by separating the *fundamental niche* of a species from its *realised niche*. The fundamental niche is that which a species would occupy in the absence of other organisms, while their presence restricts it within that to its realised niche. The important concept is the separation of the potential distribution of an organism from its actual distribution.

Hutchinson's idea translates into plant ecology as the difference between the physiological and ecological responses of plant species to environmental gradients. The physiological response of a plant species is that defined by its (genetically determined) physiological characteristics, in the complete absence of any other organisms. Its ecological response is the restriction of its physiological potential by the presence of other organisms such as herbivores and competitors. I will argue in Chapter 2 that the realism of BRIND suffers from a failure to distinguish between these two responses.

Plants are assumed to have very similar physiological responses (fundamental niches) because they have essentially the same requirements for light, water and nutrients (Landsberg 1986). It is further assumed that they respond differently to competition. Ellenberg (1953, 1954) found evidence of shifts in species optima between plants grown in monoculture and mixture in glasshouse experiments. Austin (1982) found that ecological optima of species grown in mixtures in the glasshouse could be predicted from an index of their relative physiological performance in monoculture, but found later (Austin *et al.* 1985) that root : shoot ratio and level of resource were important as well.

1.5 Eucalypt forest dynamics

In this section I discuss general features of eucalypt forest dynamics relevant to modelling by BRIND.

Members of the genus *Eucalyptus* have a number of typical characteristics that enhance their survival in conditions of recurrent fire and drought. It has been argued that these characteristics, such as sclerophylly and development of a lignotuber, evolved primarily as a response to low nutrients before droughts and fires became prevalent, but now serve to enhance their survival in a wide range of habitats (see Florence 1981 for a review).

Eucalypt species show great efficiency in their use of nutrients, taking them up when they are available and storing them in lignotubers and sapwood. Trees can recycle nutrients internally by withdrawing them from leaves before leaf-shed and from xylem before heartwood formation (Florence 1981; Banks 1982).

Seedlings of most eucalypt species form lignotubers, underground woody organs that contain bud strands and store nutrients (Jacobs 1955). Lignotuberous seedlings will grow slowly above ground while developing a significant root system.

They can survive repeated decapitation and fires, but when released from suppression by taller trees they shoot vigorously and grow rapidly into a sapling. The rapidly-developing sapling can maintain fast growth because of its well-developed root system and nutrient state (Florence 1981).

Eucalypt species tolerate water stress to varying degrees. Species characteristics that aid in drought tolerance include vertical leaf orientation and sclerophylly. There is evidence that some dry-region eucalypt species are very efficient in their use of water by maintaining a low leaf area index and high net assimilation rate (Florence 1981).

Some eucalypt species have limited tolerance to drought and show high rates of water use. Some species of the 'ash' group, notably *E. regnans*, are restricted in range to sites with plentiful water and shade, and with deep, well-drained soils (Florence 1982). In mainland forests the taxonomically-similar *E. delegatensis* (that occurs in the Brindabella Range) is restricted to these more sheltered sites, but in Tasmania it is found on sites with a much wider range of rainfall (700–1400 mm per year; Bowman and Kirkpatrick 1984). Trees of these drought-intolerant species will survive periods of low rainfall, but their growth can be adversely affected. For example, Amos *et al.* (1950) found regions of anomalous wood growth in small individuals of *E. delegatensis* that had been subject to an extended dry period.

Most communities dominated by eucalypt species on poorer sites have open canopies with low leaf area index and are structurally open forests or woodlands (Specht 1981a). The dominant trees in these communities are relatively widely-spaced and have sparse crowns but fully exploit the available soil resources and suppress growth of understorey plants and developing seedlings in between (Florence 1981). It is apparent that trees in these communities are not competing primarily for light, but for soil moisture and nutrients. By comparison, sheltered sites with sufficient soil support tall forests with dense canopies where competition for light is more important.

Recurrent fire is a common feature of eucalypt forests, and many eucalypt species have characteristics that indicate adaptation to a particular regime of fire intensity and frequency (Gill 1975; Noble and Slatyer 1981). Trees of most eucalypt species can survive complete canopy scorch and show recovery immediately afterwards from epicormic buds in the bark (e.g., Gill 1978). Extent of recovery varies between species and is dependent on post-fire climatic conditions (Cochrane 1966). After fire, trees can show sustained accelerated growth, perhaps as a result of release of competition and availability of nutrients (Florence 1981; Banks 1982). Lignotuberous seedlings usually survive burning of their above-ground shoots and replace them with new shoots from buds in the lignotuber (Noble 1984).

Some eucalypt species are sensitive to fire; trees of them are killed by full canopy scorch (Gill 1981a; O'Dowd and Gill 1984). After an intense fire in which all trees are killed, these species rely on post-fire germination and establishment of seedlings to maintain presence at a site. If another fire occurs before the new generation of trees has matured, the species may become locally extinct. Stands of these species initiated by fire events are generally even-aged but mixed-age stands also occur because of patchiness of fires and because seedlings can germinate in the absence of fire (Bowman and Kirkpatrick 1986; Lindenmayer *et al.*, in press).

Insects are the principal herbivores affecting eucalypts. Nutrients that would otherwise cycle very slowly through decomposition are cycled quickly through the insects' frass. There is evidence that differential herbivory of eucalypt species may help regulate species mixtures (Chilvers and Brittain 1972; Morrow 1977). Morrow and LaMarche (1978) found that experimental removal of insect grazing from *E. pauciflora* resulted in greatly increased radial growth rates in those trees.

This result suggests that chronic insect herbivory may reduce the productivity of many eucalypt forests.

Classical successional behaviour (vegetation change resulting in an eventual 'climax' formation) appears not to be found in many eucalypt forests. This is because two important components of succession—competition for light and long periods between disturbance relative to species life-spans—are uncommon. Since the publication of reviews like those of Noble and Slatyer (1981) and Florence (1981) there has been less emphasis on identification of seres and climax formations in Australian forests and more emphasis on description of important life-history characteristics and species' responses to disturbance.

The 'inhibition' and 'tolerance' pathways of Connell and Slatyer (1977) appear to be the most common interactions between plants in eucalypt forests (Florence 1981). Purdie and Slatyer (1976) studied post-fire succession in eucalypt woodland near the Brindabella Range and found that species patterns conformed to the 'initial floristics' model of Egler (1954). They found that nearly all species in the tree and shrub strata recovered quickly after the fire and that no new species invaded the study area.

Competition for light is considered the motivating force in secondary succession (Huston and Smith 1987) but canopy development of many eucalypt forests appears to be limited by soil resources. Where rainfall does not limit canopy development, light-driven succession can take place (e.g., in Tasmania; Jackson 1968).

1.6 The Brindabella Range

The Brindabella Range marks the western boundary of the Australian Capital Territory and covers altitudes from *c.* 600 m to more than 1800 m. The forests simulated by BRIND occur on the eastern side of the range, in the catchment of the Cotter River. The Cotter River catchment is described in detail in *A Resource and Management Survey of the Cotter River Catchment* (ANU Forestry 1973). In this section I discuss aspects of the geography, ecology and history of the catchment relevant to my study of BRIND.

The climate of the area is characterised by wide temperature variations, with summer maximum temperatures commonly above 30°C and winter minimum temperatures commonly below −4°C at all altitudes (screen temperatures). Both temperature and precipitation (rain and snow) vary with altitude: temperature decreases and precipitation increases with it. Precipitation does not vary greatly from month to month and is slightly higher at most stations from August to October. There have been months without rain during summer recorded at most stations in the Brindabella Range (temperature and rainfall data from CSIRO Division of Water Resources Research and the Bureau of Meteorology). Rainfall varies from year to year and severe droughts have been recorded in the region (ANU Forestry 1973). Snow falls regularly in winter at altitudes above 1000 m and stays on the ground longer at higher altitudes. Frosts occur regularly, can kill seedlings (Harwood 1980; Williams 1989) and may cause cessation of cambial growth (Banks 1982).

Soils of Cotter catchment are derived from three broad categories of parent material. The oldest materials are Ordovician sediments throughout most of the catchment area. These are overlain with Silurian to Devonian granites in some areas and Silurian volcanics in others.

The major soil groups in the Cotter catchment are red and yellow podzolics, red forest loams and alpine humus soils. Distribution of soils is correlated with geology, topography and climate. There is evidence that overall fertility increases slightly with altitude, although no soils are very fertile. The most productive soils are the red forest loams on sheltered slopes because of their depth, friability and water-holding capacity, and the least productive ones are the shallow podzolics on exposed aspects (ANU Forestry 1973).

There are a number of interrelated gradients in the Brindabella Range. Rainfall and soil nutrients increase with altitude (favouring growth of trees) but temperature decreases (limiting growth). Soil depth varies with aspect, as does moisture availability (because of differing evaporative heat loads; ANU Forestry 1973).

The most important human disturbance in the Brindabella Range is fire. It is believed that fires before European settlement were relatively uncommon. After settlement the frequency and intensity of fires increased greatly (ANU Forestry 1973). From analysis of fire scars, Banks (1982) showed that the average period between fires at one site decreased from 10–70 years before 1850 to 3–4 years (apparently from regular burning-off) during the century up to 1950.

It is believed that fires which occurred before settlement were also of lower intensity than those occurring afterwards. The evidence for this is the absence of even-aged stands originating before 1850 and from the large size and open-grown habit of pre-settlement dominant trees (ANU Forestry 1973). Major fires occurred in the Brindabella Range during the summers of 1876–7, 1899–1900, 1919–20, 1925–6 and 1938–9, and there have been numerous smaller ones. The current structure of the forests reflects this history of fire in its size structure.

Other human disturbance has been relatively minor. The higher altitude *E. pauciflora* woodlands were used for summer grazing regularly until 1911 and then less so until grazing was completely phased out by 1955. Selective logging of some stands of *E. delegatensis* and *E. fastigata* has occurred, but had ceased by 1962. These stands have regenerated well and now have a mixed-age structure (ANU Forestry 1973).

Forests of the Cotter catchment show patterns of tree species composition that vary with altitude, aspect, landform and soil type. For example, *E. pauciflora* woodland is found at the highest altitudes, in mixture with *E. dalrympleana* slightly lower on sheltered aspects. Lower on many sheltered slopes are found highly-productive tall open-forests of *E. delegatensis*, sometimes with *E. dalrympleana* and *E. pauciflora.* Further down are found *E. fastigata* and *E. viminalis*, with *E. radiata* below that. On more exposed slopes *E. pauciflora* typically occurs further downslope than on sheltered ones. Below that is commonly found *E. dives* with *E. rubida* and *E. dalrympleana*; *E. dives* continues further downslope with either *E. mannifera* or *E. rossii*, the latter on the driest slopes (ANU Forestry 1973).

1.7 This study

This chapter has provided background information to my study of BRIND. BRIND is a complex model with some hierarchical structure whose component submodels vary in their mechanistic content (Section 1.1.3). It is an essentially Gleasonian model of forest dynamic processes and as a non-equilibrium model is an appropriate for predicting dynamics of forests subject to recurrent disturbance.

I will argue in Chapter 2 that some important components of BRIND's structure are lacking in realism and that, while BRIND can make qualitatively correct predictions (Shugart and Noble 1981), those predictions are likely to be wrong in detail because they are based on some unrealistic mechanisms. In Chapter 3 I show some detailed errors of prediction by BRIND. The severity of these errors can be lessened by modifications to species parameters and program constants, but such modifications do not address the unrealistic elements in the model's structure.

Chapters 4 to 7 describe the development of alternative submodels for BRIND that better represent the mechanisms of growth, establishment, mortality and fire response in eucalypt forests. The submodels of growth (Chapter 4) and establishment (Chapter 5) also attempt to expand the applicability of the model by incorporating response to the environmental gradient of soil moisture as well as that of temperature. In Chapter 8 I combine the new components into a complete model and examine its predictions to assess the modifications.

Chapter 2

Structure of BRIND

In this chapter I examine the structure of BRIND in detail, relating it to ecological theory and published data, and describe how construction of the model is expected to affect its behaviour. Each of the major forest processes as simulated by BRIND is discussed in turn.

2.1 Introduction

BRIND requires three sets of information to carry out a simulation. Firstly, it needs parameters describing the species found in the Brindabella Range. The parameters are listed in Table 2.1, and values for the species simulated by BRIND are given by Shugart and Noble (1981, their Table 2). Secondly, initial site conditions from which to simulate are required that comprise a list of the numbers and sizes (as DBH) of trees of each of the species. BRIND maintains a list of the trees on the simulated plot: their species, age and DBH, and information about their state of suppression and fire damage. Trees are added to the list (establishment), and removed from it (mortality) during the course of simulations.

Table 2.1: Parameters used for each species in BRIND. The model processes where the parameters are used are given, together with a brief description of their meaning. 'Flags' are boolean parameters set to TRUE or FALSE

Name	Process	Description
G	Growth	Potential growth rate, derived from $D_{\max}$, $H_{\max}$ and $\mathrm{Age}_{\max}$ parameters
$D_{\max}$	Growth	Maximum attainable DBH
$H_{\max}$	Growth	Maximum attainable height
$\mathrm{Age}_{\max}$	Growth, Mortality	Maximum attainable age
$\mathrm{DEGD}_{\min}$	Growth, Establishment	Minimum tolerated DEGD (degree-days, see text)
$\mathrm{DEGD}_{\max}$	Growth, Establishment	Maximum tolerated DEGD
FRSTMAX	Establishment	Maximum number of frosts tolerated by seedlings in a year
SDSIZE	Establishment	Minimum DBH of trees for seed production (size at maturity)
SDLIFE	Establishment	Life span of viable seeds
SWITCH(1)	Establishment	Flag: TRUE if species needs fire for germination
SWITCH(2)	Establishment	Flag: TRUE if species is well-dispersed
SWITCH(3)	Establishment	Flag: TRUE if species shows enhanced germination after fire
SPRTND	Sprouting	Flag: TRUE if species can sprout
SPRTMN	Sprouting	Minimum DBH for tree to sprout
FORM	Fire	Ratio of bole height to total tree height
FA, FB, FC, FD, FE	Fire	Parameters for calculating mortality from fire (see Section 2.5)

Lastly, BRIND requires values for parameters that describe the site. Site parameters are divided into two categories: those used to describe climate, which affects establishment and growth of trees; and those used for description of fire regime. Climate is described by the measure of temperature called degree-days (DEGD, Section 2.2.1) that reflects altitude of the sites. Mean and standard deviation of DEGD for each site are supplied to BRIND, defining a normal distribution from which annual values are chosen randomly. Another site parameter related to altitude is the number of frosts occurring each year (fixed from year-to-year in BRIND, Section 2.3).

Fire regime is described by annual probability of wildfire and fire danger parameters that represent the slope of the site and the weather conditions on the day of a fire (Section 2.5).

Stochastic behaviour in BRIND is a result of random choice of annual DEGD, random occurrence of fire and randomness in modelling of tree establishment and mortality. The computer program includes a subroutine that provides pseudo-random numbers from a uniform distribution from zero to one. Pseudo-random numbers are generated in sequences that are indistinguishable from sequences of true random numbers but which can be repeated exactly by the provision of the same 'seed' number to start the pseudo-random number generator. Repeatability is useful in simulation work where a stochastic model (such as BRIND) can be run a number of times with differences in parameters or starting conditions but identical sequences of pseudo-random numbers, or with some sequences of pseudo-random numbers identical, and others different.

Choice of uniformly random events such as fire is made by generating a pseudo-random number between zero and one and comparing its value with the probability assigned to that event's occurrence. If the pseudo-random number is less than the probability, the event is simulated (e.g., calling the FIRE subroutine) is called. For choice of DEGD from a normal distribution, uniform pseudo-random numbers are transformed to a normal distribution defined by the user-specified mean and standard deviation of DEGD.

The annual sequence of computations is outlined in Fig. 2.1 and starts with the calculation of DEGD for the year (subroutine CLIMATE). If the year is a fire year (chosen randomly as described above) the FIRE subroutine is called. In this routine, intensity of the fire is calculated and the response of each tree to a fire of that intensity simulated (Section 2.5). The KILL subroutine removes trees marked as being killed by fire, and calculates mortality of trees from suppression and random factors (Section 2.4). The SPROUT subroutine simulates vegetative resprouting of trees (Section 2.6) and the BIRTH subroutine adds new trees to the plot according to site conditions and fire occurrence (Section 2.3). The GROW subroutine calculates DBH increments of all the trees, including the effects of temperature and competition, and marks slow-growing trees as being suppressed (Section 2.2).

2.2 Growth and competition

Calculation of annual growth increments (subroutine GROW) is central to all gap models, and is essentially similar in all of them. Growth is the only process in BRIND that is completely deterministic: all others have stochastic components.

2.2.1 The growth equation

In this section I describe the components of the central growth equation in BRIND and their derivation. They are discussed in more detail in the following sections.

The growth equation in BRIND can be summarised as follows: for each tree a maximum DBH increment is calculated that represents the annual increment

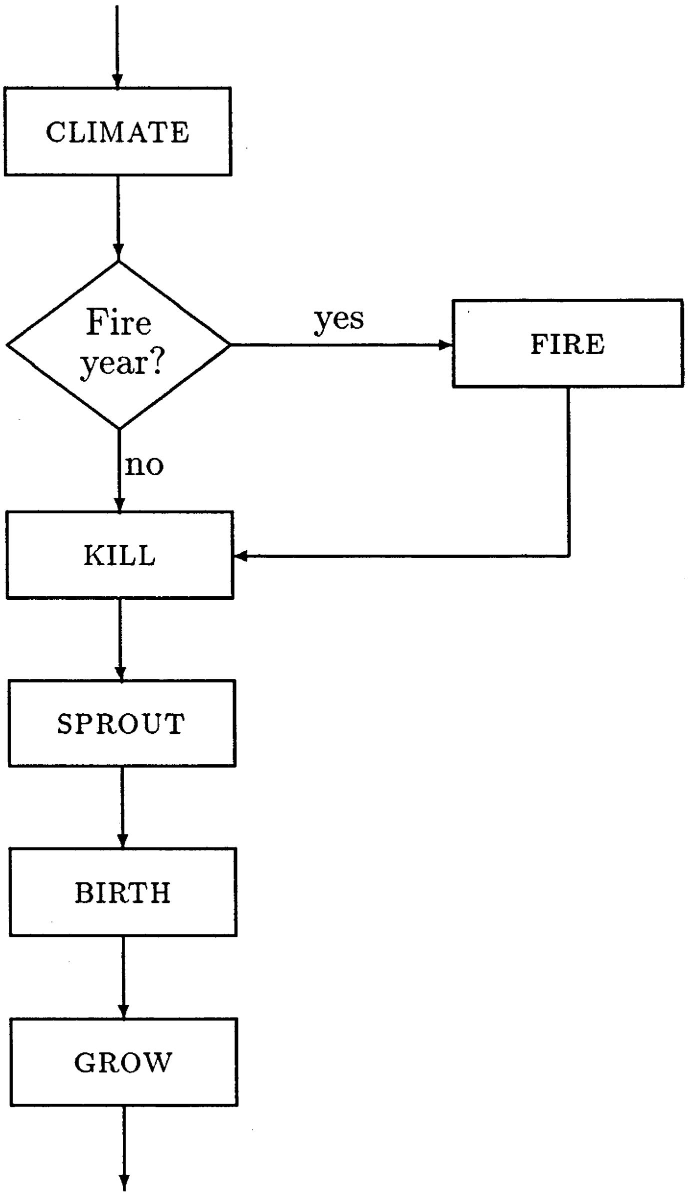

Figure 2.1: Flowchart of subroutines in the annual loop of BRIND. Other subroutines in the program are for input, output, program control, and 'bookkeeping'.

of a tree of that species under optimal conditions for growth. This maximum increment is reduced by indices that represent sub-optimal conditions of temperature, direct competition from other trees and below-ground resource availability. The indices are constrained to have values from zero to one (inclusive) and are multiplied with the maximum increment to calculate the actual increment for the tree in that year. Mulitplicative equations of this type are often used in modelling forest tree growth (Dale *et al.* 1985) and have been used for modelling plant growth more generally (e.g., Fitzpatrick and Nix 1970, Specht 1981b).

The equation is:

$$\Delta D = \underset{(\mathrm{i})}{G} \quad \underset{(\mathrm{ii})}{\frac{D\,(1 - DH/D_{\max}H_{\max})}{274 + 3b_2D - 4b_3D^2}} \quad \underset{(\mathrm{iii})}{\mathrm{T(DEGD)}} \quad \underset{(\mathrm{iv})}{\mathrm{r(AL)}} \quad \underset{(\mathrm{v})}{\mathrm{S(BAR)}} \tag{2.1}$$

where D is DBH and H is height (both in cm). The roman numerals in parentheses underneath are used to refer to parts of the equation in the following discussion:

(i) G is a parameter that describes the intrinsic growth rate of trees of a species (Table 2.1). Values are calculated from species parameters of maximum DBH, height and age, using a formula given by Botkin *et al.* (1972), shown in Table 2.2 (Eqn 2.6). This formula is based on the assumption that an optimally-growing tree will achieve two-thirds of its maximum DBH in half of its maximum age. The growth parameter G and its calculation are discussed in more detail in Section 2.2.2 below.

(ii) This expression gives the maximum growth increments for a tree growing under optimal conditions. It produces a curve that is sigmoidal with fastest growth in trees of small–medium size and increments decreasing in size as the tree approaches its maximum DBH (Fig. 2.2). It is derived

Table 2.2: Growth equations used in BRIND. Roman numerals in parentheses correspond to those in the text.

(i), (ii) Growth increment under optimal conditions:

$$\Delta V = R \, \mathrm{LA} \left(1 - \frac{DH}{D_{\max} H_{\max}} \right) \tag{2.2}$$

where V is wood volume, LA is leaf area, R is a constant, D is DBH (cm) and H is height (cm). Assuming that $\mathrm{LA} = C\ D^2$, and given

$$V = D^2 H \tag{2.3}$$

where

$$H = 137 + b_2 D - b_3 D^2 \tag{2.4}$$

for $b_2 = 2(H_{\max} - 137)/D_{\max}$ and $b_3 = (H_{\max} - 137)/D^2_{\max}$, and differentiating V with respect to D, we get:

$$\Delta D = G \, \frac{D\,(1 - DH/D_{\max} H_{\max})}{274 + 3b_2 D - 4b_3 D^2} \tag{2.5}$$

where $G = R\ C$. If we assume that at $1/2\mathrm{Age}_{\max}$, $D = 2/3 D_{\max}$, G can be calculated using:

$$G = \frac{4H_{\max}}{\mathrm{Age}_{\max}} \left\{ \ln\left[2(2D_{\max} - 1)\right] + \frac{a}{2} \ln \left(\frac{9/4 + a/2}{4D^2_{\max} + 2aD_{\max} - a} \right) \right. \\ \left. - \frac{a + a^2/2}{\sqrt{a^2 + 4a}} \ln \left[\frac{\left(3 + a - \sqrt{a^2 + 4a}\right)\left(4D_{\max} + a + \sqrt{a^2 + 4a}\right)}{\left(3 + a + \sqrt{a^2 + 4a}\right)\left(4D_{\max} + a - \sqrt{a^2 + 4a}\right)} \right] \right\} \tag{2.6}$$

where $a = 1 - 137/H_{\max}$.

(iii) Index of temperature effects on growth (each species, each year:

$$\mathrm{T(DEGD)} = \max \left[0, \frac{4(\mathrm{DEGD} - \mathrm{DEGD}_{\min})(\mathrm{DEGD}_{\max} - \mathrm{DEGD})}{(\mathrm{DEGD}_{\max} - \mathrm{DEGD}_{\min})^2} \right]. \tag{2.7}$$

Table 2.2: (continued)

(iv) Index of shading (each tree, each year):

$$r(\mathrm{AL}) = 1 - e^{-4.64(\mathrm{AL}-0.05)} \tag{2.8}$$

from available light,

$$\mathrm{AL} = e^{-k\,\mathrm{SLAR}} \tag{2.9}$$

where k is the light extinction coefficient (0.5), and SLAR is the summed leaf area index above the tree. Leaf area index is calculated as

$$\mathrm{LA} = 6.471 \times 10^{-6} D^{2.65} \tag{2.10}$$

as each tree's contribution, scaled by plot size.

(v) Index of below-ground resources (whole plot, each year):

$$\mathrm{S(BAR)} = 1 - \frac{\mathrm{BAR}}{\mathrm{SOILQ}} \tag{2.11}$$

where BAR is the total plot biomass (t/ha) and SOILQ is the maximum biomass (1000 t/ha). Biomass for each tree is calculated as

$$B = 8.546 \times 10^{-4} D^{2.58} \tag{2.12}$$

in t/ha, adjusted for plot size.

---

from an equation of wood volume growth (Table 2.2, Eqn 2.2) where the annual increment in volume is proportional to the leaf area of the tree, reduced by a factor to account for the accumulation of standing biomass $(1 - DH/D_{\max}H_{\max})$. The volume equation is differentiated with respect to DBH to give the difference equation for ΔD, assuming that leaf area is proportional to the square of DBH and also the DBH–height relationship shown in Table 2.2 (Eqn 2.4). The shape of the growth curve predicted by the difference equation (Fig. 2.2) is similar to that found by Banks (1982) for open-grown trees of *Eucalyptus pauciflora* (Fig. 2.3). Calculation of

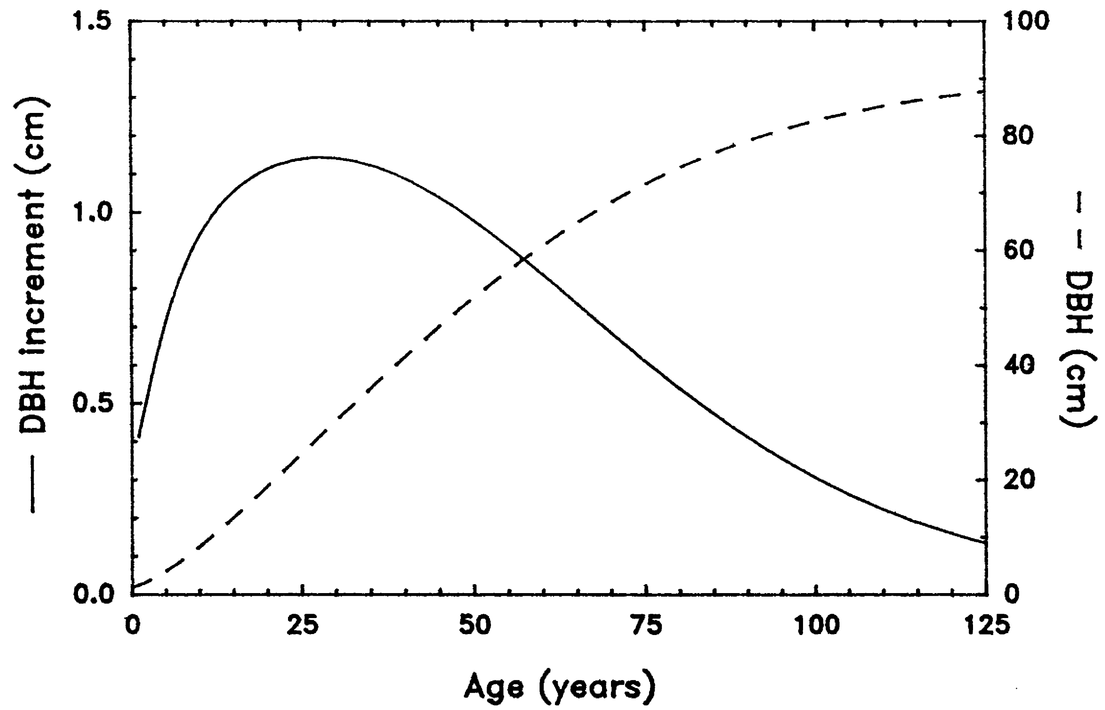

Figure 2.2: DBH increments and DBH of optimally-growing *Eucalyptus pauciflora* as predicted by BRIND, plotted against age of the tree. These curves were calculated using Eqn 2.5 using values of G, $D_{\max}$ and $H_{\max}$ for *E. pauciflora* given by Shugart and Noble (1981) and are typical of those produced by BRIND.

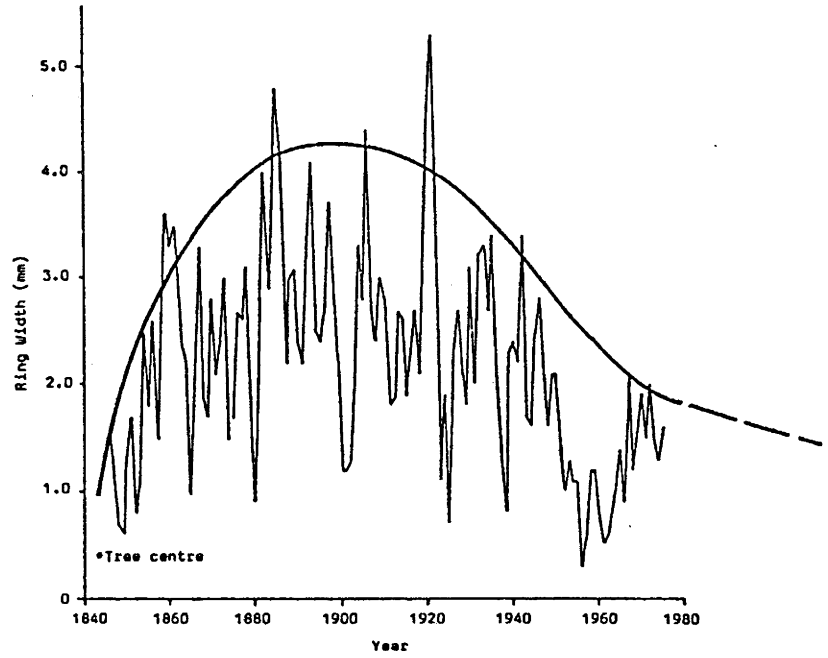

Figure 2.3: Annual radial growth ring data from an open-grown tree of *E. pauciflora*; curve drawn by hand as an 'envelope' (reproduced from Banks 1982)

maximum growth increments in BRIND is discussed in more detail in Section 2.2.2.

(iii) The index T(DEGD) models the effect of temperature on photosynthesis (Botkin *et al.* 1972). It is calculated from the range of growing degree-days (DEGD) in which a species occurs (Table 2.1). Degree-days is an integrative estimate of the thermal energy available to the plants for growth in a year. It is defined as the sum of (mean daily air temperature) – (base temperature) over a year, where 'base temperature' is that below which growth is assumed to stop. Days with mean temperature below the base temperature do not contribute to degree-days. Shugart and Noble (1981) found a very good correlation between latitude, altitude and DEGD for most of Australia and the regression model fitted to these data was used in calculating species parameters for BRIND. T(DEGD) is a function with parabolic shape between minimum and maximum DEGD values for a species and zero outside that range. The parabolic section of T(DEGD) ranges from zero at $DEGD_{min}$ and $DEGD_{max}$ to one at $(DEGD_{max} - DEGD_{min})/2$ (Table 2.2, Eqn 2.7; Fig. 2.4).

Along the gradient of DEGD, the T(DEGD) index has the effect of interacting with G to modify species' growth rates, altering their relative competitive abilities. This interaction of T(DEGD) and G models the response of the species to the gradient of DEGD. The T(DEGD) index is discussed in detail in Section 2.2.3.

(iv) The index r(AL) is an index of competition for light, where each tree is considered to be shaded by all taller trees on the small plot. Leaf area (adjusted for plot size and expressed as leaf area index) of all the trees is calculated using Eqn 2.10 (Table 2.2). This leaf area equation was derived

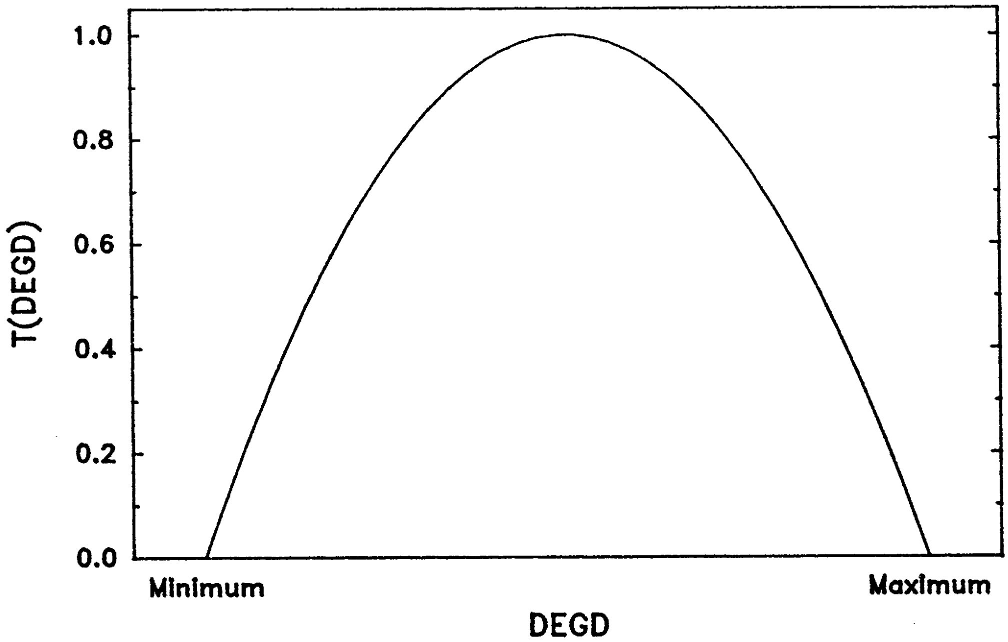

Figure 2.4: The T(DEGD) index, calculated from Eqn 2.7. 'Minimum' and 'Maximum' are species parameters supplied to BRIND ($DEGD_{min}$ and $DEGD_{max}$, Table 2.1.)

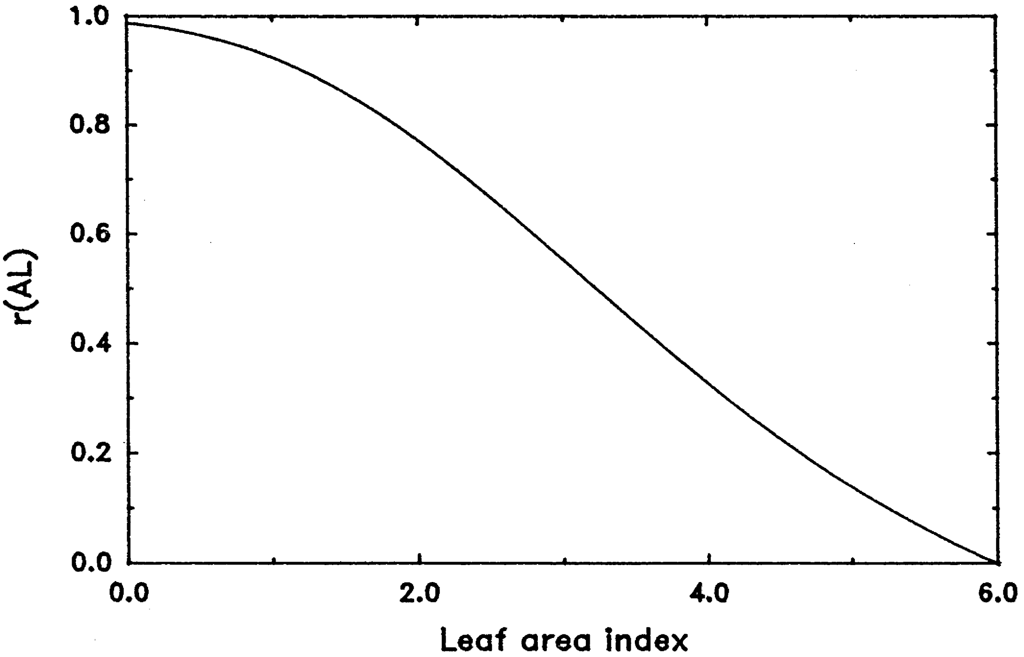

Figure 2.5: The r(AL) index, shown as a function of leaf area index, calculated from Eqns 2.8 and 2.9. For any tree, leaf area index is calculated from the summed leaf areas of taller trees on the plot.

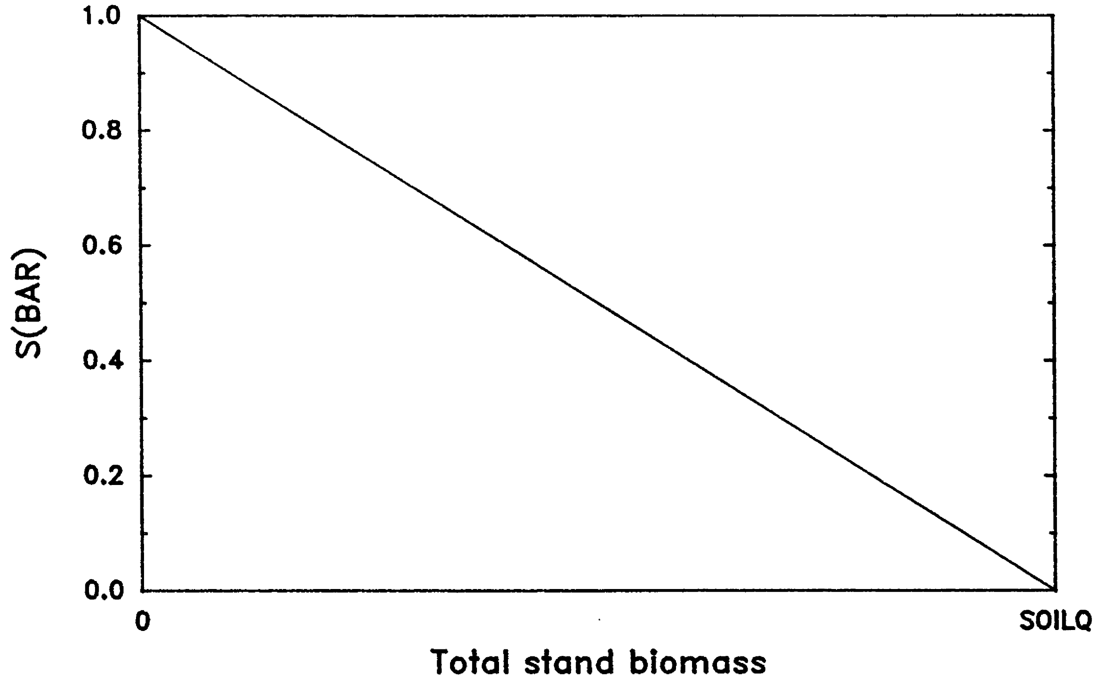

Figure 2.6: The S(BAR) index, calculated using Eqn 2.11.

from Attiwill's (1966) empirical equations relating leaf mass to DBH in *E. obliqua* and Ashton's (1976) equations relating leaf area to leaf mass in *E. regnans* and *E. sieberi*. For each tree, the leaf areas of taller trees are summed and attenuation of light by the total leaf layer calculated. The index r(AL) is calculated from the reduced light level (Table 2.2, Eqn 2.8; Fig. 2.5). In BRIND all species have the same response to light but other gap models have two or more equations for different levels of tolerance to shading (see Shugart 1984 for examples). Competition in BRIND is discussed in more detail in Section 2.2.4.

(v) The last index, S(BAR), is a simple, linear model of below-ground resource use and is governed by the plot parameter of maximum biomass called SOILQ (Table 2.2, Eqn 2.11; Fig. 2.6). The equation relating above-ground biomass to DBH (Eqn 2.12) was taken by Shugart and Noble (1981) from an empirical relationship established for *E. regnans* and *E. sieberi* by Ashton (1976). The upper limit on plot biomass set by SOILQ is unlikely to be approached closely because at these biomasses the S(BAR) index has a low value that reduces the growth of all trees severely (Fig. 2.6). The S(BAR) index is discussed further in Section 2.2.4.

Some gap models incorporate other indices that affect growth, e.g., indices of soil moisture availability (Pastor and Post 1986) and soil nutrients (Weinstein *et al.* 1982; Pastor and Post 1986). Pastor and Post also changed the growth model so that the maximum growth increment was only multiplied by the index having the smallest value, rather than by all of them. The rationale for this ('Liebig's law of the minimum'; Liebig 1840) is that when one factor is limiting growth, variation in others has little effect, so only the index with the smallest value should be used to modify growth; use of all of them together would reduce growth too much. Pastor and Post (1986) found that their model made more accurate predictions of stand dynamics using the minimum index value instead of using all of them.

2.2.2 Maximum growth increments

The equation that calculates maximum growth of trees (Eqn 2.5) theoretically predicts their growth under physiologically optimal conditions. Growth of trees under these conditions is not restricted by availability of light, water or nutrients and is limited only by genetically-determined internal (physiological) constraints. The parameter G sets the maximum growth rate for trees of each species and thus expresses physiological potential for growth. The parameters D_{max} and H_{max} set the limits of DBH and height attainable by the trees of each species.

Values of $D_{\max}$ and $H_{\max}$ used in BRIND were taken by Shugart and Noble (1981) from Hall *et al.* (1970)[1]. These values are taken from the largest trees known, that are presumably open-grown. While open-grown trees will have encountered little or no competition, they will have grown in the presence of herbivores and pathogens and under conditions of fluctuating climate. Thus, open-grown trees will not have grown under *physiologically* optimal conditions for *all* of their lives, as assumed by the $D_{\max}$ and $H_{\max}$ parameters.

Assuming that values of maximum DBH and height attainable under physiologically optimal conditions are greater than those attained by open-grown trees, the values of $D_{\max}$ and $H_{\max}$ from open-grown trees are underestimates of their true values. Because G is a monotonically increasing function of both $D_{\max}$ and $H_{\max}$ (Eqn 2.6) its values will also be underestimated. Values of G calculated using Eqn 2.6 are inversely proportional to $\mathrm{Age}_{\max}$ for a species, so that uncertainty in the estimation of $\mathrm{Age}_{\max}$ results in added uncertainty in the value of G.

The equation most often used to calculate G (Eqn 2.6) is based on an assumption which has no theoretical basis, that a tree will reach two-thirds of its maximum DBH in half of its maximum age (Botkin *et al.* 1972). Botkin *et al.* state that empirical data may show values of G calculated in this way to be too low or too high and that they should be adjusted if found to predicting unrealistic growth rates. They suggest that G can also be calculated from maximum annual DBH increment data for the species and suggest the relationship:

$$G \approx 5\,(\Delta D)_{\max}\,H_{\max}/D_{\max} \tag{2.13}$$

---

[1] Shugart and Noble (1981) gave $D_{\max}$ and $H_{\max}$ measurements to the nearest centimetre but these seemingly precise values are just metric conversions of measurements given in rounded imperial units by Hall *et al.*—for example, $D_{\max}$ for *E. dalrympleana* of 4 feet became 122 cm, and $H_{\max}$ for *E. delegatensis* of 200 feet became 6096 cm.

based on the implicit assumption that potential physiologically maximum growth rate is reflected in largest observed growth increments. In a number of gap models, growth data have been available and used to determine values of G (e.g., the FORCAT model of Waldrop *et al.* 1986).

Moore (1989b) found G calculated by Eqn 2.6 to be to small for *Pinus virginiana* according to published field data and recommended that field data should be used if at all possible. He also suggested a simplification of the maximum growth equation which he describes as biologically more reasonable than Eqn 2.5:

$$\Delta D = \frac{GD\,(1 - H/H_{\max})}{274 + 3b_2D - 4b_3D^2} \tag{2.14}$$

A further complication with G is that the value derived from the largest or fastest-growing tree of a species is used to describe the potential growth rate of all trees of that species. There is always genetic variation between organisms within a species and the largest or fastest growing tree of a species may be genetically atypical. Thus a value of G is calculated that underestimates the actual potential growth rate of the largest/fastest-growing tree of a species from which it was derived (as discussed above) but which may be fortuitously closer to the actual potential growth rate of most trees of that species.

Obtaining growth rate information from trees in the field is further complicated by differences in microhabitat (the immediate growth environment of each tree). Banks (1982) found variation in growth of *E. pauciflora* within a small area which he attributed to microhabitat differences. It is practically impossible to separate genetic from microhabitat contributions to variation in field growth rates. Within the gap model structure, an appropriate change is to use a distribution of values of G for each species. The resultant variation in simulated growth rates would simulate genetic variation between species and differences between trees' microhabitats, neither of which are modelled by BRIND.

In summary, the maximum growth increment equation used in gap models is intended to predict the growth potential of all trees of a species to physiological limits but actually describes the growth to ecological limits of the largest or fastest growing trees. The models may or may not predict with accuracy the potential maximum growth rates of 'average' trees of a species. However the assumption that all trees of a species have the same potential growth rate and the mismatch between the physiological basis of the growth equation and ecological determination of parameter values reduces their realism.

While predicted rates of growth may be inaccurate, it is comparative growth rates between species that determine competitive success (see Section 2.2.4, below). A gap model may thus predict realistic forest composition at a site while simulating inaccurate growth rates of the species. In Chapter 4 I use radial growth data collected from trees of *E. pauciflora* to estimate variation in G for this species, allowing for environmental effects on growth.

2.2.3 Response to environmental gradients

> The function T(DEGD) ... represents an attempt to take account of the effect of temperature on photosynthetic rates. (Botkin *et al.* 1972, p. 856)

T(DEGD) is therefore a theoretical model of the physiological response of tree species to temperature. But estimation of the species parameters $\mathrm{DEGD}_{\mathrm{min}}$ and $\mathrm{DEGD}_{\mathrm{max}}$ suffers from the same problem as estimation of D_{max} and H_{max}, described in Section 2.2.2. The $\mathrm{DEGD}_{\mathrm{min}}$ and $\mathrm{DEGD}_{\mathrm{max}}$ parameters represent physiological temperature limits to tree growth, but their values are calculated from field distribution (i.e., ecological) data: extremes of species ranges along the gradient of temperature (Botkin *et al.* 1972; Shugart and Noble 1981). Clearly, while T(DEGD) is intended as a physiological index of growth, it functions as an ecological index. The ideal situation would be to have physiological data available for all species modelled, but the paucity of such data and the difficulty in collecting them make it necessary to use an ecological index.

The major consequence of using the T(DEGD) index with ecologically-derived parameter values is that BRIND predicts zero radial growth of all trees of a species at the extremes of its ecological distribution (DEGD range: Fig. 2.4). Therefore, it is implicitly assumed in the formulation of BRIND that species distributions are principally governed by predicted growth rates. I believe that this assumption is unrealistic, and is a structural weakness in BRIND and other gap models that use T(DEGD) in the same way. BRIND may predict realistic species ranges along the DEGD gradient, but does so by predicting unrealistic growth rates at the extremes of those ranges (most of them—it is possible that the extreme of a species' ecological range coincides with that of its physiological range).

In reality, the presence of a tree at any site indicates that, for its species, propagules were available and that environmental conditions were suitable for establishment and growth. The success of an individual plant at a site will depend on its weakest life-stage being able to survive the environment at that site. (By 'success' I mean the ability to grow to reproductive size and produce seeds, and by 'weakest' I mean the life-stage most susceptible to competition for resources in its immediate above- and below-ground environment.) The weakest life-stage of any plant species is the smallest, after any seed reserves have been used up (Grubb 1977).

In BRIND and other gap models where new trees are added to the simulated plot at sapling size (Section 2.3), the model should consider all factors that affect trees from germination to that size in selecting species and numbers of trees to establish. Therefore in BRIND I believe it is more realistic to model species' responses to temperature and other gradients primarily as ability to establish rather than as rate of growth. Establishment is discussed further in Section 2.3, and in Chapter 5.

Another consequence of using an ecological T(DEGD) index is that BRIND may be unable to predict changes in a species' ecological response under changed conditions of competition. The ecological response of a plant species results from modification of its physiological response by the presence of other organisms, commonly competitors (Chapter 1). If a competitor is removed or a new one invades, the ecological response of a species may change to a greater or lesser degree (Ellenberg 1953; but see criticisms by Ernst 1978). Because the shape of a species' physiological response is not known, it is difficult to predict the change in its ecological response brought about by a change in competition. Figure 2.7 shows the physiological and ecological responses of a hypothetical species along a single gradient, where the peak of response is shifted by the presence of a competing species (from Mueller-Dombois and Ellenberg 1974). If the competing species were removed, the growth model in BRIND would not be able to predict the increased response and shift in peak that results. This problem apparently did not occur with the FORET model of Shugart and West (1977), which predicted composition of forests with and without chestnut (*Castanaea dentata*).

I have argued that realism of gap models is reduced by the use of ecological response data because the model equations are based on physiological responses. As with the analogous problem in prediction of maximum growth increments (Section 2.2.2), it is necessary to use ecological data because the physiological data are not available, and are very difficult and time-consuming to obtain. The reduction in realism here presents a fundamental limit to the progress of modelling growth of trees from simple physiological principles and—importantly—presents a limit

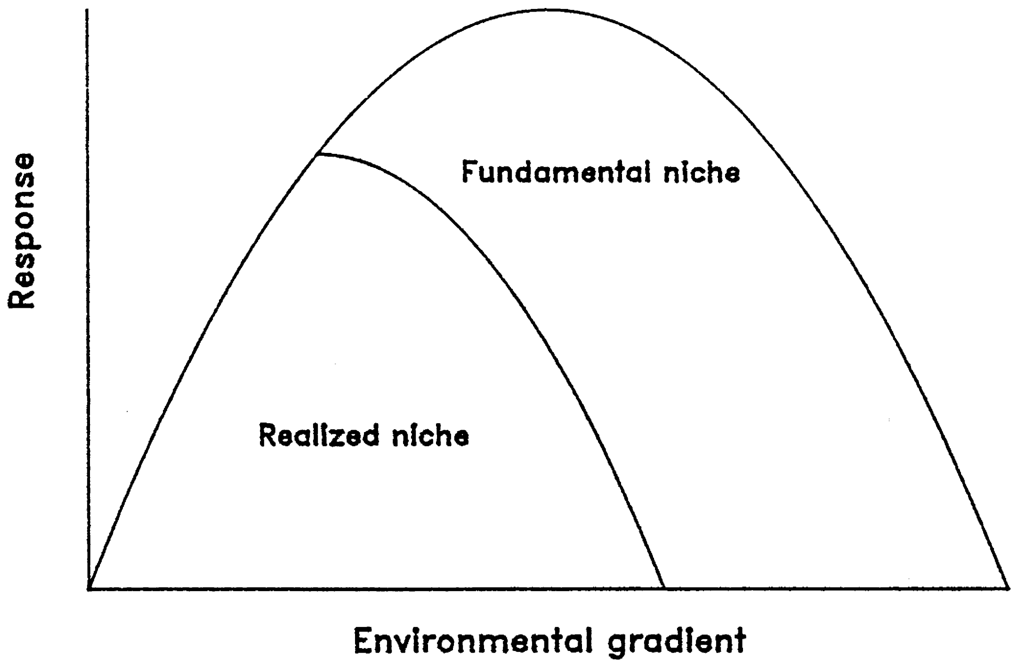

Figure 2.7: Diagram of hypothetical physiological and ecological response curves for a species. From Mueller-Dombois and Ellenberg (1974).

---

to the reliability of predictions by such models in extraordinary (extrapolative) circumstances.

In developing the FORCAT gap model, Waldrop *et al.* (1986) introduced further variation in G with environment by using, for each species, different values of G at each of 20 land types in their study area. They calculated G from increment core data for one land type, and estimated values for the other 19 from published site index information. Waldrop *et al.* thus tailored variation in G specifically to their study area (although FORCAT maintains use of the T(DEGD) index). They did not collect detailed growth information for all 20 land types, and suggest that estimated G values will probably need adjustment as data become available.

In Chapter 4 I use tree-ring data to examine the effects of temperature and soil moisture on growth in *E. pauciflora* and to construct an alternative model to T(DEGD), which does not necessarily reduce radial growth of all trees to zero at the extremes of species' ranges. In Chapter 5 I investigate the use of statistical descriptions of species distributions as models of establishment probability. These alternative models are constructed with the aim of shifting the emphasis of modelling environmental response from growth of established trees to establishment of new trees.

2.2.4 Competition

The light competition index in BRIND is the same as that in FORET (Shugart and West 1977) and JABOWA (Botkin *et al.* 1972), and a number of other gap models (Shugart 1984). Canopy structure is simple, with no vertical canopy overlap: a tree is shaded by all leaves of all taller trees, and all of its leaves shade those of all shorter trees. This means that each individual canopy is modelled as a thin layer at the height of its tree. There is also no explicit horizontal structure in gap models: each tree shades or is shaded by all other trees on the simulated plot.

A modification of this canopy model was used by Leemans and Prentice (1987) in their FORSKA gap model. They calculated canopy depth of the conifers they were modelling and used that to calculate extent of vertical canopy overlap. Vertical overlap is important under low light angles that are common at high latitudes.

Noble *et al.* (1988) compared predictions of the simple vertical canopy model used in JABOWA and BRIND with those made by more complex structural models incorporating varying amounts of vertical canopy overlap (including that of Leemans and Prentice 1987). They found that changing the canopy structure made little difference to predicted competitive effect, and that predictions are more sensitive to the value of the light extinction coefficient (see below).

Similarly, Lorimer (1983) compared competition indices having simple horizontal structure (like that in BRIND) with models that use inter-tree distances to weight competitive effects. He found that the more complex indices did not correlate better with field growth data than the simple ones. Lorimer suggested that, within a certain radius of a tree, local density of competitors was as good a predictor of competitive effects as detailed inter-tree distance information.

Values used for k, the light extinction coefficient (Eqn 2.9), vary between gap models, but the outcome of competition as predicted by r(AL) is sensitive to the its value (Noble *et al.* 1988). Botkin *et al.* (1972) suggested the value of k be adjusted to give "reasonable shading" but Shugart and West (1977) gave no reasons for their choice of values. Values of k taken from published empirical data were used in BRIND (0.5; I. R. Noble, personal communication) and in SILVA (0.47; Kercher and Axelrod 1984). Landsberg (1986) stated that values of the light extinction coefficient measured from forest canopies vary from 0.3 to 1.5 depending on species and canopy density but an average value for coniferous and broad-leaved forests has been found to be *c.* 0.5 (when leaf area is expressed as leaf area index).

In the terminology of modelling described in Chapter 1, it is apparent that k has been tuned in some cases to produce acceptable model predictions. Because this parameter has a clear theoretical basis (in extinction of light by forest canopies) it should not be tuned in this way. However values of k in real forests are empirically (as opposed to theoretically) derived, and the average of 0.5 is the only indication of the value it *should* have.

In gap models, the amount of growth reduction from shading is a function of k (Eqns 2.8 and 2.9), so k is effectively an expression of the intensity of competition between the trees. Increasing the value of k in a model reduces the probability of survival of small trees through increased suppression, and thus reduces the number of trees on the plot.

Even though r(AL) uses a simplistic canopy structure, this structure is an idealisation (Chapter 1) that has been shown to have little effect on model predictions. I believe r(AL) works because it is a one-sided competition model in a general sense: a tree adversely affects growth of all smaller trees on the plot but is unaffected by them. One-sided competition models have been found by some authors (e.g., Lorimer 1983) to be correlated with observed growth rates as well or better than more complicated models. Weiner and Thomas (1986) found that plant competition is primarily one-sided (which they called 'resource pre-emption'). They suggested that asymmetry in competition is not complete (where larger plants capture all of the available resources) but they acquire a disproportionately large share of them.

In BRIND the advantage in competition is in favour of taller trees so two trees of the same species growing on the plot will always maintain the same ranking in size. A species with a fast rate of height growth (such as *E. delegatensis* in BRIND, $G = 235$: cf. *E. dalrympleana*, $G = 80$; *E. pauciflora*, $G = 135$) will dominate the simulated plot once established, provided that its temperature (DEGD) environment is suitable. A small tree of a fast-growing species may be able to 'catch up' to a larger, slower-growing tree in spite of being in shade if its growth rate is fast enough and the size differential is not too great.

It is the comparative rates of height growth that principally affects outcome of competition in BRIND. Height growth rate is a function of DBH growth rate (determined by G) and the DBH–height relationship for a species (Eqn 2.4). The model may make incorrect predictions about outcomes of competition between species because of uncertainty in values of G (Section 2.2.2) and the DBH–height relationship for a species. Growth rate is modified by T(DEGD) along the gradient of temperature but there is uncertainty about the realism of T(DEGD) as a model of temperature effects on growth (Section 2.2.3).

In BRIND the relationship between diameter and height for a species remains the same under all environmental conditions (and degrees of competition). This idealisation reduces precision in predictions of model growth but greatly simplifies the growth model; also data of variation in the relationship between diameter and height are available for few species. However it may cause BRIND to make incorrect predictions about competition in a case where the DBH–height relationship for a species changes with its environment.

The S(BAR) index (Eqn 2.11) is described by Botkin *et al.* (1972, p. 856) as a "crude expression of the competition for soil moisture and nutrients on the plot." The SOILQ parameter represents the maximum basal area (JABOWA) or biomass (FORET and subsequent gap models) attainable on the plot. From the description by Botkin *et al.* (1972), its value should be adjusted according to site differences (in JABOWA it is adjusted according extent of rock outcropping in the simulated plot).

Shugart and Noble (1981) appear not to varied SOILQ across the range of altitudes. Shugart and Noble calculated the value of 1000 t/ha from measurements of old stands of *E. delegatensis* in the Brindabella Range. Some of the forest types modelled by BRIND (e.g., stands of *E. pauciflora* at higher altitudes) have an open structure and are not as tall as *E. delegatensis* stands (ANU Forestry 1973). The productive capacity of such stands (represented by SOILQ in BRIND) appears to be much lower than 1000 t/ha.

The S(BAR) index reduces growth of all trees on the simulated plot and, importantly, is the only index that reduces growth of the tallest tree or trees as plot biomass accumulates. It represents the accumulation of nutrients in living biomass and limits to productivity set by amounts of soil resources. The gradients of increasing soil nutrients and rainfall with altitude (Chapter 1) in the Brindabella Range suggest that SOILQ should increase with altitude. But stands at the highest altitudes (dominated by *E. pauciflora*) are more open and less productive than some at intermediate altitudes so growth is limited by some other factor, most likely temperature or soil moisture.

In Chapter 4 I use tree-ring data to investigate effects of competition on growth in *E. pauciflora* and to compare alternative models of competition. In Chapter 3 I examine sensitivity of biomass prediction by BRIND to different values of SOILQ.

2.3 Establishment

Each year of simulation, new individuals can become established on the plot. The modelling of establishment varies greatly in detail between gap models but all in principle determine which species are eligible to become established according to conditions that year and select how many trees of which species actually do. There is a large stochastic component in all of these establishment models.

Establishment in BRIND (subroutine BIRTH) is modelled in a number of stages. Firstly, the number of sites to be occupied by new trees is determined. This number is from one to an upper limit, randomly chosen from a uniform distribution. In the absence of fire, the upper limit is 4 in most years and 30 in 10% of years (randomly chosen with an annual probability of 0.1). In fire years, the upper limit of number of sites is 200 to reflect the clearing of understorey and litter by fire and the 'ashbed' effect (Pryor 1960), or perhaps seed predator satiation (O'Dowd and Gill 1984).

A 'seed pool' is then constructed from the species that are eligible to establish in that year according to some of the species parameters shown in Table 2.1. Species are randomly chosen to occupy the sites for establishment according to their proportional contribution to the seed pool. A species is eligible if the DEGD value for that year falls within the range it can tolerate (the same range used in modelling growth) and if the number of frosts in that year is less than the maximum number it can tolerate (FRSTMX, Table 2.1).

A species' contribution to the seed pool (SEEDS, dimensionless) is set to 1.0 if it has mature trees (DBH > SDSIZE: Table 2.1) present on the plot. When all the mature trees of a species die, decay of remaining seeds on the plot is simulated by a seed decay factor:

$$s = \exp(-2.30/\mathrm{SDLIFE}) \tag{2.15}$$

where SDLIFE is the longevity of seeds of the species. SEEDS is multiplied by s, resulting in 10% of seeds to surviving to their maximum life-span (SDLIFE).

A small addition is made to SEEDS for each species to simulate dispersal into the plot. This addition ensures that any species has the potential to become established within its DEGD range by setting a minimum value for SEEDS: 0.01 for most species and 0.04 for species that are well dispersed (non-eucalypt and non-acacia species; see Shugart and Noble 1981). Species that show enhanced germination after fire (acacias and *E. delegatensis*) have their values of SEEDS increased by a factor of 3 in fire years and those that require fire to trigger germination (acacias) have theirs reduced by a factor of 1000 in non-fire years. Shugart and Noble (1981) experimented with different values of many of these parameters and found predictions by BRIND to not be sensitive to them.

One important difference between BRIND and JABOWA or FORET is that in the latter two models all eligible species are equally likely to establish in any year, whereas in BRIND the selection of species is strongly influenced by the species composition of the plot. Botkin *et al.* (1972) based their establishment model (where species are chosen with equal probability) on a lack of empirical correlation between species present and those establishing, whereas with BRIND

Shugart and Noble (1981) were modelling the limited dispersal capabilities of eucalypts by assuming that species that have mature trees present on the plot are more likely to establish than those that do not.

While BRIND weights species' probabilities of establishment according to presence of seed-bearing trees, there is no distinction made between a species with one small mature tree and another with many large individuals. Larger trees produce many more seeds (Jacobs 1955), a fact that should be incorporated into the establishment submodel.

In BRIND, trees are added to the plot as small saplings with a DBH randomly set between 1.2 and 1.3 cm. The addition of trees of this size to the plot is computationally much simpler than maintaining a list of individuals shorter than breast height (i.e., without DBH) and makes little difference to stand dynamics because small individuals have little effect on competition. Addition of trees to the plot at sapling size means that all the establishment and growth of trees to sapling size is modelled as occurring in one year and that the number of trees 'born' is actually the number surviving all early life stages. Therefore the establishment submodel in BRIND should take account of all the factors that affect growth of trees to sapling size, not only those affecting germination of seeds and establishment of seedlings.

Other gap models incorporate different constraints on species' abilities to establish, e.g., presence of mineral soil or leaf litter, temperature or herbivore abundance (see Shugart 1984 for a review).

In BRIND, environmental constraints on establishment are simple: a species can establish with equal probability anywhere within its DEGD range if there are not too many frosts occurring. Shugart and Noble (1981) did not state how many frosts per year were included in the simulations they described; if it was the default value of 60 set by the program (Noble *et al.* 1980a) then it had no effect on species establishment. I believe that this uniform rate of establishment throughout a species' environmental range is unrealistic because the occurrence of a species at a site will be determined by the ability of its most vulnerable life-stage (i.e., its smallest one) to survive.

Jacobs (1955) described variation in seed production in *E. resinifera*, which produces much more seed in the part of its range where it shows optimum development (Boland *et al.* 1984). It is reasonable to expect that other eucalypt species behave in a similar fashion and the evidence of Austin *et al.* (1983, 1984, 1990) that eucalypt species are less common towards the extremes of their ranges supports this.

In Section 2.2.3 I argued that the reduction of radial growth to zero at the extremes of a species field distribution is unrealistic. The coupling of such growth response with uniform establishment along the gradient of temperature is the product of unrealistic structure in BRIND: it is implicitly assumed by the model that species distributions along the gradient are primarily controlled by growth rates and not establishment probability. In BRIND, reduced growth can result in less establishment: seedling input is dependent on mature trees being present, but a species with one small mature tree is as likely to establish as one with several large trees, so the effect of slow growth on establishment is reduced.

A more realistic model would not reduce growth of all trees to zero at the extremes of a species' ecological range, but would restrict its ability to establish to sapling size. In Chapter 5 I investigate modelling establishment probability as a function of environmental gradients.

Lignotuberous seedlings are more common than true seedlings in many eucalypt forests (Florence 1981; Noble 1984) and the BIRTH subroutine in BRIND is functionally a model of their emergence as saplings into the tree layer of the forest. In Chapter 8 I introduce an enhancement of the establishment submodel where lignotuberous seedlings have greater longevity to simulate time spent in the understorey in the absence of mature trees and are introduced to the plot at a larger size than non-lignotuberous seedlings.

2.4 Mortality

BRIND uses a model of tree mortality with three components: 'background', 'suppression' and fire-damage mortality.

It is assumed that all trees are subject to a constant low rate of mortality that is independent of tree age and size and of any other trees on the plot, representing random events of low probability that can result in the death of any tree at any time ('background' mortality). The annual probability of background mortality for a species in BRIND is derived from its maximum age parameter so that it has a probability of 0.01 of reaching that age (Table 2.1):

$$p = 1 - \exp(-4.605/\mathrm{Age}_{\mathbf{max}}) \tag{2.16}$$

In addition to this, trees that survive a fire are considered to be damaged by it and have their probabilities of mortality increased for some years afterwards (Section 2.5).

The second part of the mortality model assumes that trees whose growth rates are reduced below a certain level are more susceptible to death from other causes. These are trees whose growth is suppressed by competition from other trees or unfavourable climatic (DEGD) conditions and large trees whose DBH is approaching the maximum value for their species (Fig. 2.2). These trees with reduced growth are subject to a probability of mortality of 0.368, which gives them a 1% chance of surviving 10 years in the suppressed state. The criterion of suppression in BRIND is an annual DBH increment of less than 0.1 cm.

In a recent review of mortality in forest trees, Harcombe (1987) described typical mortality curves for trees as being concave: small trees suffer high mortality that reduces to a much lower rate as they reach a certain size, that size varying between species and stands. High mortality of small trees is associated with suppression from larger ones, while the low rate of mortality found in larger trees is attributed to a variety of unpredictable causes. There is also some evidence for a 'U'-shaped mortality curve where very large trees suffer increased mortality (Harcombe 1987). The treatment of mortality in gap models theoretically replicates the behaviour described by Harcombe, where suppression is the major cause of mortality in small trees and larger trees free from suppression are principally killed by apparently random, density-independent causes.

Even though the idea of background mortality from assorted random events is supported by evidence (Harcombe 1987), the probabilities calculated in BRIND from life expectancy are too high for realistic annual mortality. As an illustration, the maximum age parameter for *E. pauciflora* is 125 years, from which an annual probability of mortality of 0.036 is calculated. While this annual probability appears quite low, the probability of any tree being killed in 10 years is $1-(1-0.036)^{10}$, which is 0.308. In a stand of *E. pauciflora*, nearly a third of the trees will be killed by BRIND in any 10 years, *irrespective of size and position in the canopy*. For longer-lived species, the proportion over 10 years would be lower: 0.206 for 200-year, 0.142 for 300-year and 0.109 for 400-year species. There is no evidence for such high mortality rates in established eucalypts in the absence of disturbance. Even though BRIND may predict correct plot biomass or species distributions, it predicts unrealistic survivorship.

Harcombe (1987) did not give exact annual mortality rates for individual species, but the species with lower rates (*c.* 0.0009–0.005) included *Fagus grandifolia* and *Acer saccharum* that are modelled by FORET as having background mortality rates of 0.011 (from $\mathrm{Age}_{\mathrm{max}} = 400$) and 0.015 (from $\mathrm{Age}_{\mathrm{max}} = 300$), respectively (Shugart and West 1977). These rates suggest all gap models that use Eqn 2.16 for calculating background mortality rates predict excessive mortality.

In Chapter 6 I examine life tables predicted by BRIND and investigate effects of lower background mortality rates on model behaviour.

2.5 Fire response

The modelling of fire and its effects in BRIND is in two stages: calculation of fire and flame characteristics and then of response of each tree to the fire.

Intensity of the fire is expressed as flame height; this is calculated using equations derived by Noble *et al.* (1980b) from McArthur's (1967) forest fire danger meter. The McArthur meters were designed to be used on a day-to-day or hour-to-hour basis to predict a fire danger index (FDI) from current and recent local climatic conditions and then to predict fire characteristics (rate of spread, flame height and spotting distance) from the index. The formulation of BRIND does not include short-term climatic information, so FDI is randomly selected by the model from a normal distribution whose mean and standard deviation can be set by the model user to meaningful values (Noble *et al.* 1980a; Shugart and Noble 1981).

From the stochastic value of FDI 'on the day of the fire' BRIND calculates fire rate of spread (in m/hr) using fine fuel weight and slope of the site. Fuel load is calculated from time since last fire using a negative exponential equation and slope is a site parameter specified by the user (Shugart and Noble 1981). A fire with rate of spread less than 1 m/hr is assumed not to spread and does not affect trees on the plot. The rate of spread is then used with the fuel load to calculate flame height in metres (Noble *et al.* 1980b).

Kessell and Good (1985) and Kessell (1990) have shown that the McArthur (1967) meters and the equations derived from them (Noble *et al.* 1980b) can make misleading predictions about flame characteristics. Kessell (1990) presented rates of spread measured from 32 fires in New South Wales with predictions by the McArthur meter equations and the 3-strata Rothermel model (Kessell 1979). Ten of the fires were in eucalypt communities like those found in the Brindabella Range; in those 10, rates of spread calculated by the McArthur equations were *c.* 7 times the measured rates on average, while those calculated by the 3-strata Rothermel model were within 15% of the measured values on average.

Similarly, in 10 000 years of simulations with BRIND the McArthur meter equations predicted mean flame height from 189 fires of 17 m (for a slope of 10°). Fires of this magnitude are crown fires of high intensity and can occur, but would not be expected on average across all sites (Luke and McArthur 1978).

The 3-strata Rothermel model of fire behaviour is the most consistently accurate one available, being based on combustion physics rather than empirical behaviour data (Kessell 1990). But it also requires extremely detailed information about fuel loads (quantity and moisture content in several size classes in three strata) and small-scale weather conditions (at the fire front, not above the forest canopy) (Kessell 1988).

I am not concerned with detailed fire behaviour modelling at small time-scales in my improvements to BRIND but a better method of predicting intensity of fires occurring in random years is needed. As a temporary 'fix', in Chapter 7 I scale predictions of flame characteristics by the McArthur equations to realistic values.

Fire response of trees in BRIND is calculated from flame height, tree height and canopy depth. It is assumed that if flame height is greater than 16 m the fire is a crown fire and flame height is reset to a very large number. Leaf scorch height is calculated from flame height and is assumed to be six times flame height (Luke and McArthur 1978). The proportion of each tree's canopy scorched by the fire is calculated from scorch height and canopy depth, which is calculated from tree height and form factor (Table 2.1). Probability of immediate mortality from fire is calculated from proportion canopy scorch (see below). Trees surviving the fire have a damage factor calculated from proportion canopy scorch that subjects them to increased mortality during the following years.

Parameter values used in the fire mortality equations divide the species into those that are sensitive to fire and those that are resistant. *E. delegatensis* (a non-lignotuberous species) and all non-eucalypt species are modelled as sensitive, and other eucalypts as resistant. Gill (1981b) described tree species as falling into two such categories, but the parameter values used in BRIND for the species result in nearly all trees of fire-sensitive species being killed in any fire and very few of fire-resistant species. The equation used to calculate probability of mortality from fire is:

$$p = \mathrm{FA} - \mathrm{FB}\exp(-\mathrm{FC} \times \mathrm{PSCH}) \tag{2.17}$$

where PSCH is proportion canopy scorch; FA, FB and FC are species fire-response parameters (Table 2.1; see Shugart and Noble 1981 for their derivation). Equation 2.17 is a negative exponential: as PSCH increases, calculated probability tends toward an asymptote specified by the parameter FA; and when PSCH is zero, the probability is FA − FB (if the probability is calculated as a negative number, it is interpreted as being zero). All fire-sensitive species have values for FA and FB of 2.0 and 4.0—the value for FB of 0.02 given by Shugart and Noble (1981) is erroneous (I. R. Noble, personal communication)—and fire-resistant species values of 0.01 and 0.02.

Trees of fire-sensitive species in BRIND can survive 50% canopy scorch (Luke and McArthur 1978), but most are usually killed by the model because the McArthur meter equations predict excessive flame heights (see above). Fire-resistant trees have a maximum probability of 0.01 of being killed by a fire (FA – FB, Eqn 2.17), so very few are killed directly by fire, but are subject to increased mortality after fire (see below).

For trees of fire-resistance species that survive a fire in BRIND a fire-damage factor (FARDAM) is calculated that is used in the KILL subroutine to increase the tree's probability of mortality. FARDAM has the value 1.0 for an unburnt tree and is increased after fire by a function of PSCH: a fully-scorched tree of a fire-resistant species has its probability of mortality increased by a factor of *c.* 3.6 in the first year following the fire. FARDAM (whose value is lower for trees with less than full canopy scorch) is reduced in non-fire years so that it decreases by *c.* 50% every ten years (Shugart and Noble 1981). The fire damage factor contributes significantly to mortality of trees because it is multiplied by the annual background mortality rate—which is unrealistically high in the absence of fire (Section 2.4).

Fire effects are only modelled in BRIND as immediate death or increased mortality of trees. Many eucalypt species are known to be able to survive fire and re-sprout from epicormic and lignotuberous buds (Gill 1975, 1981a, 1981b). In Chapter 7 I present an alternative model of fire response that predicts a range of responses of fire-resistant trees, derived from field data collected after a wildfire in eucalypt forest in the southern Australian Capital Territory.

2.6 Sprouting

The SPROUT subroutine in BRIND simulates sprouting of trees of suitable species by the addition of 'sprouts': saplings like those added by the BIRTH subroutine (Section 2.3). Sprouting species have a parameter that sets the minimum DBH for a tree to sprout (SPRTMN, Table 2.1). Lignotuberous eucalypt species have a value for SPRTMN of 0 cm to simulate sprouting from the lignotubers of even very small trees.

Upon the death of any tree of a sprouting species with DBH greater than SPRTMN, there is a probability of 0.75 that it will be replaced by a sprout. Sprouts are considered more vigorous than seedlings and are given a starting DBH of between 2.0 and 2.1 cm (with random variation in size, as in the BIRTH subroutine). Most species in BRIND can sprout, the exceptions being *E. delegatensis* and *E. fastigata*, that do not produce lignotubers, and *Banksia marginata*.

Basal sprouting in eucalypts is generally only seen as a response to fire when above-ground parts of a tree are killed but the lignotuber survives (Jacobs 1955). Many eucalypt species also sprout from epicormic buds under their bark, particularly in response to fire (Gill 1981b). Epicormic sprouting is not modelled by BRIND but could be simulated as part of the process of recovery from fire.

The main effect of the SPROUT subroutine in BRIND is to increase the numbers of new trees added to the plot each year, especially lignotuberous eucalypt species, that constitute most of the species simulated by BRIND. This model behaviour is likely to result in too many trees becoming established.

The submodel of fire response described in Chapter 7 includes modelling lignotuberous and epicormic sprouting as explicit responses to fire, rather than as a response to tree death generally.

2.7 Conclusions

In this chapter I have criticised details of the structure of BRIND, arguing that some of its component submodels are unrealistic representations of forest dynamic processes. The areas where BRIND is lacking in realism are:

1. The confusion between physiological design and ecological implementation of parts of the growth equation (Sections 2.2.2 and 2.2.3), that has two effects:
   (a) Values of $D_{\max}$, $H_{\max}$ and G calculated from field data are likely to be inaccurate estimates of the physiological parameters they represent.
   (b) The T(DEGD) equation, that predicts growth response to temperature, unrealistically reduces the growth of all trees of a species to zero at the extremes of their field distribution.
2. Uniform distribution of species establishment throughout its field range. This formulation, combined with reduction of growth described in 1(b) above, implies that growth rate is the primary determinant of the distributions of species in the field, rather than ability to establish.
3. High mortality rates, whose effects on model behaviour are uncertain.
4. Response to fire modelled as only death of trees, not reduction in growth.
5. Sprouting that serves to replace most dead trees of many species, irrespective of cause of death.

In spite of these problems with BRIND, it can make plausible predictions of forest dynamics, species combinations and forest stocking in the Brindabella Range (Shugart and Noble 1981). These results can be interpreted in terms of detailed model structure.

Table 2.3: Values of $H_{\max}$ and G modified by T(DEGD) at 1600 degree-days for each species in BRIND. Values calculated from data given by Shugart and Noble (1981).

Species	Hmax (cm)	$G \times$ T(DEGD)
Eucalyptus delegatensis	6096	233
Eucalyptus fastigata	4572	0
Eucalyptus robertsonii	3962	71
Eucalyptus dalrympleana	3658	74
Eucalyptus viminalis	3658	0
Eucalyptus rubida	3048	5
Eucalyptus dives	2438	11
Acacia melanoxylon	2134	92
Eucalyptus pauciflora (H)	1829	112
Eucalyptus pauciflora (L)	1829	0
Eucalyptus stellulata	1524	48
Acacia pycnantha	1219	353
Acacia dealbata	1219	320
Acacia implexa	1219	207
Acacia falciformis	1067	325
Olearia argophylla	914	235
Bedfordia salicina	914	156
Banskia marginata	914	154
Exocarpos cupressiformis	762	97

BRIND was able to predict dynamic changes in composition of stands in the Brindabella Range at approximately 1300 m altitude (DEGD = 1600; Shugart and Noble 1981, Fig. 3). In the absence of fire, *E. delegatensis* dominates the plot because it is the tallest species and has the fastest growth rate of the taller species (Table 2.3). Mixtures of *E. dalrympleana* and *E. pauciflora* are maintained because each has a competitive advantage under different conditions (greater height for the former species and faster growth for the latter; also the greater $H_{\max}$ of *E. dalrympleana* means that the differential in height growth is less than that in G). During the course of a simulation, differing conditions will result in each of the species gaining an advantage. Shrub species in BRIND are all comparatively small and short-lived (Shugart and Noble 1981): they can grow quickly in a gap but are eventually overtopped by the taller eucalypt species.

An apparent anomaly in the results presented by Shugart and Noble (1981) for these simulations is the absence of *E. robertsonii*[2] in significant quantities. From its parameters (Table 2.3) it is expected to occur as commonly as *E. dalrympleana* because it grows nearly as fast and to a greater height. Simulations by BRIND described in Chapter 3 show *E. robertsonii* (*E. radiata*) to be as common as *E. dalrympleana* at a similar value of DEGD (Fig. 3.5).

BRIND predicted plausible combinations of dominant species over a range of DEGD at four different fire probabilities (Shugart and Noble 1981, Fig. 4). In all cases, the most common dominant species or codominant species are those with highest values of G modified by T(DEGD) and highest values of $H_{\max}$, affected further by fire response (with the exception noted above that *E. robertsonii* is expected to be as common as *E. dalrympleana* at DEGD of 1600, but was not predicted so by BRIND).

The effect of fire is to reduce significantly the occurrence (averaged over many years of simulation) of the fire-sensitive species *E. delegatensis* and to increase occurrence of species that require fire for germination (e.g., *Banksia marginata* and *Acacia dealbata*; Shugart and Noble 1981). In BRIND, all trees of *E. delegatensis* are killed in a fire and if the inter-fire period less than the maturation time of the species it will be lost from the simulated plot.

BRIND also predicted basal area, mean DBH and stocking rates of *E. delegatensis* after fire that were close to those reported in an independent data set (Shugart and Noble 1981, Fig. 5). These results indicate that the value of G

---

[2] *Eucalyptus robertsonii* referred to by Shugart and Noble (1981) is *E. radiata* Sieber ex DC. subsp. *robertsonii* (Blakely) L. Johnson & Blaxell and in the remaining chapters of this thesis is referred to as *E. radiata*.

for this species, modified by T(DEGD) in simulations, enables BRIND to predict plausible DBH values and thinning predicted by BRIND for the stand occurs at a plausible rate.

The main error in prediction by BRIND in this test was of initial stocking rate: *c.* 2500 stems/ha predicted, *c.* 5600 stems/ha in the field data (Shugart and Noble 1981, p. 161). From its parameters, BRIND is able to establish a maximum of 200 new trees in the year following a fire into its 1/12 ha plot (thus 2400 stems/ha). This limitation suggests that initial input of seedlings following fire could be much higher in BRIND.

I believe that BRIND is, in most cases, making only qualitatively correct predictions. The tests of BRIND described by Shugart and Noble (1981) are primarily qualitative in nature: prediction of species associations and species replacement with and without fire. The close match between BRIND's predictions of growth and stocking rate and independent data suggest that G modified by T(DEGD) for this species has an appropriate ecological value for those data: given the doubts about the validity of methods for calculating G discussed earlier in the chapter, this may have been just good fortune!

BRIND's prediction of the outcome of competition is essentially qualitative: the biggest trees do best. For a tree to become the biggest on the plot, it must have the fastest rate of height growth or a significant head start over faster-growing species. The fastest-growing species at each site will dominate most of the time, except when a sequence of events allows slower-growing species to gain a head start.

Fire affects the demographic behaviour of BRIND, but here again response is essentially qualitative: all trees of sensitive species are killed in most fires, nearly all those of resistant species survive and many more sites are made available for establishment of suitable species.

Chapter 3

Comparisons with field data

This chapter describes quantitative comparisons of predictions by BRIND with field data from the Brindabella Range. Comparisons were made of total plot biomass, numbers of trees (total on plot and size-class distributions) and species dominance (expressed as contribution to biomass). Further comparisons were then made using a version of BRIND without the SPROUT subroutine (Section 2.6). Simulations were then performed to investigate sensitivity of biomass prediction by BRIND to changes in the maximum biomass parameter SOILQ and the light extinction coefficient *k* used in modelling competition.

3.1 Field survey

Field data from eucalypt stands in the Brindabella Range were collected to compare with model predictions. The aim of the survey was to cover the range of altitudes and aspects and to sample sites on soils from the two common geological parent materials, sediments and granites, while minimising variation from other sources (such as topographic position and disturbance) as much as possible.

Table 3.1: Numbers of field plots in six 200 m altitude classes

Class	Altitude (m)	Number of plots
6	> 1500	14
5	1301–1500	20
4	1101–1300	20
3	901–1100	20
2	701–900	20
1	≤ 700	12

Slopes were chosen from 1:25 000 scale topographic maps on all aspects covering an alititudinal range of 660–1670 m within the western Cotter River catchment. A listing of site information for the 106 sample plots is given in Appendix A and numbers of plots in six 200 m altitude classes are shown in Table 3.1. While this range of altitude is greater than that for which BRIND was designed (Shugart and Noble 1981), the data were also to be used for extending the model. Sample plots were located in mid-slope positions (i.e., no gullies or ridges) at a random distance from the nearest road and avoiding sites showing evidence of disturbance.

Historically, major disturbances of the study area have been from fire. Time since last fire for each of the sites was estimated from a map showing the extents of major fires during this century in the Brindabella Range (ANU Forestry 1973). The map provides only general information about the boundaries of the fires but it indicates that most of the sample sites were last burnt by wildfire between 1926 and 1944 (40–60 years, plot data collected in 1984 and 1985).

At each site the altitude, aspect and angle of slope were recorded and a sample plot 50 m by 20 m (0.1 ha) marked out running across the slope. This plot size is similar to that of the plots simulated by BRIND and is a size commonly used in forest survey (Husch *et al.* 1982). In each plot, all trees taller than 'breast height' (at which DBH is measured; 137 cm in BRIND[1]) were recorded. The species of each tree was recorded and for trees with DBH less than 30 cm, size was estimated in 5 cm DBH classes; diameters of larger trees were measured with a girth tape. BRIND produces lists of trees and their DBH that are directly comparable with the data from the field plots.

3.2 Simulation methods

Simulations were performed using BRIND to predict plots that could be compared with those collected in the field. BRIND could not be set up to replicate the field plots exactly because details are not known of their past composition and fire history before the last major fire. The stochastic behaviour of many components in the model (establishment, mortality, climate, fire occurrence) also make it impossible to specify to BRIND the exact sequence of events that occurred in the past (Chapter 2).

The model was run 50 times at sites representing six altitudes to produce a distribution of predicted sites with which the field sites could be compared. All simulations in each set of 50 were run with the same initial conditions and parameter values but with different seeds to start the pseudo-random number generator. For analysis, the predictions were treated as statistical replicates, although there is some doubt as to the validity of using such Monte Carlo simulation results in this manner (R. B. Cunningham, personal communication).

Species and parameters used in the model were those described by Shugart and Noble (1981) but with the corrected value for the fire response parameter

---

[1] DBH is measured at 4.5 feet in the U. S. A. and at 1.3 m in countries that use metric measurements (Husch *et al.* 1982). The value of 137 cm used by BRIND is a conversion to metric of 4.5 feet. For this field survey, 137 cm was used to allow comparison with model predictions.

FB of 4.0 for fire-sensitive species (Section 2.5). Six values of mean DEGD were chosen to approximate the six altitude classes of the field data. These values correspond to class midpoints and altitudes of 660 and 1600 m for the end classes. The parameter for maximum stand biomass (SOILQ) was set to 1000 t/ha, as used by Shugart and Noble (1981).

Because the past fire history of the sites is not known in detail, simulations were performed covering a range of frequencies of occurrence. The model was run with an annual probability of wildfire of 0.0, 0.005, 0.01 and 0.02 (expected return times of never, 200, 100 and 50 years, respectively) in each DEGD (altitude) class.

Because the past composition of the field sites was unknown, each simulation was started from a plot with four small individuals of each species and run for a lead time of 500 years. Starting plots with trees present were used rather than bare plots (as with many gap models, see Shugart 1984) because selection of species for establishment in BRIND is weighted by the species present (Chapter 2). The 500-year lead time was used to allow each simulation to 'settle' according to environmental conditions and was long enough to ensure that trees present when the plot was sampled had established during the simulation and were not from the starting plot. All processes (establishment, growth, mortality, fire, sprouting) described in Chapter 2 were simulated during the lead time.

After expiration of the lead time, each simulation was sampled three times and the model stopped. Three samples were taken 40, 50 and 60 years after the first fire which was followed by a 60-year fire-free period. Simulations without fire were sampled 40, 50 and 60 years after the lead time had elapsed.

Three samples of each simulation were taken to assess the effect of time since fire on model predictions. Analyses of variance in plot biomass and numbers of trees were performed at each combination of DEGD and fire probability. In all cases, no difference was found between sampling times ($P > 0.1$), so only the

50 years post-fire plots were used for comparison with the field data.

3.3 Total biomass

Total biomass on each field plot was calculated by summing estimated biomasses of all trees using the relationship from BRIND (Eqn 2.12), adjusted for the difference in plot size (0.1 vs. 0.083 ha). A generalised linear regression model (hereafter GLM) with normal distributions of errors (Nelder and Wedderburn 1972; McCullagh and Nelder 1983) was fitted to the field biomass values to examine effects of altitude (six 200 m classes), aspect (four 90° classes) and geology type (two classes: sediments and granites). Biomass is only affected by altitude ($P < 0.001$), with all other terms non-significant ($P > 0.1$). The GLM predicts highest biomass at intermediate altitudes and less at both high and low altitudes.

There were two sites which were clear outliers with very high calculated biomass. These two sites are both in the 1101–1300 m altitude class and included a number of very large trees with DBH greater than 100 cm. These biomass estimates may not be accurate because the equation used to calculate them was derived from trees of less than 50 cm DBH (Ashton 1976) and is sensitive to large values of DBH. However the same equation was used to calculate plot biomass predicted by BRIND, so if the model predicts similar size structure it will predict similar biomass values.

Regression analysis like that described above for the field plots was carried out on the simulated plot data (terms in the GLM were DEGD class and fire frequency). There was significant variation in predicted biomass with DEGD (altitude) class ($P < 0.001$) but no difference with fire frequency ($P > 0.1$). The data for each DEGD class were combined to produce distributions of 200 values for comparison with the combined field data from the corresponding altitude class.

The comparisons are presented visually in Fig. 3.1, where the histograms show the distributions of predicted biomass and the symbols along the biomass axes show calculated biomass of the field sites, in each of the six altitude classes. Clearly, BRIND does not predict the variation in total biomass with altitude found in the field plots. The model predicts reasonably consistent average biomass across the range of altitude in spite of changes in dominant species (Section 3.5).

It is possible that the lack of variation in predicted biomass with altitude, in spite of variation in species, is because of the constant value of 1000 t/ha for SOILQ used in the stand crowding index (S(BAR), Eqn 2.11). Because this index restricts the growth of all trees on the plot, it is expected to be important in determining plot biomass. The effect of the S(BAR) index on predicted biomass is examined in Section 3.7.1.

3.4 Numbers of trees

Total numbers of trees predicted by the model were compared with those in the field plots, using the same method as that used for comparing plot biomass, adjusted for the different plot size. Examination with regression (as for biomass) showed that numbers of trees in the field plots varied only with altitude class ($P < 0.001$) and not with aspect nor geology class ($P > 0.1$).

Regression analysis of numbers of trees on the simulated plots showed that predicted numbers of trees varied with both DEGD and fire probability and that a DEGD–fire interaction term was also significant ($P < 0.001$ in all cases). At all DEGD values, the effect of fire was greatest at the highest probability of 0.02. The field data are shown compared to model predictions without fire in Fig. 3.2 and with predictions resulting from a fire probability of 0.02 in Fig. 3.3.

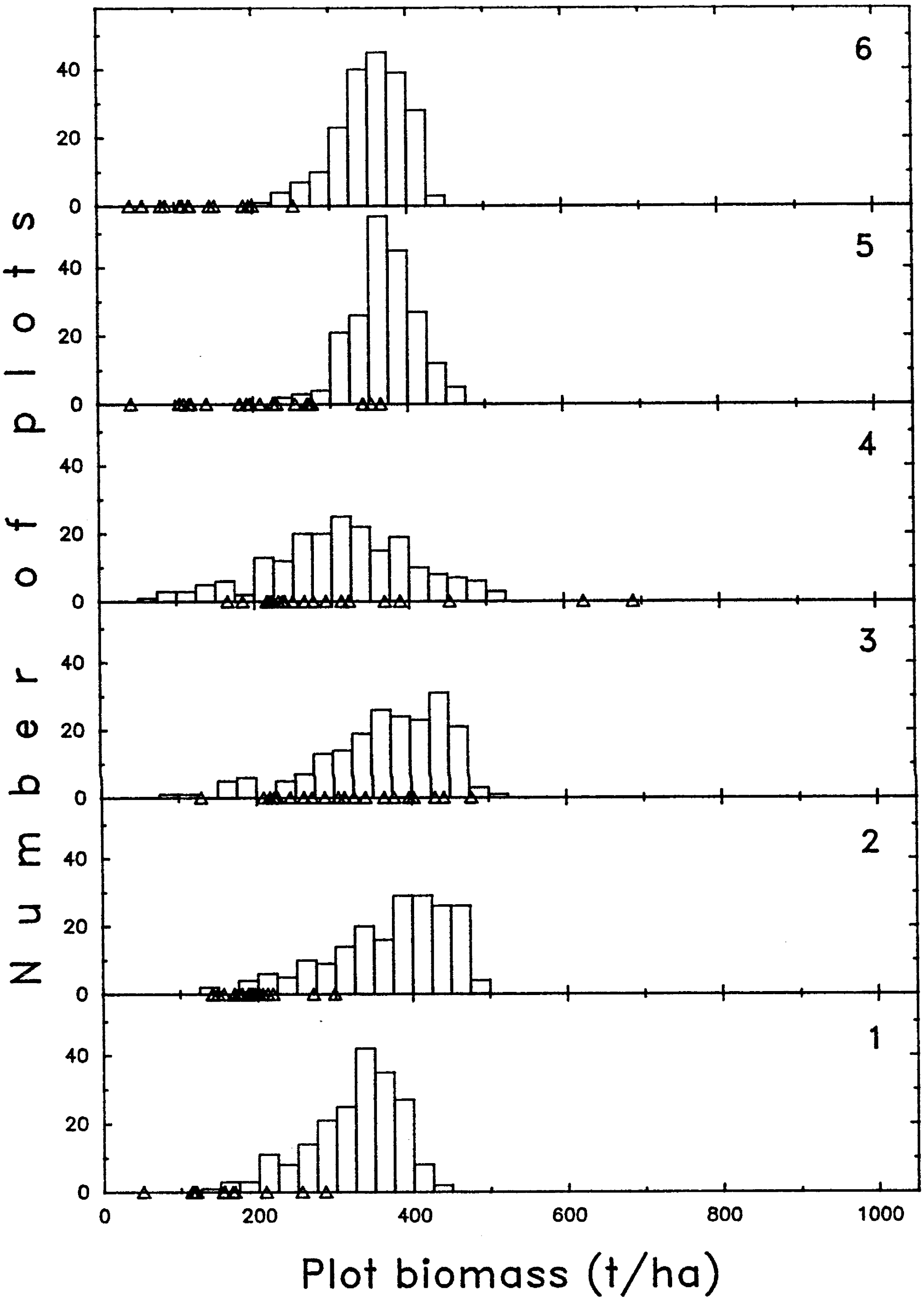

Figure 3.1: Distributions of predicted plot biomass from 200 simulations by BRIND (histograms) and estimates calculated from field plot data (symbols) in each of the six 200 m altitude classes shown in Table 3.1.

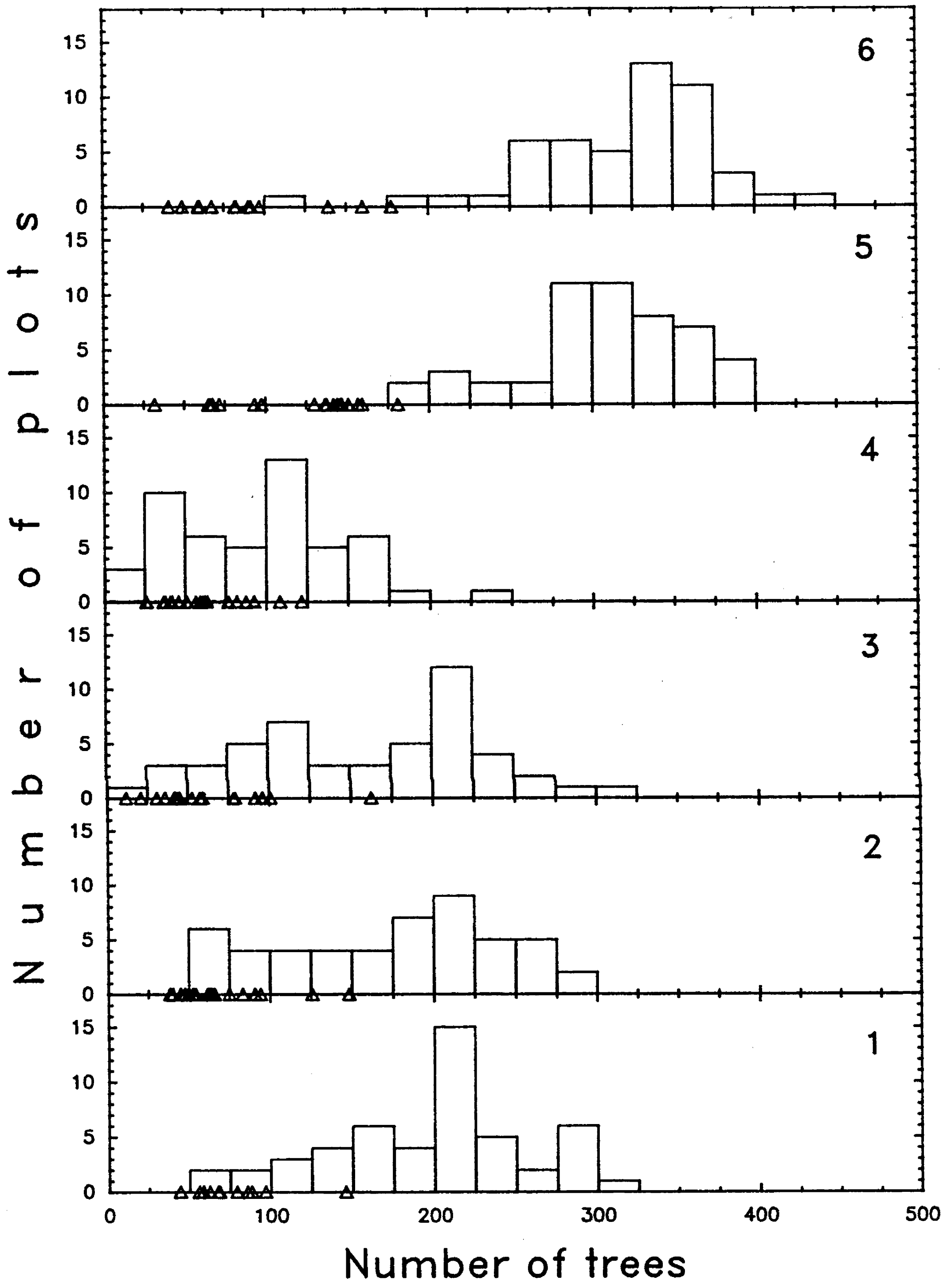

Figure 3.2: Distributions of predicted numbers of trees from 50 simulations by BRIND without fire (histograms) and data from field plots (symbols) in each of the six 200 m altitude classes shown in Table 3.1.

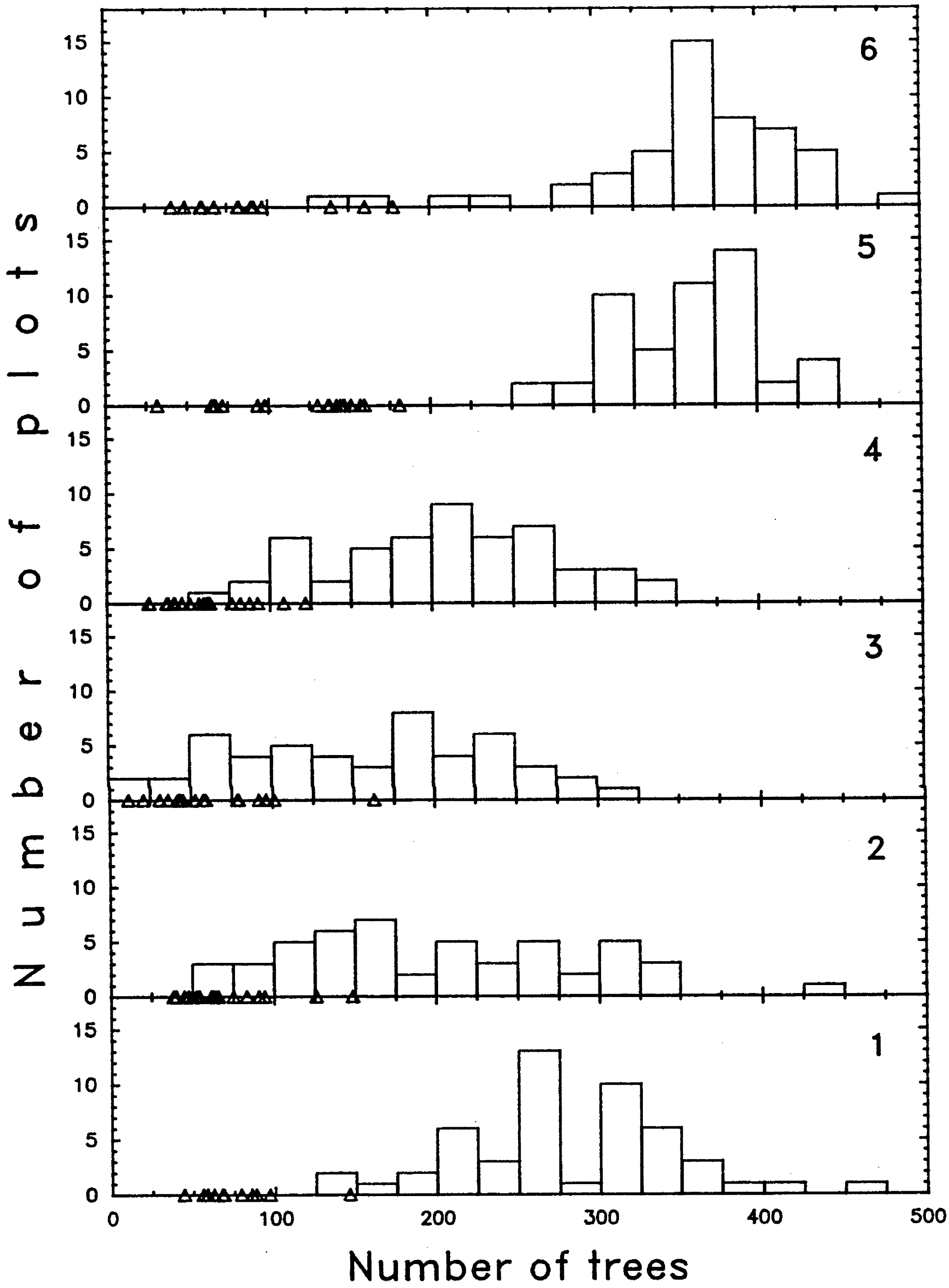

Figure 3.3: Distributions of predicted numbers of trees from 50 simulations by BRIND with annual probability of fire of 0.02 (histograms) and data from field plots (symbols) in each of the six 200 m altitude classes shown in Table 3.1.

BRIND mostly predicts too many trees, except in altitude class 4 (1101–1300 m) in the absence of fire. It is unlikely that all of the field sites in that altitude range have escaped fire during the past 60 years (the available fire history data only indicate which general areas were burnt in each fire). At all altitudes more trees are predicted after fire (Fig. 3.3, cf. Fig. 3.2).

More detailed comparisons were made between actual and predicted numbers of trees in 10 cm DBH classes at selected altitudes. Regression analysis like those used for total biomass and numbers was performed on each of the DBH classes but detected no effect of aspect and geology on numbers in each ($P > 0.1$). Comparisons were made for altitude classes 1, 4 and 6: in class 1, BRIND predicted excessive biomass and too many trees; in class 4 it predicted plausible biomass and numbers without fire but too many trees with fire; and in class 6 it predicted too much biomass and too many trees by a factor of 3 to 4.

Results of the comparisons are shown in Fig. 3.4. For each 10 cm DBH class, the mean and range of values are shown for the field sites (green), simulated sites without fire (black) and simulated sites with a fire probability of 0.02 (red). There are two vertical scales on each graph: 0–200 for size classes of less than 40 cm DBH and 0–20 for larger ones (separated by the dashed line). The largest size class contains all trees with DBH greater than 100 cm. If the model is predicting correct size-class distributions of the trees, the means and ranges of predicted counts in each class should be similar to those of counts recorded in the field.

BRIND predicts too many trees with DBH less than 10 cm, particularly at the extreme altitudes. In altitude class 6 (Fig. 3.4(a)), the model never predicted fewer than 140 trees of less than 10 cm DBH, where at least one field site had fewer than 10. The extent of overprediction of numbers of small trees was less in altitude class 1 (Fig. 3.4(c)) and still less in class 4 (Fig. 3.4(b)). At all altitudes, the model predicted more small trees in the presence of fire than without. Larger

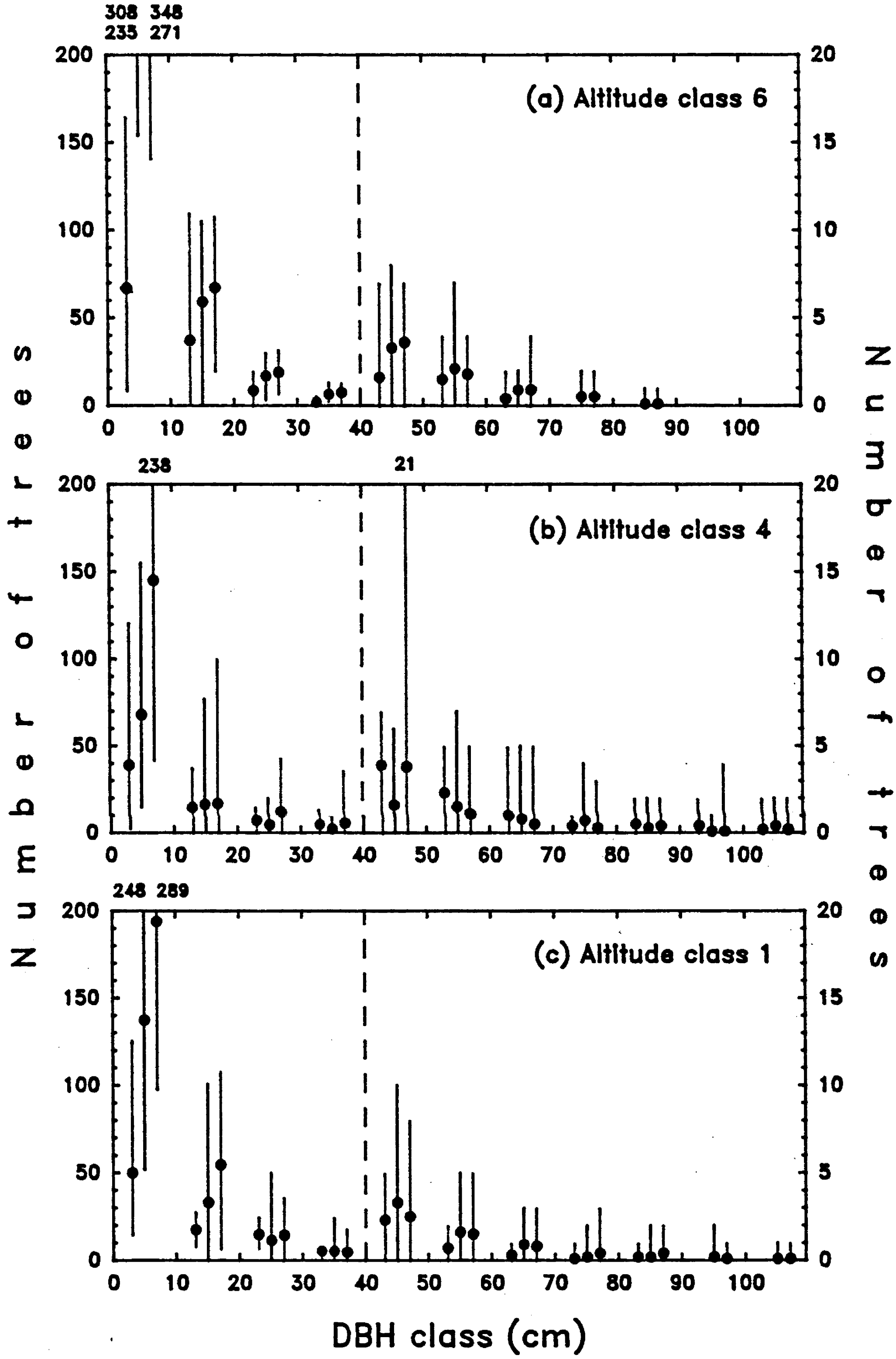

Figure 3.4: Mean values and ranges of numbers of trees in 10 cm DBH classes from field plots (green), 50 simulations by BRIND without fire (black) and 50 simulations with annual probability of fire of 0.02 (red), in three altitude classes. Each graph has two vertical scales: 0–200 trees to the left of the dashed line; and 0–20 to the right of the line.

trees (DBH greater than 10 cm) were predicted in numbers similar to those found in the field plots.

At extreme altitudes, BRIND predicts the survival of trees to bigger sizes than those found in the field. In altitude class 6, the largest trees recorded in the field data had DBH less than 70 cm but the model predicted one or two trees with DBH up to 90 cm (Fig. 3.4(a)). Similarly in class 1, trees up to 90 cm DBH were found in the field, but the model predicted one or two individuals with DBH greater than 100 cm (Fig. 3.4(c)). These large trees predicted by the model contribute little to total numbers but greatly to plot biomass.

These results suggest that BRIND is predicting too much establishment overall, but this varies with altitude. In the three altitude classes shown, field and simulated plots are dominated by different species: exclusively *Eucalyptus pauciflora* in class 6, principally *E. delegatensis* in class 4, and *E. dives* and *E. radiata* in class 1 (see Section 3.5 for comparisons of species distributions between field and simulated plots). In the extreme altitude classes, BRIND predicts dominance of lignotuberous species, while *E. delegatensis* that dominates class 4 does not form lignotubers. In the model, lignotuberous species sprout prolifically (see description of the SPROUT subroutine in Section 2.6), which probably leads to the prediction of excessive numbers of small trees as shown here. *Eucalyptus delegatensis* is sensitive to frequent fire—if the period between two fires is less than the maturation time for this species it is lost from the simulated plot (Shugart and Noble 1981)—and is replaced in BRIND by lignotuberous species (such as *E. pauciflora* and *E. dalrympleana*) after repeated fires, a factor which may result in excessive numbers of trees.

Some observations about mortality in BRIND can be made from Fig. 3.4. Even when numbers of small trees (DBH $<$ 10 cm) predicted by the model were five times too high, numbers in the next size class were too high by less than 50% in mean value and have plausible ranges (Fig. 3.4(a)). Mortality of small trees is thus very high and shows density-dependent behaviour expected from the structure of the one-sided competition model (Chapter 2). Prediction by BRIND of trees larger than those found in the field plots at extreme altitudes indicates that the combination of survivorship and growth rate for these trees is such that they either live too long or grow too fast.

3.5 Species distributions

A subset of the field data were compared with predictions by BRIND of species distributions along the gradient of DEGD. Because BRIND implicitly assumes that water is not limiting to tree growth and establishment (Chapter 2), it models forests on the sheltered slopes of the Brindabella Range. Of the field data, those sites with aspects from 90° to 180° are most appropriate for comparison with predictions by BRIND. There are a limited number of these plots (in altitude classes 1–6: 2, 6, 5, 5, 5 and 3 plots, respectively) but comparison with model predictions provides insights into model behaviour.

Figure 3.5 shows the field data and predicted species abundances for the eight most abundant eucalypt species predicted by BRIND. Each graph shows comparisons for the six altitude classes. In each class are shown the individual values of percentage biomass for the species in each of the field sites joined by a green line; and the minimum, mean and maximum values predicted by BRIND, both without fire (black) and with a fire probability of 0.02 (red). Where a species was not found or was not predicted to occur in an altitude class, no symbols are plotted. Zero values are plotted for a species when there were also non-zero abundances in that altitude class.

Two observations can be immediately made from Fig. 3.5: there are differences

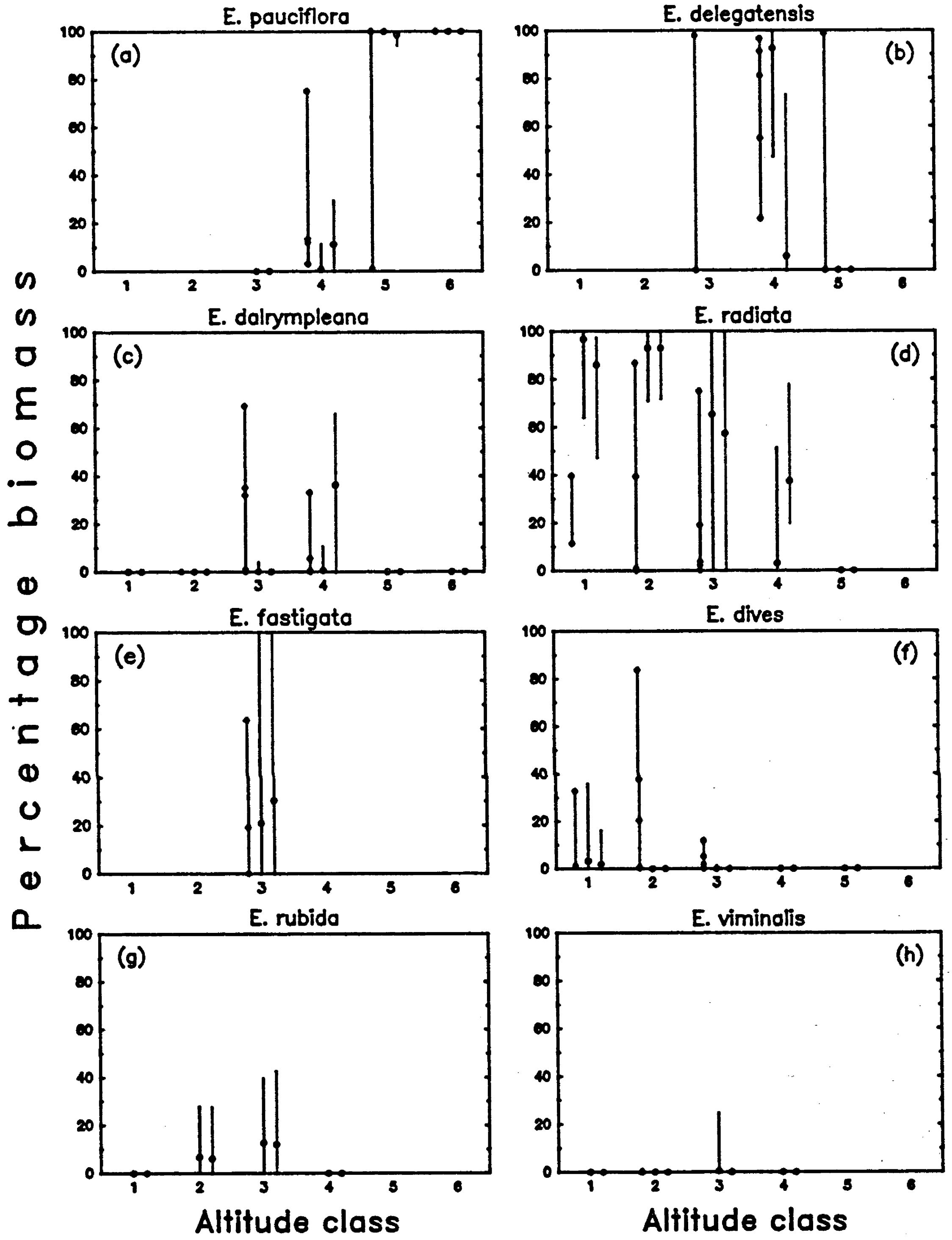

Figure 3.5: Relative species abundances (biomass as percentage of plot total) of the eight most common species simulated by BRIND and field data from plots on aspects from 90° to 180°. Individual field plot values are shown, connected by a green line. Data predicted by BRIND are shown as mean values and ranges from 50 simulations without fire (black) and with annual probability of fire of 0.02 (red). Numbers of field plots in altitude classes 1–6 were 2, 6, 5, 5, 5 and 3, respectively.

in range and magnitude of species distributions and there is often less variance in predicted species abundance than found in the field data. If BRIND's predictions are accurate, the field data should fall mainly within the ranges of altitude and abundance shown for the model.

For some species, predicted altitude ranges are greater than in the field data: *E. dalrympleana* (Fig. 3.5(c)), *E. radiata* (Fig. 3.5(d)), *E. dives* (Fig. 3.5(f)), *E. rubida* (Fig. 3.5(g)), *E. viminalis* (Fig. 3.5(h)) and to lesser extent *E. pauciflora* (Fig. 3.5(a)). The predicted altitude range for *E. fastigata* matched that in the field data (Fig. 3.5(e)) but BRIND predicted a narrower range for *E. delegatensis* than was found in the field (Fig. 3.5(b)).

Differences between predicted and field species distributions can be reduced by making small adjustments to species parameters in BRIND that control them: $\mathrm{DEGD_{min}}$, $\mathrm{DEGD_{max}}$ and G (Chapter 2). However *E. rubida* was not found in any field plots on sheltered slopes and *E. viminalis* was found in only one. *Eucalyptus rubida* is widespread but uncommon (ANU Forestry 1973) and is more common in the neighbouring Tinderry Range to the south-east (personal observations), that has lower rainfall (annual rainfall *c.* 600–700 mm vs. *c.* 1000–1100 mm for the Cotter catchment; Adomeit *et al.* 1984). *Eucalyptus viminalis* is most commonly found on lower slopes and gullies in the Brindabella Range (ANU Forestry 1973). The distributions of these species are affected by environmental differences that cannot be adequately described by their response to temperature alone.

BRIND predicts lower variance in species abundance for some species than found in the field data. Lower variance is particularly evident for *E. pauciflora* and also for *E. radiata* in altitude classes 1 and 2, where it is predicted with high abundance. If the model were making accurate predictions, variance in species biomass in 50 simulations would be greater than that in a limited number of field sites.

Lower variance in species abundances predicted by BRIND is a consequence of the emphasis in the model on growth as the main controller of species distributions. The result of competitive interactions always favours taller species with higher values of G (as adjusted by T(DEGD) at a particular DEGD value; Chapter 2), except where a slower-growing species can gain a 'head start' from random events of establishment and mortality. BRIND predicts almost complete dominance by a species whose value of $G \times$T(DEGD) is much greater than that of other species at that altitude, for example *E. pauciflora* in altitude class 5 (Fig. 3.5(a)) and *E. radiata* in classes 1 and 2 (Fig. 3.5(d)). Even where a tree of a species near the end of its DEGD range can establish, its growth is limited so much by the T(DEGD) index that it may be suppressed and killed by BRIND, so variance introduced into the model by stochastic behaviour of the establishment and mortality submodels is reduced by the deterministic nature of the growth submodel.

Species with lower values of G are predicted by BRIND to be very uncommon (but not absent) throughout their DEGD range, except in a small range near their DEGD optima. This model behaviour produces species distributions with 'tails' (e.g., *E. dalrympleana* and *E. dives*).

The effect of fire on model predictions is seen mainly in altitude class 4, where *E. delegatensis* is reduced in abundance when fires are more frequent and *E. dalrympleana* and *E. radiata* show increased abundance.

Eucalyptus mannifera and *E. macrorhyncha* were found in the field plots at lower altitudes but are not in the BRIND species set. *Eucalyptus mannifera* is found on all aspects in the Brindabella Range but more commonly on exposed sites, whereas *E. macrorhyncha* is found typically on east and south-east slopes at low altitudes (ANU Forestry 1973).

3.6 Removal of the SPROUT subroutine

In Chapter 2 I argued that the SPROUT subroutine in BRIND is not a realistic model of the phenomenon of lignotuberous sprouting in eucalypts and results in continuous re-establishment of sprouting species after deaths of any trees. It appears likely that removal of this routine will improve prediction of biomass and tree numbers by BRIND. To investigate this, sets of 50 simulations were performed in each of the six altitude classes as before (Sections 3.3 and 3.4) using a version of BRIND without SPROUT. In these simulations, sequences of annual DEGD, fire years and fire intensities were exactly the same as for the simulations described above.

While fire has no effect on biomass prediction by BRIND with SPROUT (Section 3.3), when SPROUT is removed the addition of fire (with annual probability of 0.02) results in a significant change in biomass ($P < 0.001$). Distributions of predicted biomass in 50 simulations at each altitude class without fire are shown in Fig. 3.6 and with fire in Fig. 3.7.

These figures can be compared with Fig. 3.1, from which it is apparent that removing SPROUT results in a marked improvement in prediction of plot biomass. There is variation in predicted biomass with altitude (lower biomass at extreme altitudes and higher biomass at intermediate altitudes) that is similar to that in field biomass (Fig. 3.6). In altitude classes 1, 2 and 6 the model without SPROUT still predicts too much biomass on average.

At all altitudes, the addition of fire results in lower predicted biomass (Fig. 3.7, cf. Fig. 3.6). Without SPROUT, fire-related mortality (Chapter 2) in BRIND affects plot biomass but its effects were masked by the artificially high establishment rates from SPROUT. Predicted biomass with fire is closer to field biomass in all altitude classes except for classes 3 and 4.

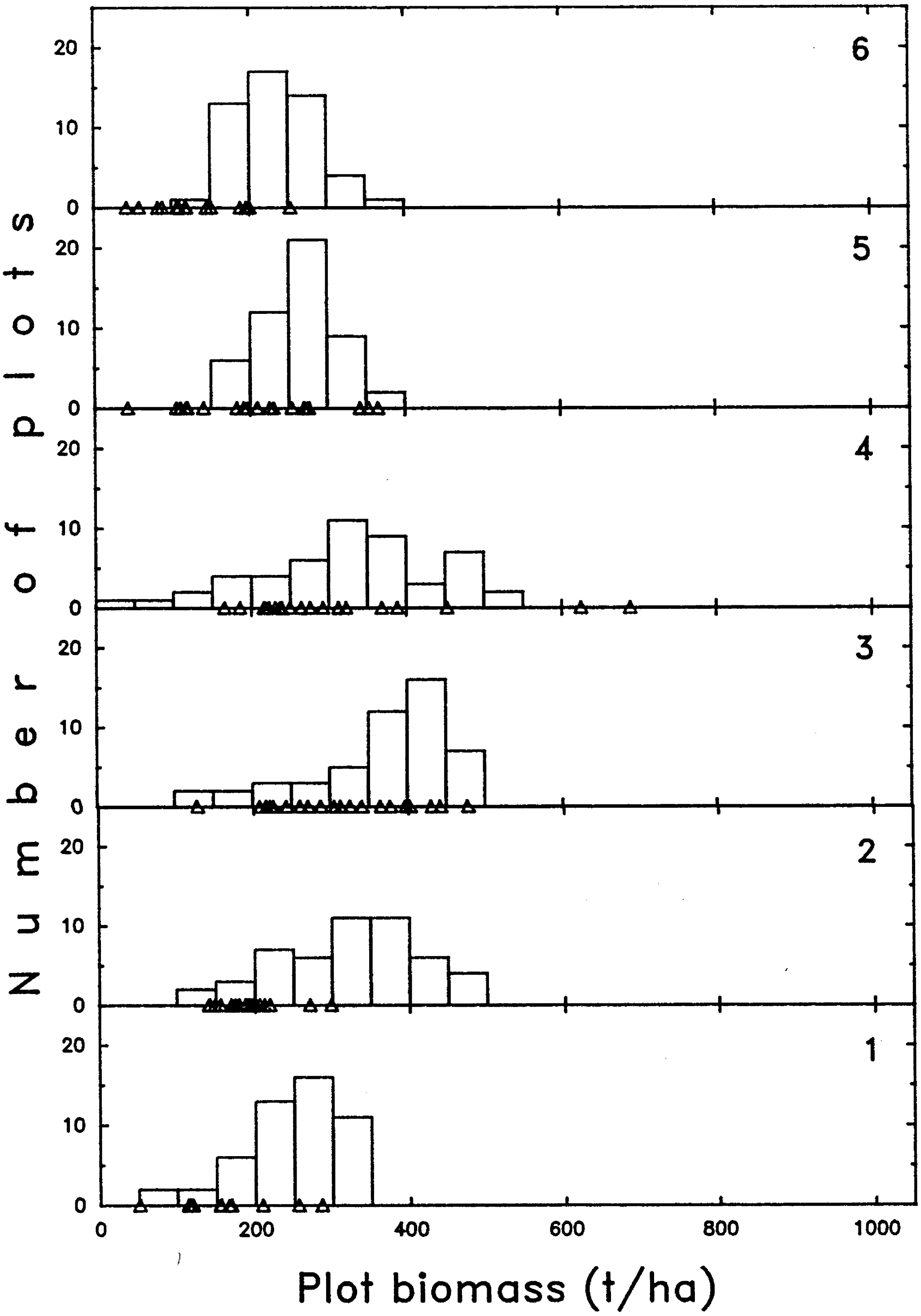

Figure 3.6: Distributions of predicted plot biomass from 50 simulations by BRIND without the SPROUT subroutine and without fire (histograms) and data from field plots (symbols) in each of the six 200 m altitude classes shown in Table 3.1. Compare with Figs 3.1 (model with SPROUT) and 3.7 (with fire).

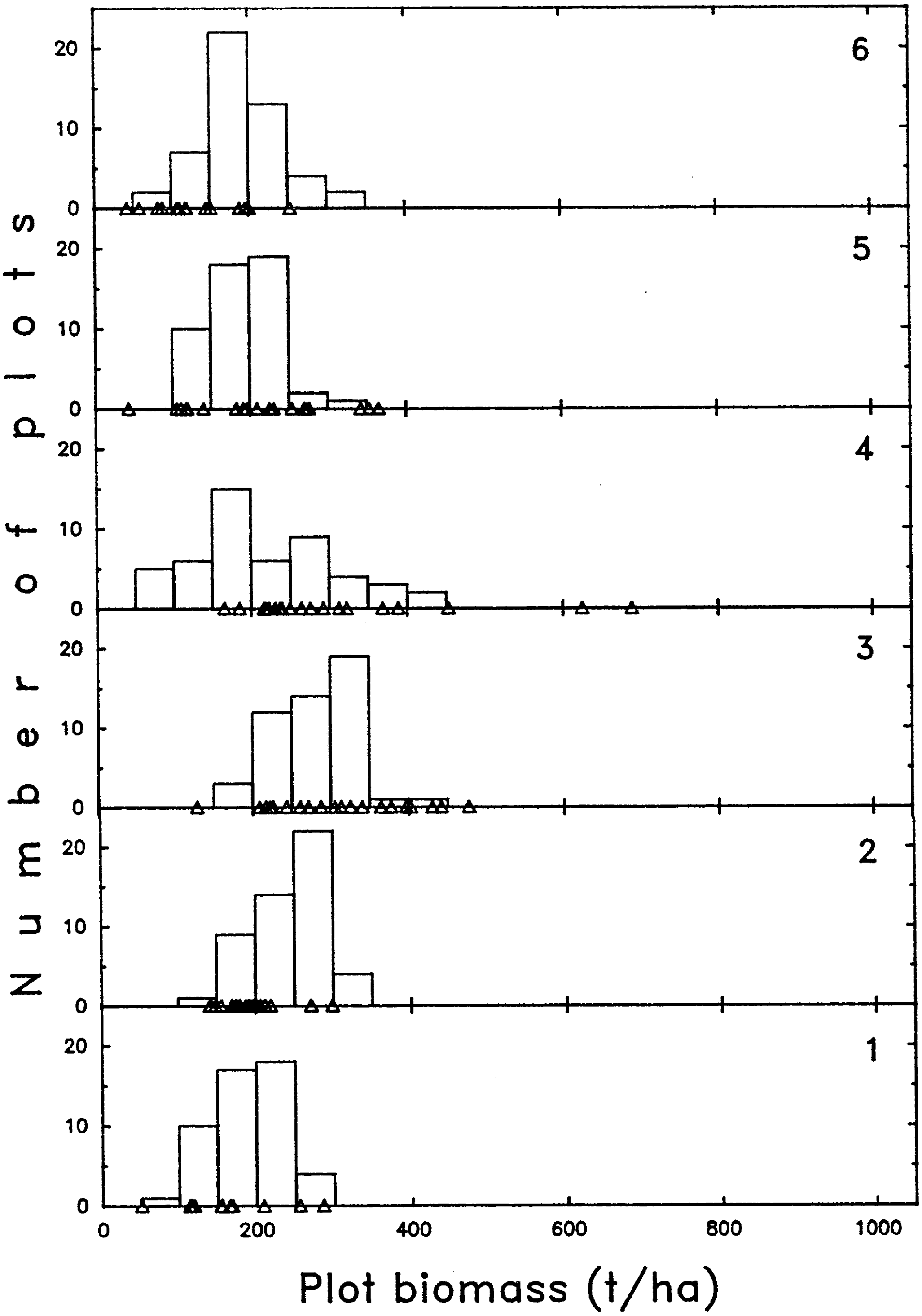

Figure 3.7: Distributions of predicted plot biomass from 50 simulations by BRIND without the SPROUT subroutine and with annual probability of fire of 0.02 (histograms), and data from field plots (symbols) in each of the six 200 m altitude classes shown in Table 3.1. Compare with Figs 3.1 (model with SPROUT) and 3.6 (without fire).

Removal of SPROUT resulted in a noticeable improvement in prediction of numbers of trees in most size classes. Figure 3.8 shows mean values and ranges of numbers of trees in 10 cm size classes predicted by the model without SPROUT at altitude classes 1, 4 and 6, with and without fire, and can be directly compared with Fig. 3.4.

As with BRIND including SPROUT, the predicted number of trees in many size classes was higher in the presence of fire, with the exception of the plots in altitude class 6 (dominated by *E. pauciflora*), where fire had a lesser effect. In altitude classes 1 and 6, BRIND without SPROUT still predicted the growth and survival of trees larger (by up to 20 cm DBH) than found in the field (Fig. 3.8(a) and (c)).

Figure 3.9 shows species abundances (as percentage biomass) for the eight common species predicted by BRIND without SPROUT and can be compared with Fig. 3.5. Patterns of species distributions were similar to those predicted by BRIND with SPROUT. There is some reduction in the dominance of faster-growing, sprouting species in some cases (e.g., *E. pauciflora* in altitude class 5 and *E. radiata* in classes 1, 2 and 3), with corresponding increases in abundance of slower-growing species (*E. dalrympleana* in class 5, and *E. dives* and *E. viminalis* in classes 1, 2 and 3). Removal of SPROUT also results in the removal of 'tails' from species distributions because suppressed trees near the extremes of their DEGD ranges are not replaced by sprouts.

The match between predicted and field distributions of species can be improved by adjustment to species parameters. From Fig. 3.9 it appears that the DEGD_{min} and DEGD_{max} parameters for *E. dalrympleana* should be adjusted for BRIND to make better predictions of the distribution of this species. Similarly, it appears that G for *E. radiata* should be reduced. These adjustments are appropriate, given the uncertainty about the parameter values (Chapter 2).

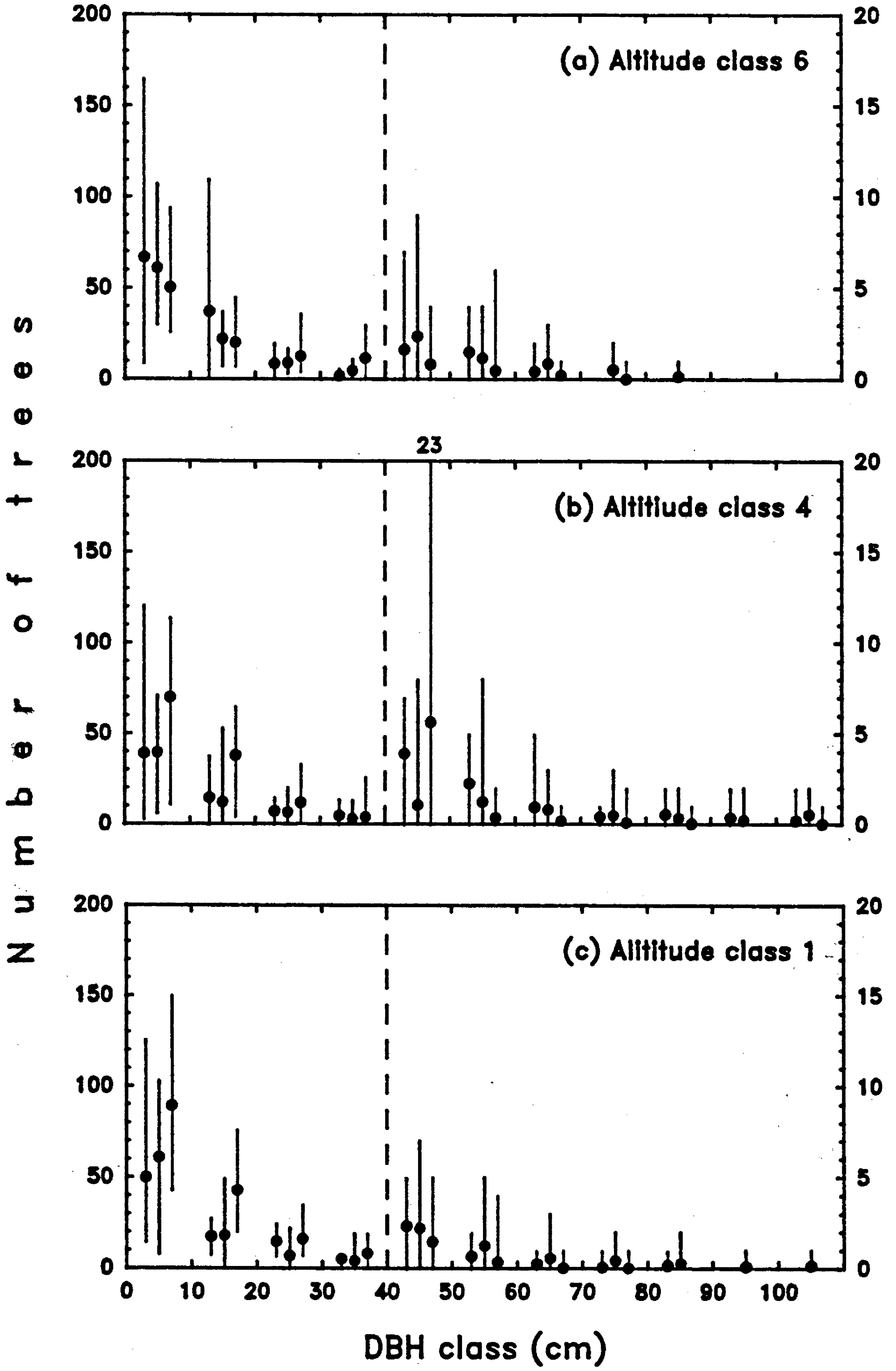

Figure 3.8: Mean values and ranges of numbers of trees in 10 cm DBH classes from field plots (green), 50 simulations by BRIND without the SPROUT subroutine without fire (black) and 50 simulations with annual probability of fire of 0.02, in three altitude classes. Each graph has two vertical scales: 0–200 trees to the left of the dashed line; and 0–20 to the right of the line. Compare with Fig. 3.4 (BRIND with SPROUT).

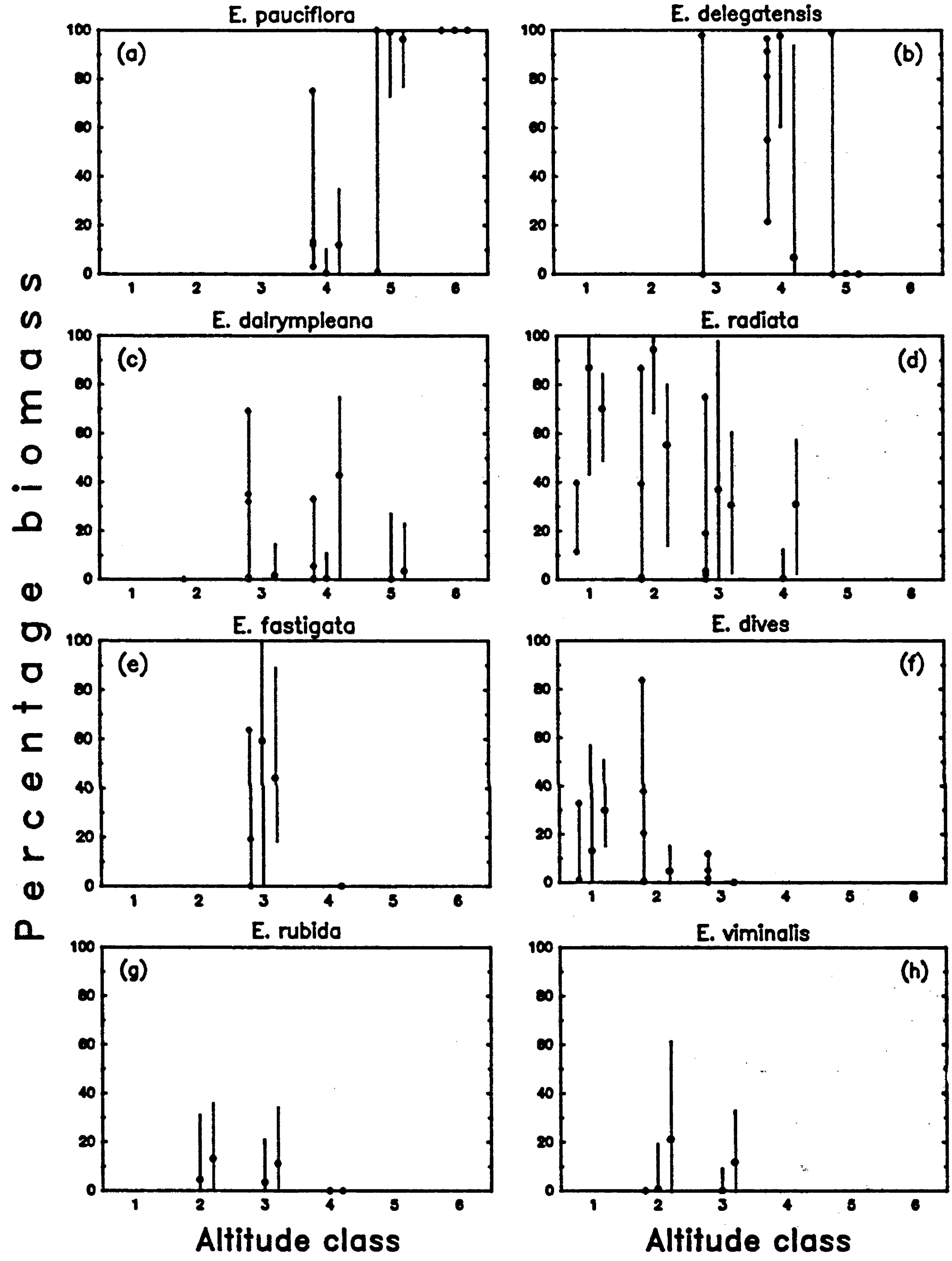

Figure 3.9: Relative species abundances (biomass as percentage of plot total) of the eight most common species simulated by BRIND without the SPROUT subroutine and field data from plots on aspects from 90° to 180°. Individual field plot values are shown, connected by a green line. Data predicted by BRIND are shown as mean values and ranges from 50 simulations without fire (black), and with annual probability of fire of 0.02 (red). Compare with Fig. 3.5 (BRIND with SPROUT).

3.7 Plot biomass

The model without SPROUT still predicts too much biomass in some cases. It is not clear how sensitive plot biomass is to variation in parameters in the growth model in BRIND, specifically the maximum biomass parameter SOILQ and the light extinction coefficient k. The former affects the growth of all trees on the plot through the S(BAR) index and the latter has been shown to greatly affect the outcome of competitive interactions (Noble *et al.* 1988).

3.7.1 Maximum biomass parameter (SOILQ)

The S(BAR) index reduces the growth of all trees on the simulated plot, reducing it severely when plot biomass approaches the maximum biomass value SOILQ (Chapter 2).

To investigate the effect of varying SOILQ on plot biomass, sets of 50 simulations were performed in altitude class 6 using BRIND without SPROUT for five values of SOILQ: 400 t/ha, 600, 800, 1000 (that used in the simulations described above) and ∞ (no S(BAR) index). Simulations were performed exactly as described in Section 3.2, without fire. In this altitude class the model without SPROUT still predicted nearly twice as much biomass on average as found in the field sites (234 t/ha predicted, 133 t/ha field).

Figure 3.10 shows distribution of predicted plot biomass for each value of SOILQ, with finite values of SOILQ indicated by dashed lines and mean predicted biomass in parentheses. A 60% reduction in SOILQ (1000 to 400 t/ha) resulted in only 24% reduction in mean biomass. Without the S(BAR) index, mean predicted plot biomass was only 32% higher than with SOILQ of 1000 t/ha. There are clearly other components of BRIND which limit accumulation of biomass when there is no restriction from S(BAR).

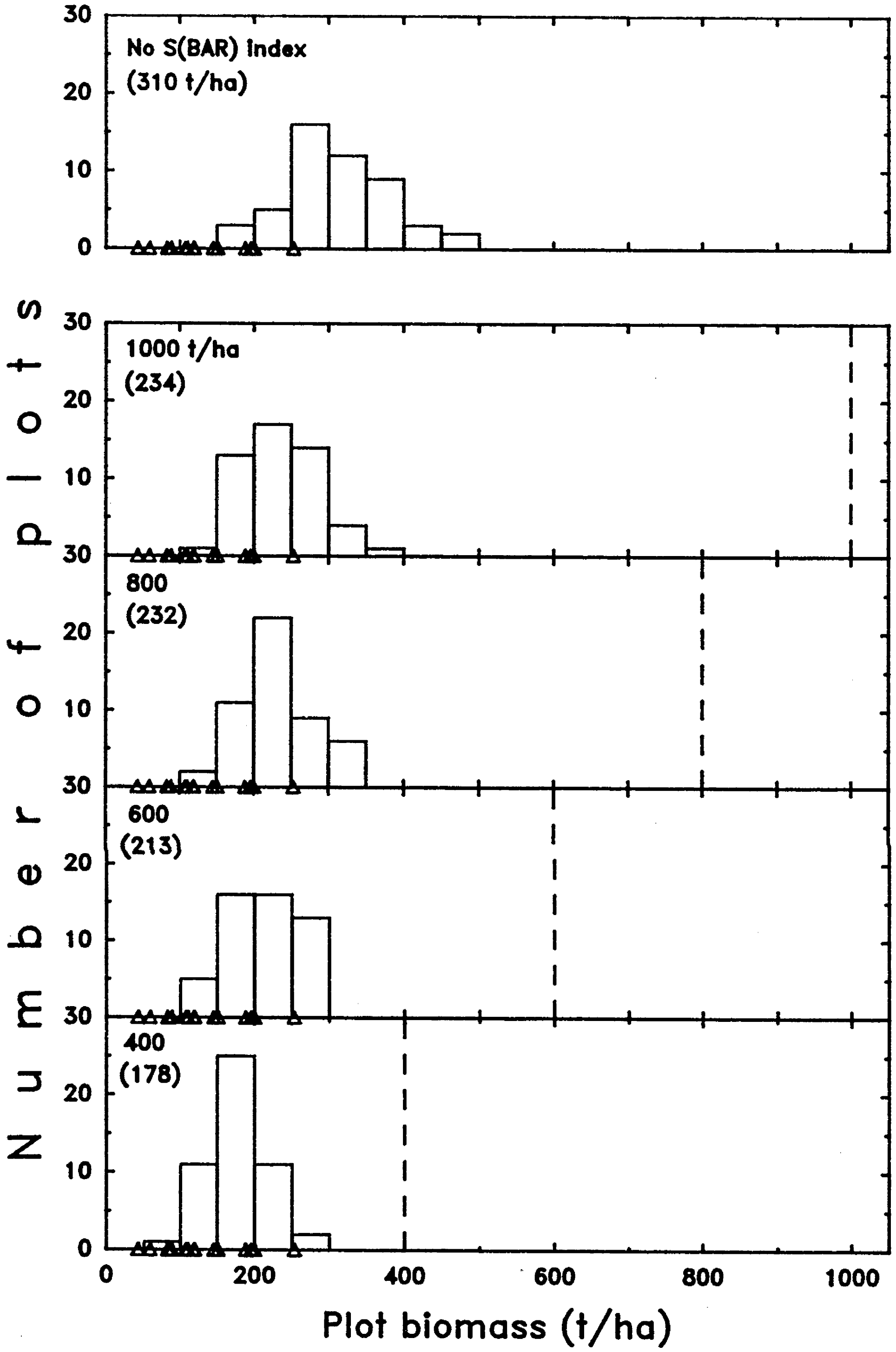

Figure 3.10: Distributions of predicted plot biomass from 50 replicate simulations by BRIND without SPROUT, using values for SOILQ of 400, 600, 800 and 1000 t/ha and without the S(BAR) index. Dashed lines show finite values of SOILQ on the appropriate graphs and symbols show biomasses of the 14 field plots in this altitude class. Simulations were run with mean DEGD of 1020 (altitude class 6) and without fire. Dominant species was *E. pauciflora*. Numbers in parentheses are mean predicted plot biomass values.

Detailed examination of stands with and without the S(BAR) index showed that, while the growth of all trees was reduced by decreasing SOILQ, only the smallest trees have their growth reduced enough to be killed in significant numbers by the suppression mortality component of BRIND (Section 2.4). Small trees contribute comparatively little to the total biomass of the stand. With low SOILQ the model was still predicting the occurrence of trees with DBH larger than any found in the field.

3.7.2 Light extinction coefficient (k)

The light extinction coefficient k in the shading equation r(AL) expresses the extent of growth reduction from shading and hence intensity of competition between trees on the simulated plot. Noble *et al.* (1988) showed that this index is more sensitive to the value of k used than to changes in competition model structure. It is likely that predicted biomass is also sensitive to k as a result.

To investigate the effect on plot biomass of varying k, sets of 50 simulations were performed in altitude classes 4 and 6 using BRIND without SPROUT and without the S(BAR) index for four values of k: 0.3, 0.5 (the value in BRIND), 0.7 and 0.9. Simulations were performed without fire. In altitude class 4 the plots are dominated by a large, fast-growing species (*E. delegatensis*); in class 6 they are dominated by a smaller, slower-growing species (*E. pauciflora*).

Figure 3.11 shows for altitude class 6 the distribution of predicted plot biomass for each value of k and Fig. 3.12 shows the same for altitude class 4. Mean values of predicted biomass from each set of simulations are shown in parentheses. It is apparent that plot biomass is sensitive to variation in k and exhibits a curved response to it (lower values of k result in much higher biomass). If k were zero, there would be no restriction on growth from the presence of trees on the plot

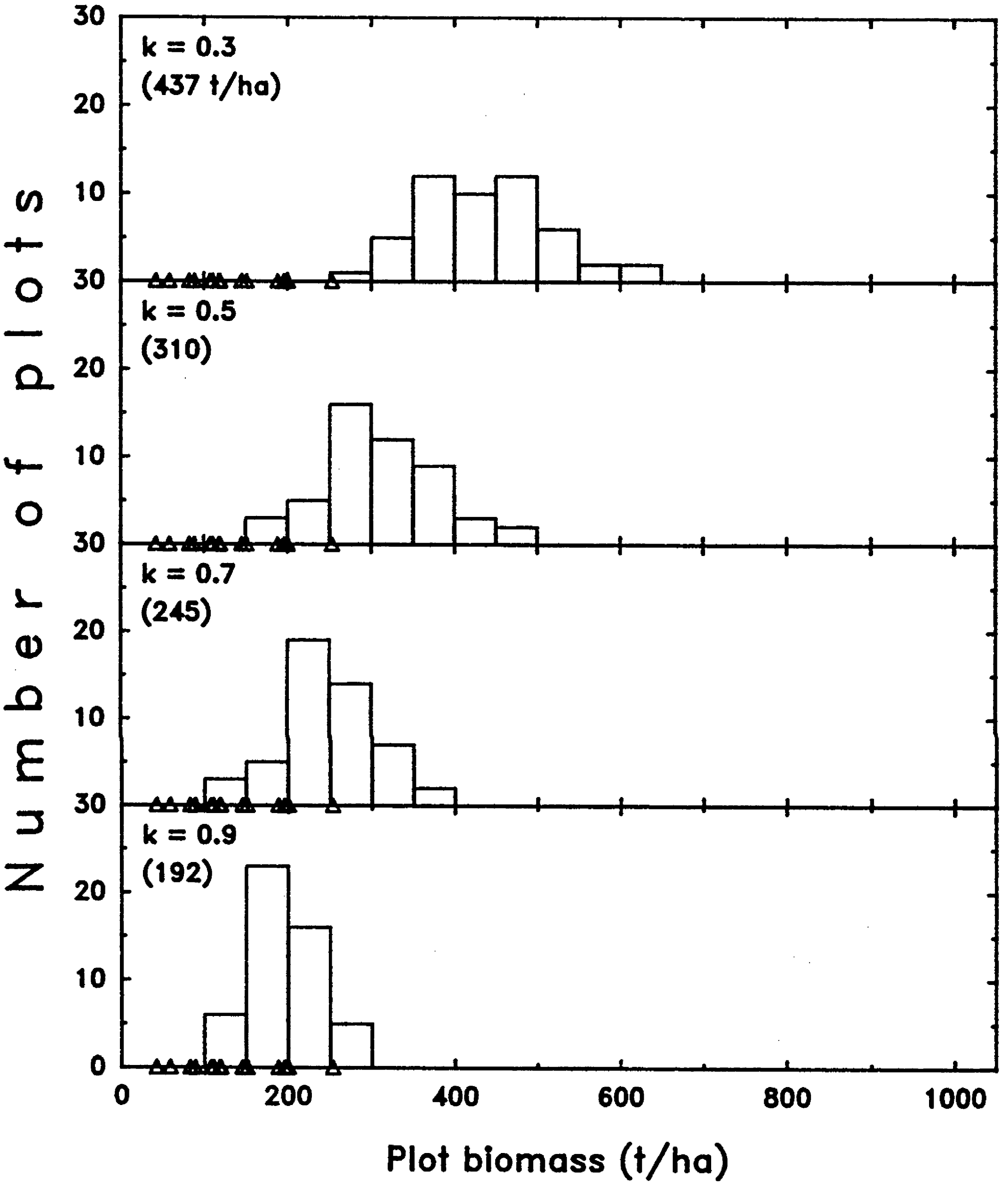

Figure 3.11: Distributions of predicted plot biomass from 50 replicate simulations by BRIND without SPROUT, using values of the light extinction coefficient (k) of 0.3, 0.5, 0.7 and 0.9. Simulations performed at altitude class 6 without fire and without the S(BAR) index. Symbols show biomasses of the 14 field plots in this altitude class. Dominant species was *E. pauciflora*. Numbers in parentheses are mean predicted plot biomass values.

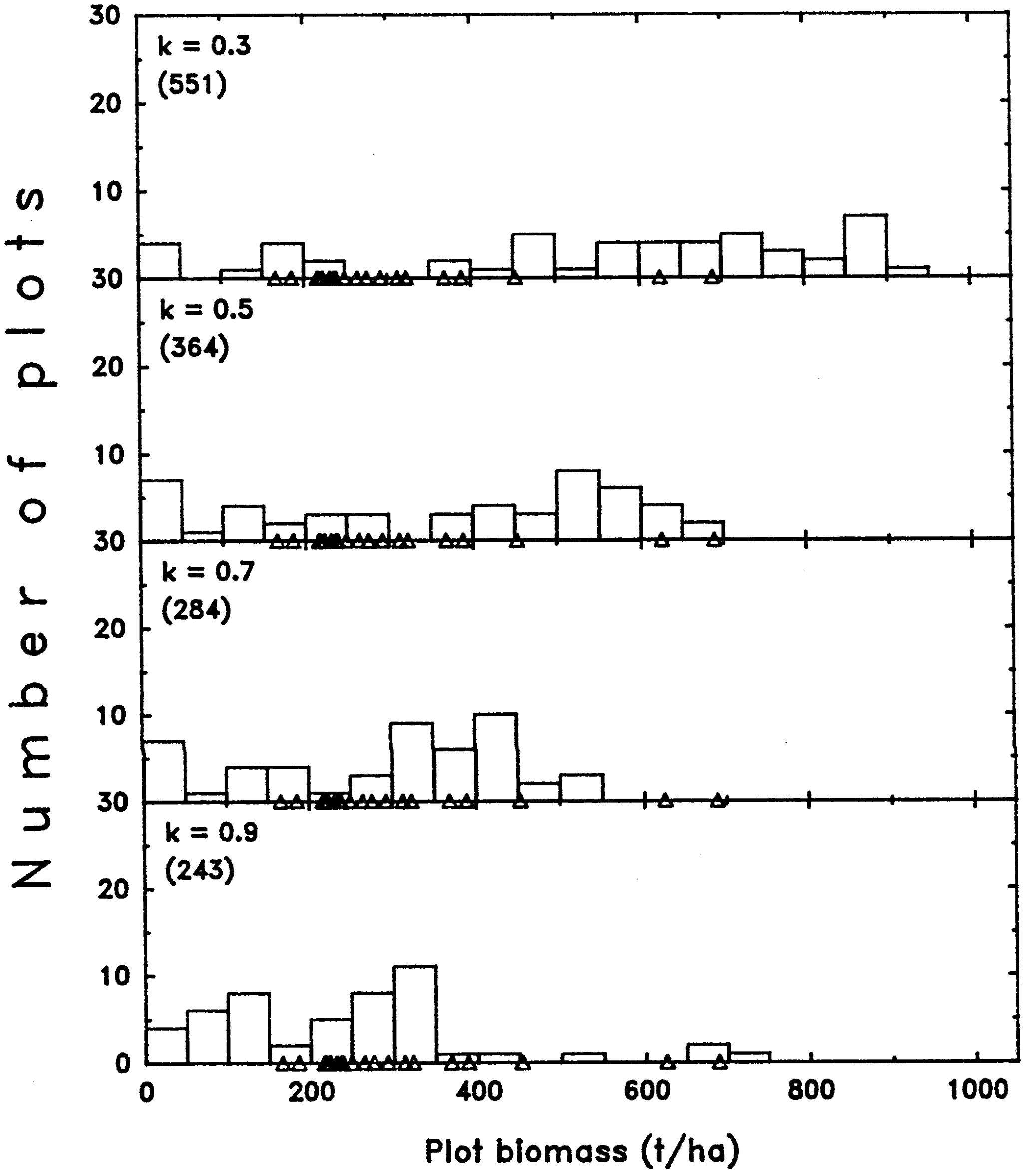

Figure 3.12: Distributions of predicted plot biomass from 50 replicate simulations by BRIND without SPROUT, using values of the light extinction coefficient (k) of 0.3, 0.5, 0.7 and 0.9. Simulations performed at altitude class 4 without fire and without the S(BAR) index. Symbols show biomasses of the 20 field plots in this altitude class. Dominant species was *E. delegatensis*. Numbers in parentheses are mean predicted plot biomass values.

and the achieved biomass would be a product of the growth rates of the species and their longevities.

3.8 Conclusions

Some conclusions about model performance can be drawn from the analyses presented in this chapter. Plot biomass predicted by BRIND was generally much higher than that found in the field plots but over-prediction was reduced by removal of the SPROUT subroutine. Biomass prediction was found to be more sensitive to the value of the light-extinction coefficient in the competition model (k) than the maximum biomass parameter (SOILQ). Numbers of small trees predicted by the model were too high but were reduced to plausible values by removal of the SPROUT subroutine. Predicted species distributions were similar to those in the field plots and differences could be reduced by altering values of the species parameters $\mathrm{DEGD_{min}}$, $\mathrm{DEGD_{max}}$ and G.

In general, model performance (after removal of SPROUT) could be improved by changing values of these species parameters or program constants such as k. However such changes to the model are superficial and do not address the lack of realism in model structure that I described in Chapter 2.

The remainder of the this thesis is concerned with detailed investigation of the submodels in BRIND and construction of alternative ones to be incorporated into the model. These submodels of growth and competition (Chapter 4), establishment (Chapter 5), mortality (Chapter 6) and fire (Chapter 7) are designed to overcome problems I have identified in BRIND.

Chapter 4

Growth and competition

This chapter describes a detailed study of modelling of growth and competition by BRIND. Tree-ring data were collected from *Eucalyptus pauciflora* and compared with climatic data to assess the effects of temperature (which controls growth in BRIND) and soil moisture availability (which I intend to add to the model) on radial growth. The tree-ring data were compared directly with predictions by BRIND of radial growth for the same trees. They were also used to investigate competitive effects of neighbouring trees as expressed by a range of competition indices.

A new growth submodel for BRIND was constructed using information obtained from the tree-ring study, including the addition of an index of soil moisture effects on growth. The new submodel was designed according to the desired change in overall model structure, so it does not reduce the growth of trees to zero at the extremes of their environmental ranges.

4.1 Tree-ring study

Tree-ring data from *E. pauciflora* were collected and analysed to answer specific questions about radial growth in this species. Specifically, the following were examined: the nature of effects of annual temperature variation on radial growth; effects of soil moisture variation; and evidence for reduction of growth by the presence of neighbouring trees.

4.1.1 Climate and radial growth in *E. pauciflora*

Eucalypt species generally do not produce distinct annual growth rings (Ogden 1978), but *E. pauciflora* is one of the few species that does (Dunwiddie and LaMarche 1980, Boland *et al.* 1984). Cambial growth in *E. pauciflora* has been shown to be seasonal with a period of dormancy during the coldest part of the year resulting in clear boundaries to the growth rings (Banks 1982). The period during which the cambium of the tree is active varies from year to year, between sites, and between trees within years (Banks 1982). Initiation of growth in a year is affected by temperature, with growth starting later at lower temperatures (i.e., at higher altitudes), although time of growth cessation varies less with temperature (Green 1967). Green found that trees that started growth later grew faster and that total cambial growth over one season was similar between trees at a site.

Both Green (1967, 1969a) and Banks (1982) found the rate of cell production by the cambium to vary throughout the growth season and to be affected by rainfall and temperature. Green (1967, 1969b) found that growth was fastest when temperature was moderate; when temperature was high growth was slower, but there is a confounding effect of high temperatures being correlated with low soil moisture and high vapour pressure deficits. Rainfall events triggered bursts of fast growth, particularly after dry periods. Banks (1982) found that cambial

Table 4.1: Location and site information of the 3 sites used in the tree-ring analysis. Grid references are to the nearest 50 m on the Australian Metric Grid.

Site	Grid E (m)	Grid N (m)	Altitude (m)	Aspect (°)	Slope (°)
1	660950	6074900	1405	315	4
2	660850	6076400	1410	280	13
3	660350	6072550	1390	275	17

activity ceased when mean daily temperature dropped below 6° to 7.5°C.

In spite of this clear correlation between climate and cambial growth within a growth season, some authors have not found clear climatic signals in ring series from *E. pauciflora.* Griffith (1977) found no clear signal in ring data collected in Victoria; Keith (1982) found no correlation between climatic indices and ring index from a site at approximately 1200 m altitude in the Brindabella Range; and Banks (1982) found no correlation between climate and ring index in trees from the Brindabella Range and Snowy Mountains.

Morrow and LaMarche (1978) showed that diameter growth in *E. pauciflora* can be increased significantly by the removal of chronic insect grazing. They suggested that fluctuations in insect numbers would have such a marked effect on diameter growth that direct effects of climate would be obscured and that climate may only affect ring width indirectly through its effects on insect numbers.

4.1.2 Coring methods

Three stands were chosen in the Brindabella Range for sampling. They were of nearly pure *E. pauciflora* but two had occasional small individuals of *E. dalrympleana.* The three sites were close together, with similar altitude, aspect and slope (Table 4.1) and all on sedimentary soils. It was assumed that the sites are environmentally very similar and share the same climatic history. The sites are all in an area described as having been last burnt by wildfire in 1938 or 1939 (ANU Forestry 1973). The sample plots were circular with a radius of 20 m.

At each site, all trees taller than breast height were mapped in polar coordinates from a centre post using a theodolite and tape measure. For large areas with comparatively few individuals, this method is quicker than using rectangular coordinates, and with azimuth measured to the nearest degree the greatest error of measurement (at 20 m from the centre post) was approximately 0.15 m. The DBH of each tree was measured using a diameter tape.

The three sites were different in their DBH distributions (Fig. 4.1). Site 2 has very few trees with DBH of 5–30 cm and site 1 has only five trees with DBH over 40 cm. Trees for coring were selected to cover the range of DBH at each site and a range of distances to neighbouring trees. Trees nearer the centre of the plot were preferred to those near the edge to maximise information available in the analysis of effects of neighbouring trees.

The cores were taken with a 4 mm increment borer (Fritts 1976) at breast height. Three cores were taken from most trees, but from smaller trees two cores were taken through from one side to the other, giving four ring series. The cores were approximately evenly spaced around the trunk, except where the living tissue was discontinuous because of fire scars and were of varying length, depending partly on the size of the tree and whether or not it was hollow. Detailed climatic data available for comparison covered a total period of only 22 years, so short cores were adequate.

The cores were clamped to wooden mounts to dry for a week, then glued to the mounts with vessels aligned vertically so that a transverse section could be obtained for ring analysis. It was common for cores to break while being removed from the borer, so care was taken to maintain the correct order of the pieces.

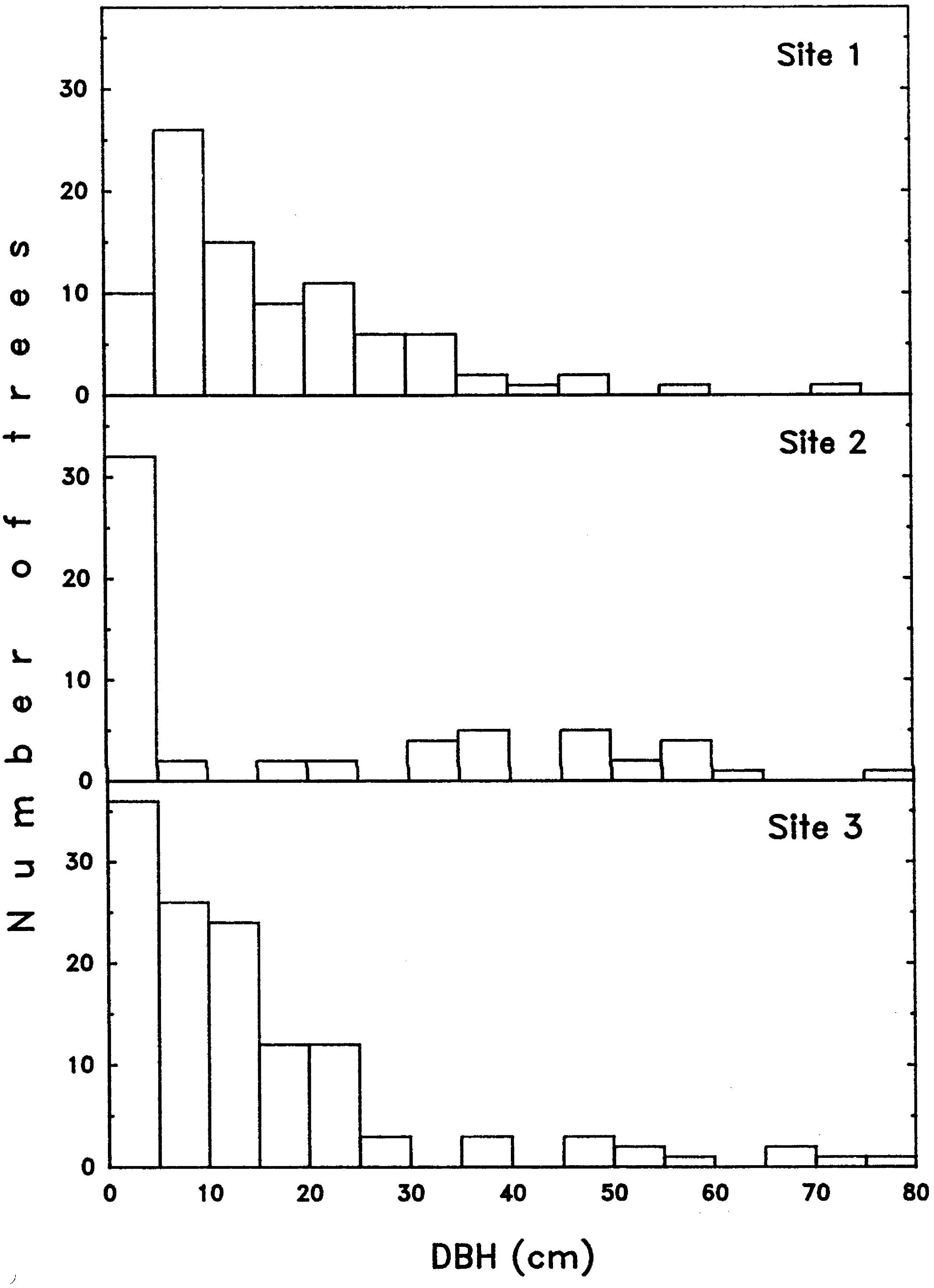

Figure 4.1: DBH distributions in 5 cm classes of the three samples plots used in the tree-ring analysis.

Each core was sanded to a flat surface with 100 grade paper first, then finished with 600 grade paper.

4.1.3 Ring widths

When collecting tree-ring data, it is important to determine whether each ring seen represents a year's growth for that tree. This is done by crossdating: matching distinctive rings between cores within and between trees. This way, a chronology is built up for the tree (Fritts 1976). Rings which have a characteristic appearance from growth patterns of earlywood and latewood are essential in crossdating. For example, Banks (1982) found that the ring produced in the 1964–1965 growth season commonly has a very narrow latewood band. I found this structure in many (but not all) of the cores examined. I also found that the 1982–1983 growth ring was usually narrow, also with very narrow latewood. Matching must be done with care because a particular pattern of growth will not always be seen in all rings of all trees (as with the 1964–5 ring). If such patterns were a response to adverse environmental conditions (such as drought, e.g., 1982–3) it is probable that some trees were less affected by those conditions than others.

Particular care must be taken to allow for false and missing rings which can make determination of a tree's ring chronology difficult; these should be revealed by the establishment of a chronology. A false ring occurs when a tree produces dense, dark wood similar to latewood part of the way through the growth season and then reverts to producing earlywood. This behaviour may be in response to some environmental factor which may or may not affect other trees, and the growth pattern may not be present all the way around the trunk. Missing rings can occur when growth in a year stops without latewood being produced so that there is no clear boundary between two years' growth; this may also happen only part of the way around the trunk.

Banks (1982) found that false and missing rings in *E. pauciflora* were infrequent but false rings may occur as a result of severe drought and suppressed trees were most likely to have missing rings. He also found areas of anomalous growth that were a result of either severe frost or fire. Deposits of kino may also be present as a result of fire or drought.

Some of the cores had uninterpretable rings that could not be crossdated and were discarded from further analysis. These cores had rotten wood or rings that were very narrow with indiscernible boundaries. Some trees had only two usable cores instead of three. In many cores, chronologies could only be reliably established for less than ten years. Table 4.2 lists the numbers and lengths of ring series used in further analyses. After the chronologies were estimated, ring widths were measured using a stereomicroscope with an eyepiece graticule and recorded to the nearest graticule unit, which was *c.* 0.08 mm.

Preliminary analyses of the ring data were carried out using 1-way analyses of variance to investigate variation between cores and between trees. Separate ANOVAs were performed for each year because there were clear trends of changing ring width with time (autocorrelation) and ANOVA assumes independence of observations. The results of these analyses indicated that there were significant differences between mean ring width of the trees over all years at sites 1 and 3, but only in some years at site 2 ($P < 0.05$ in all cases). The difference between sites is probably a result of the limited range of size of trees cored at site 2. These analyses also showed that within-tree variation in ring width was less in most cases than between-tree variation, so it is appropriate to use mean ring width for each tree in further analyses.

In sets of observations that are derived consecutively from the same individual (such as a tree-ring series), one observation could be expected to be influenced

Table 4.2: Numbers of cores and lengths of ring series from each tree used in the growth analyses.

Site no.	DBH (cm)	No. of cores	Length of ring series (years, max. 22)			
1	8.0	3	9	9	8	9
	8.5	4	9	9	9	9
	12.5	3	22	22	22	
	17.0	3	20	16	18	
	17.5	3	9	9	9	
	17.5	3	22	22	22	
	18.0	3	22	22	22	
	18.0	3	8	14	12	
	19.5	2	16	16		
	22.5	3	22	16	13	
	23.0	3	22	22	22	
	24.0	3	10	10	10	
	25.0	3	22	22	22	
	27.0	3	22	22	22	
	27.5	3	22	22	22	
	31.0	3	22	22	22	
	32.0	3	22	22	20	
	33.5	3	22	22	22	
2	36.5	3	11	11	3	
	45.0	2	9	9		
	46.5	2	8	8		
	49.5	3	9	8	9	
	52.0	3	11	12	12	
	53.0	3	7	9	8	
	57.0	3	9	8	8	
	60.5	3	9	9	9	
3	6.5	4	18	21	15	15
	8.5	3	10	11	10	
	11.0	3	20	20	19	
	14.0	3	21	21	20	
	14.0	3	21	18	21	
	16.5	3	18	16	18	
	18.0	3	22	22	22	
	19.5	3	22	22	22	
	20.5	3	22	22	22	
	21.0	3	22	22	22	
	23.0	3	8	8	9	
	26.5	3	15	21	21	
	29.0	3	21	21	17	
	38.0	3	12	12	8	
	73.5	3	22	22	22	
	79.5	2	9	10		

by preceding ones. This phenomenon, autocorrelation, means that a series of untransformed ring width data consists of non-independent observations and be unsuitable for many statistical analyses. Monserud (1986) examined 33 published long tree-ring series for autocorrelation and found it to be significant (with $P < 0.05$) in all of them. He fitted autoregressive models that transform ring data to reduce the correlation. Autoregressive statistical procedures require ring series longer than those in this study to yield any significant results (R. B. Cunningham, personal communication) and so were not used.

Tree-ring series commonly show an overall growth trend on to which is superimposed variation resulting from climate, competition, herbivory and other factors. To enhance correlation of ring width data with climatic data the growth trend should be removed. Standardisation is often used to achieve this, where a curve is fitted to the ring data and each ring width divided by the corresponding fitted value to give a ring index. Curves fitted are often exponential, polynomial or running means, and sometimes high-pass and low-pass filters are used. Ring indices exhibit less autocorrelation than unstandardised data (Fritts 1976).

There is trend of decreasing growth evident in all the ring series in this study (Fig. 4.2). Ring index is defined as ring width R divided by a function of time $f(t)$. If some function of climate and competition $g(C, O)$ is modelled by ring index:

$$\frac{R}{f(t)} = g(C, O)$$

then

$$R = f(t)\, g(C, O). \tag{4.1}$$

which has the same form as the growth equation in forest gap models: parts (i) and (ii) of Eqn 2.1 described a growth trend in time for a tree and the remaining parts are functions of the effects of climate and competition on growth. A multiplicative model like Eqn 4.1 can be fitted to data by using a logarithmic transformation of the dependent variate and fitting a linear regression model.

4.1.4 Effect of climate on growth

Climatic data

To look for a signal of past climate in the ring series, the climatic record of the stands was estimated using data from recording stations in the Brindabella Range. Rainfall, temperature and pan evaporation data from recording stations in the Brindabella Range were available from the CSIRO Division of Water Resources Research. The rainfall data were monthly totals from 36 stations near the Cotter River that collectively cover the period 1963–1978, with most stations having records for shorter periods. The temperature data were daily minima and maxima from six stations near the Cotter River and Uriarra Forest, covering the period 1963–1981, with no station having data for the whole period. To supplement these data for later years, daily rainfall and temperature data from stations at Bendora and Corin Dams covering the period from July 1975 to June 1985 were obtained from records kept at the Bureau of Meteorology. The evaporation data (class 'A' pan) were from four stations in the Brindabella Range and cover the period from January 1966 to July 1979.

The rainfall data were closely correlated between stations with 10 or more years of records (monthly rainfall: Pearson's r^2 from 0.733 to 0.986, $P < 0.001$ in all cases). Mean total July–June rainfall (covering tree growth seasons) showed a non-linear positive correlation with altitude, levelling off to a value of *c.* 1250 mm at altitudes above *c.* 1000 m. The temperature data were also correlated between stations (mean monthly temperature: Pearson's r^2 from 0.952 to 0.992, $P < 0.001$ in all cases). Mean July–June temperature was negatively correlated with altitude

(Pearson's r^2 0.850, $P < 0.001$). The evaporation data were correlated between stations (monthly evaporation: Pearson's r^2 from 0.964 to 0.986, $P < 0.001$ in all cases) and mean total July–June evaporation was negatively correlated with altitude (Pearson's r^2 0.469, $P < 0.001$).

Growing degree-days (DEGD) above a base temperature of 6°C were calculated from the monthly temperature data from the Brindabella Range recording stations using the formula given by Botkin *et al.* (1972), described in Section 2.2.1. The base temperature of 6° was used because of Banks' (1982) finding of cessation of cambial growth below this temperature. Annual degree-days in the Brindabella Range are very closely correlated with mean temperature (Pearson's r^2 0.898, $P < 0.001$).

Estimated values of monthly rainfall, temperature, degree-days and evaporation for the coring sites (*c.* 1400 m) were taken from the highest altitude site (Bulls Head, 1366 m) for the years when that station had records. For other years, estimates were taken from other stations and adjusted by the mean difference between their records and those at Bulls Head. The monthly estimates of rainfall and temperature at the coring sites used in the tree-ring analysis were for the growth seasons ending 1964–1985 (22 seasons) and estimates of evaporation for those ending 1967–1979 (13 seasons). Estimates of monthly evaporation for the other 9 seasons (ending 1964–1966 and 1980–1985) were estimated from rainfall and temperature—a regression of evaporation on rainfall, temperature and the interaction of the two was found to explain 82% of variance in evaporation ($P < 0.001$).

Soil moisture models

Three soil water-balance models were constructed to calculate indices of water stress encountered by the trees each growing season. The first model calculates the number of days when the water content of the soil falls below wilting point, called drought-days. The second and third models use the ratio of actual to potential evapotranspiration as an index of moisture stress. Table 4.3 shows the equations used in each of the soil water models.

All require values of monthly rainfall, temperature and potential evapotranspiration (PE). When pan evaporation data are available, PE is estimated as a fixed proportion of it, based on empirical data of McAlpine (1971) from Alice Springs and Canberra. In all models, a starting point for soil water store is needed: they were all started with soil water at field capacity. This starting point was found to have no effect on the predictions of the models because the growth years all began with months where rainfall exceeds evapotranspiration.

Calculation of drought-days (Table 4.3(a)) uses method of Pastor and Post (1984, 1986), derived from the tables published by Thornthwaite and Mather (1957) for calculation of soil moisture. Days when soil moisture is below wilting point (drought-days) are calculated for each month from soil moisture store and summed over the year. For each month in turn, the soil water level is adjusted: if rain exceeds PE the difference is added to the water store, which is limited to field capacity, while the excess is assumed to run off. If PE exceeds rain, water loss as it accumulates during the year is calculated and used to calculate soil water at the end of that month. Soil water levels at the beginning and end of each month are used to estimate the number of drought-days, as follows: when soil water is below wilting point for the whole month, all days in the month are drought-days; when soil water is above wilting point all month there are no drought-days; when soil moisture during the month falls below or rises above wilting point (crossing it), the number of drought-days is calculated as a proportion of days in the month with moisture below wilting point, assuming linear change in soil moisture during the month. Drought-days are added up over the 'growing months' (those from

Table 4.3: Equations used in the three soil water-balance models. All water measurements are in millimetres. Note that the growth year begins with July as month 1 and June as month 12.

R_i	total rainfall in month i
T_i	mean temperature in month i
PE_i	total potential evapotranspiration in month i, assumed to be $0.8 \times$ pan evaporation
AE_i	actual evapotranspiration in month i
FC	field capacity of the soil
WP	wilting point of the soil
SW_i	soil water level at the end of month i (initialised to FC at the beginning of the model run)
$AccPWL_i$	accumulated potential water loss at the end of month i, initialised to zero at the beginning of each year
$Days_i$	number of days in month i
DD_i	drought-days in month i
MI_i	moisture index for month i

(a) Drought-days model. Based on Thornthwaite and Mather (1957) tables for soil water balance.

Annual drought-days is given by $\sum_{i=1}^{j} DD_i$ for j months with $T_i > 6$°C, where

$$DD_i = \begin{cases} Days_i \dfrac{WP - SW_i}{SW_{i-1} - SW_i} & \text{for } SW_{i-1} > WP \text{ and } SW_i < WP \\ Days_i \dfrac{WP - SW_{i-1}}{SW_i - SW_{i-1}} & \text{for } SW_{i-1} < WP \text{ and } SW_i > WP \\ Days_i & \text{for } SW_{i-1} > WP \text{ and } SW_i > WP \\ 0 & \text{for } SW_{i-1} < WP \text{ and } SW_i < WP \end{cases}$$

and

$$SW_i = \begin{cases} \min\left(FC,\ SW_{i-1} + R_i - PE_i\right) & \text{for } R_i > PE_i \\ \max\left(0,\ FC \times \exp\left[AccPWL_i\left(0.00461 - \frac{1.10559}{FC}\right)\right]\right) & \text{for } R_i < PE_i \end{cases}$$

and

$$AccPWL_i = \begin{cases} AccPWL_{i-1} & \text{for } R_i > PE_i \\ AccPWL_{i-1} + R_i - PE_i & \text{for } R_i < PE_i \end{cases}$$

Table 4.3: (continued)

(b) WATBAL model, based on that of McAlpine (1971), and Keig and McAlpine (1974).

Annual moisture index is given by $\frac{1}{j}\sum_{i=1}^{j}\frac{\mathrm{AE}_i}{\mathrm{PE}_i}$ for j months with $\mathrm{T}_i > 6°\mathrm{C}$, where

$$\mathrm{AE}_i = \min(\mathrm{PE}_i,\ \mathrm{SW}_{i-1} + \mathrm{R}_i).$$

Then soil water is adjusted by

$$\mathrm{SW}_i = \min(\mathrm{FC},\ \mathrm{SW}_{i-1} + \mathrm{R}_i - \mathrm{AE}_i).$$

(c) GROWEST model, based on that of Fitzpatrick and Nix (1970), as described by Booth and Ryan (1985).

Annual moisture index is given by $\frac{1}{j}\sum_{i=1}^{j}\mathrm{MI}_i$ for j months with $\mathrm{T}_i > 6°\mathrm{C}$, where

$$\mathrm{MI}_i = 1.02 - \exp\left[\frac{-3.5\,(\mathrm{SW}_{i-1} + \mathrm{R}_i)}{\mathrm{FC}}\right].$$

Then soil water is adjusted by

$$\mathrm{SW}_i = \mathrm{SW}_{i-1} + \mathrm{R}_i - \mathrm{AE}_i,$$

where

$$\mathrm{AE}_i = \mathrm{MI}_i \times \mathrm{PE}_i.$$

July–June when mean temperature exceeds the 6°C base temperature for growth).

The second model is based on the WATBAL model (McAlpine 1971, Keig and McAlpine 1974) and assumes simple linear extraction of soil water. Monthly actual evapotranspiration (AE) is calculated as the lesser of PE and summed soil water store and rain (Table 4.3(b)). Soil water level is adjusted according to AE and rain, with excess water assumed to run off. The ratio of AE to PE for each month is calculated and its mean value over the growing months in a year used as an index of moisture stress.

The third model is based on the GROWEST model of Fitzpatrick and Nix (1970), as used by Booth and Ryan (1985). In this model (Table 4.3(c)), the soil water store has the monthly rain added to it, from which the monthly moisture index is determined using an exponential equation of soil moisture depletion. The index is used to calculate AE from PE, which is then used to re-adjust the soil moisture store. The mean value of the monthly moisture index over the growing months is used as the annual moisture stress index.

Climatic signal in tree-ring data

There was variation in ring width in common with all cored trees, despite differences in tree size and variation between sites. This variation can be seen in Fig. 4.2, where the open symbols show mean values of DBH increments calculated from all cores at all sites for the 22 growth seasons from 1963–4 to 1984–5. To investigate the relationship between the common variation and climatic indices, a GLM with normal error term was fitted to log-transformed mean DBH increment for the 22 growth seasons. Mean DBH increment was used because it was the variation in common with all trees that was of primary interest. Logarithmic transformation of the dependent variate was taken to stabilise variance in it; this transformation also has the effect of producing a multiplicative model (described above).

Terms tested in the regression were mean growth season temperature, the degree-days index T(DEGD) (Chapter 2) and the three soil moisture models. Also fitted was a term of time, to model the overall trend in growth (Fig. 4.2).

All three soil moisture models require soil field capacity to be specified. Lamb and Florence (1973) and Talsma (1983) published figures for soils in the Brindabella Range. The former authors presented values of 60 to 120 mm for soils that

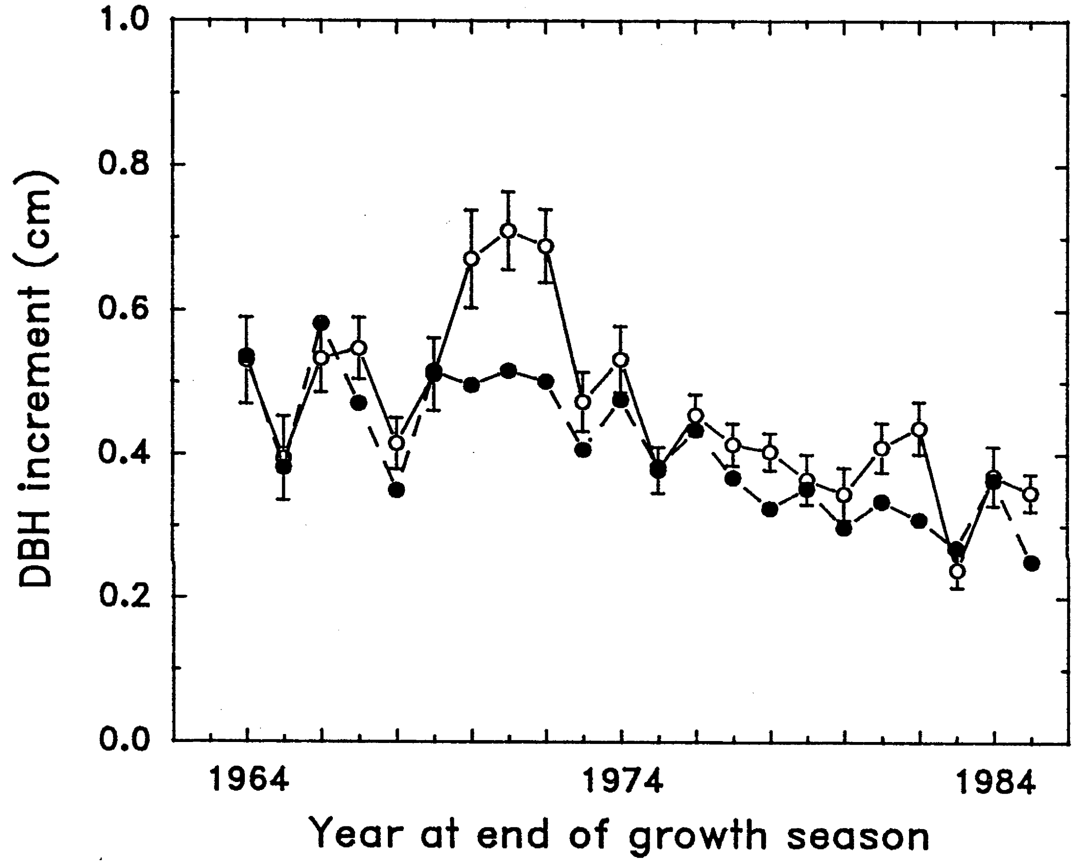

Figure 4.2: Mean DBH increment across all cored trees (solid line, open symbols) with bars showing one standard error of the mean. Dashed line and closed symbols show predictions of the climatic regression model which explained the most deviance over the 22-year period (Eqn 4.2).

Table 4.4: Soil moisture model terms added singly to the regression model for $\ln(\Delta D)$ after the time term (see text) and the percentage of remaining deviance explained by each. Codes for the models are: D for the drought-days model, G for the GROWEST model and W for the WATBAL model.

Model	Field cap. (mm)	% deviance explained	Model	Field cap. (mm)	% deviance explained
D	40	31.2	W	40	45.1
D	60	41.1	W	45	45.8
D	80	41.8	W	50	46.1
D	100	26.6	W	55	46.4
D	120	24.6	W	60	46.8
			W	**65**	**46.9**
G	40	35.8	W	70	46.7
G	60	37.7	W	75	46.3
G	80	37.4	W	80	46.1
G	100	34.8	W	100	44.8
G	120	33.0	W	120	43.1

supported two species (*E. radiata* and *E. fastigata*) but found no clear correlation between capacity and the species' distributions nor between capacity and site differences. Talsma's (1983) results show great variation in capacity at all altitudes. These data provide little indication of appropriate values to use in the soil moisture models, so they were tested in the regression for a range of capacity values from 40 to 120 mm.

The term for time explained most deviance in the regression: 43.5% of total deviance ($P < 0.001$). After the time term, the other terms were tested individually: all the soil moisture models were significant (all with $P < 0.001$), but neither of the temperature terms (mean temperature and T(DEGD)) were. Table 4.4 shows the percentage of remaining deviance explained by each of the soil moisture models for different soil field capacity values. The WATBAL model fitted the DBH increment data better than the other models at 40, 60, 80, 100 and 120 mm field capacity values; this model was also fitted for capacity values in 5 mm steps between 40 and 80 mm (Table 4.4). The WATBAL model, using field capacity of 65 mm, was then added to the regression model; after that, no further terms were significant. The GLM equation which explains the most deviance has the form:

$$\ln(\Delta D) = 51.6 - 0.027Y + 0.903W_{65} \qquad (4.2)$$

where Y is the year at the end of the growth season (1963–1985) and W_{65} is the prediction of the WATBAL model using 65 mm field capacity for the soil. The dashed line in Fig. 4.2 shows the prediction of this regression model.

Increments are reasonably well predicted by the model except for the years ending 1970–2 and 1981–2, where all trees showed accelerated growth. Periods of accelerated radial growth in *E. pauciflora* can signify release from drought (Green 1969a) or be a response to reduced insect grazing (Morrow and LaMarche 1978), which may itself be a delayed response to drought.

It is perhaps surprising that there was no direct effect of temperature on ring width detectable in these data. It appears that moisture is fundamentally limiting at these sites, so any signal of temperature in radial growth is swamped by variation in soil moisture. Booth and Ryan (1985) found similar results in their study of radial growth in *Araucaria cunninghamii*, where an effect of temperature on growth could not be detected but the effect of soil moisture was strong.

4.1.5 Comparison with BRIND simulations

The ring-width data taken from the cores enable a direct comparison to be made with simulations of growth of individual trees by BRIND. The three sites were reconstructed as they probably were in mid-1976 and simulations run from this starting point for nine years to the end of the last season with complete recorded growth rings (1985). A short time period was simulated because 41 of the 42 cored trees had chronologies of at least nine years, but only 32 had chronologies of 10 years or more (see Table 4.2). Longer runs would also have resulted in greater uncertainty about the accuracy of the reconstructed stands, especially for trees that had not been cored.

To reconstruct the three stands in mid-1976, diameters of cored trees were calculated from mean ring widths over the nine years. Diameters of other trees estimated from a regression of cored-tree diameters against nine-year DBH change. At sites 2 and 3 there were a number of very small trees that were probably not present in 1976. Any trees with 1986 diameters less than 2 cm were omitted from the starting stands for the simulations. Because BRIND simulates a plot with a smaller area than the field plots, only those trees in a 1/12 ha area in the centre of the field plots were included. The three mid-1976 plots had 59, 15 and 84 trees, respectively.

For these simulations, the establishment, mortality and sprouting components of BRIND were disabled because they are largely stochastic in their function (Chapter 2) and because I was testing only the growth component of the model. The simulations were also run without fire because there were none during the nine-year period. This comparison is similar to that described by Botkin (1981) for JABOWA, where he used 10-year deterministic (i.e., no establishment or mortality) simulations and compared predicted with real growth increments.

Degree-days values needed for running BRIND were taken from those calculated for the climatic comparisons described above. Ring-width has been shown not to be sensitive to variation in temperature but BRIND may still be able to predict approximately correct increments over a nine-year period of simulation.

Figure 4.3 shows total DBH increment predicted by BRIND against the actual DBH increment over the nine-year period, with a different symbol for each of the three sites. It is immediately apparent that BRIND grossly over-predicted the growth of most trees at sites 1 and 2 during the nine-year period and both under-

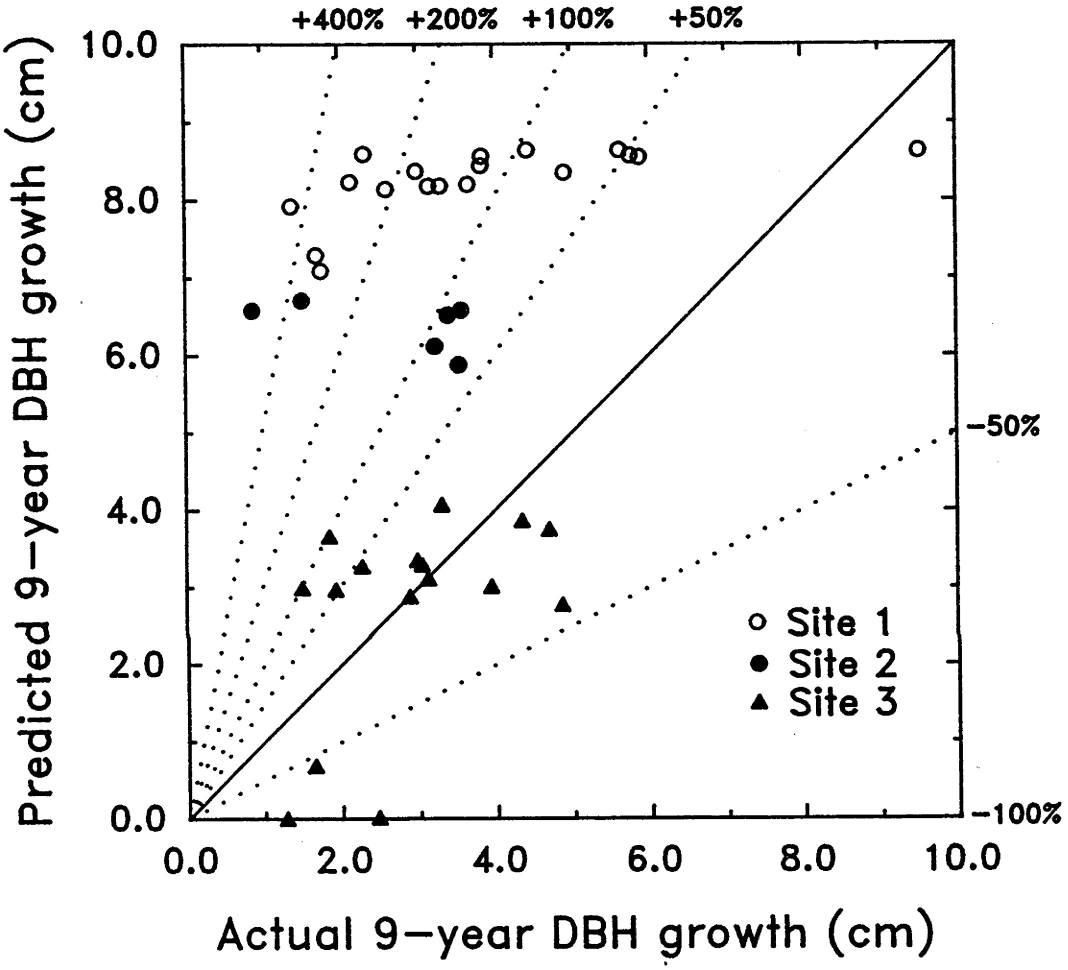

Figure 4.3: DBH growth over 9 years predicted by BRIND vs. actual DBH growth during that period for cored trees at three sites. Dotted lines show extent of errors in prediction.

Table 4.5: Biomass estimates in tonnes per hectare of the three 1/12 ha plots simulated by BRIND in 1976 (beginning of simulation) and 1985 (predicted by the model); and calculated 1986 field biomass values of the same.

Plot		Biomass (t/ha)		
	Number of trees	starting 1976	predicted 1985	field 1986
1	59	52	139	92
2	15	147	210	174
3	84	304	375	370

and over-predicted growth of trees at site 3 (but to a lesser degree than at the other sites). There are clear differences between the sites in error of prediction and at each site the range of predicted increments was much less than the range of actual increments.

There was a ranking between the sites in error of prediction: it was most positive at site 1 and most negative at site 3. The nine-year simulations for the sites were identical except for the lists of trees used to begin them. Differences in growth between the sites are therefore the result of model components that are functions of the trees present on the plot: stand crowding (the S(BAR) index) and competition (the r(AL) index). Table 4.5 shows the three plot biomass at the beginning and end of the 9-year simulations and the calculated biomass in 1986.

Figure 4.4 shows (a) actual and (b) predicted 9-year DBH growth plotted against the DBH of each cored tree in 1976; the curved line shows the potential 9-year growth for these trees, calculated using the growth equation in BRIND (Chapter 2). Trees at sites 2 and 3 show consistent reductions in growth that correspond to plot biomass (Table 4.5): those at site 3 (with highest biomass) were predicted to grow the least; those at site 1 (lowest biomass) the most; and those

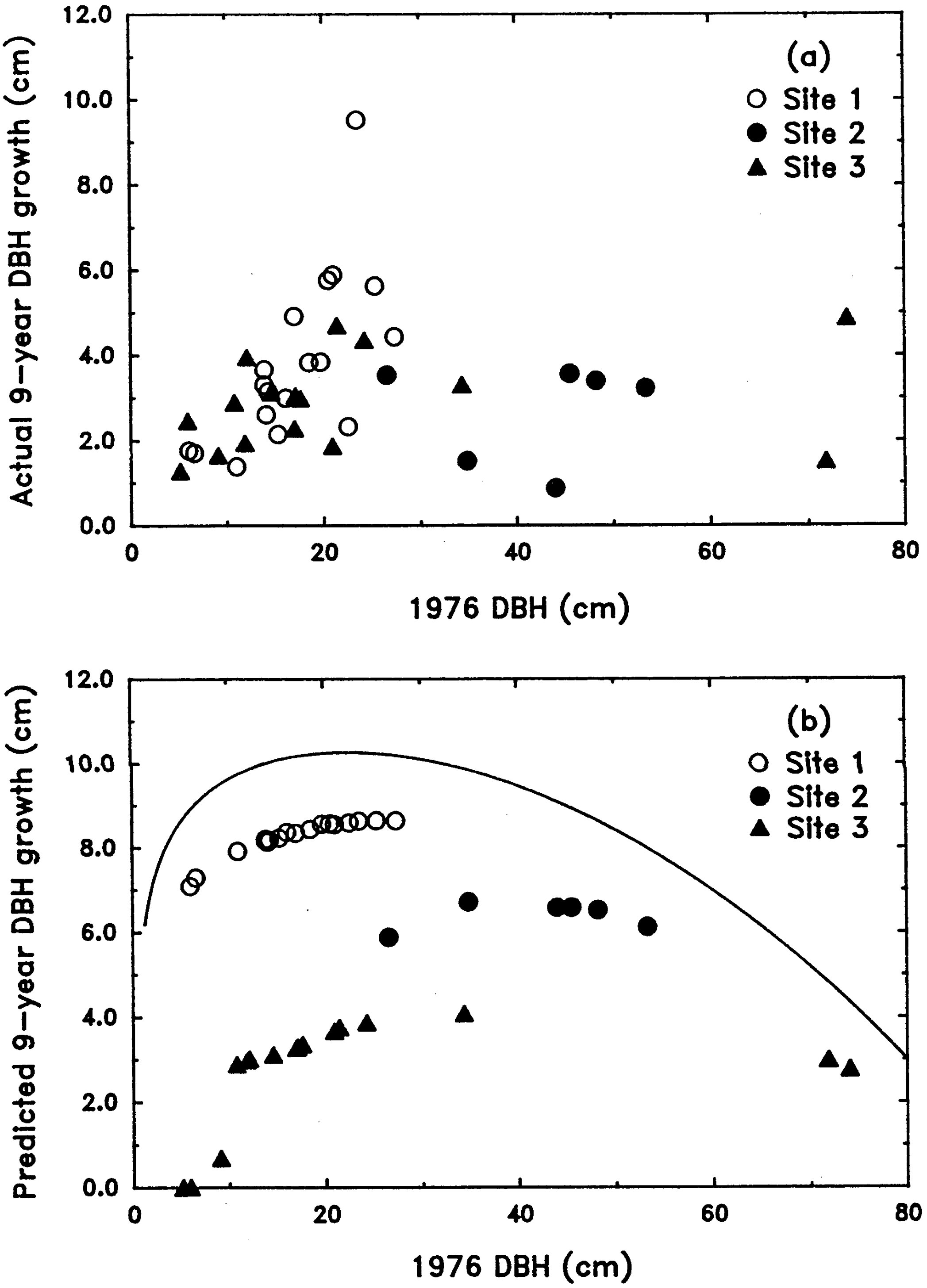

Figure 4.4: (a) Actual and (b) predicted (by BRIND) 9-year total DBH growth vs. DBH at the start of the 9-year period. Line in (b) shows BRIND maximum potential growth curve for *E. pauciflora*.

at site 2 an intermediate amount. This behaviour is a product of the S(BAR) index (Table 2.2, Eqn 2.11). Effects of the r(AL) index (Table 2.2, Eqn 2.8) can also be seen in Fig. 4.4(b) where growth of smaller trees at a site was a smaller proportion of potential growth than that of larger trees. Three trees at site 3 were suppressed for some or all of the 9 years (in the GROW subroutine the model actually sets the growth increments of suppressed trees to zero).

Actual growth of the trees over the 9-year period showed no such pattern with total biomass. It is clear that any effect of standing biomass on DBH growth is more complex than predictions of S(BAR). Plants respond to their immediate environment (e.g., available nutrients and water), which itself is only indirectly affected by standing biomass. Here is a limit to the precision of predictions by BRIND set by the use of an index operating at a different spatial scale (that of the whole plot).

It is also clear from Figs 4.3 and 4.4 that BRIND predicts less variance in growth between trees than was found in the field data. In the model, two trees of similar size will display similar growth rates but there was much greater variation found in the field, which is particularly obvious in Fig. 4.4(a). If BRIND used a range of values of *G* for each species—modelling genetic variation in growth rates and microhabitat differences as suggested in Chapter 2—it may be able to predict variation in DBH increments like that found in the field sites.

4.1.6 Influence of other trees

While the growth model in BRIND assumes that radial growth of trees is reduced by shading from taller trees, I argued in Chapter 2 that the open canopies found in many of the eucalypt stands on exposed aspects in the Brindabella Range suggest that, for these sites, a shortage of soil moisture may limit growth and survival more than availability of light. The tree-ring–climate analysis also demonstrated the importance of moisture to radial growth. The sites chosen for the tree-ring work had open canopies so I chose a general approach to competition, rather than a model of competition for a particular resource (such as light). I investigated effects of neighbouring trees on the recent DBH growth of the cored trees.

Competition indices

Competition indices are designed to express the effect of neighbouring trees on a tree's growth. They vary in structure and complexity, but all indices sum the effects of all neighbouring trees considered to be influential. Dale *et al.* (1985) reviewed many of the indices that have been constructed and classified them into stand density measures, influence-zone overlap, distance-weighted size ratio and growing-space polygon indices.

Stand density measures simply sum the basal areas of neighbouring trees (e.g., Beck 1974). The S(BAR) index in gap models is an example of this type of index, with basal area divided by stand capacity to produce a growth-modifying index (Botkin *et al.* 1972). Some gap models (including BRIND and FORET) use calculated biomass instead of basal area.

Influence-zone overlap indices calculate a circular zone of influence for each tree and sum the extents of overlap by neighbouring trees (e.g., Bella 1971). A development of the influence-zone overlap idea is the concept of an ecological field, based in classical field theory, developed by Wu *et al.* (1985). Each plant has a field of influence contributed to by its crown, stem and roots and at any point in space there is an interference potential which is the sum of the fields of plants in the vicinity. Based on experimental results of Penridge and Walker (1986), a competition index based on ecological field theory was incorporated into the dynamics model RESCOMP (Penridge *et al.* 1987).

In distance-weighted size ratio indices, competitor tree size is expressed as a proportion of subject tree size, adjusted by a function of distance between them, then summed over all competitors (e.g., Hegyi 1974; Daniels 1976). Size ratios are sometimes divided by the square of the inter-tree distance so the index follows the inverse-square law.

Growing-space polygons are non-overlapping geometric apportionments of land calculated according to relative size of, and distance to nearest competing trees (e.g., Moore *et al.* 1973). The proportion of resources allocated to a tree then depends on the area of its polygon.

Lorimer (1983) compared published results of competition indices of the types described above and found little difference in predictive performance (expressed as coefficient of determination of growth), despite wide variation in design. These results suggest that it may not be necessary to construct a complex competition index when a simpler one will be as powerful a predictor of growth.

I investigated competition in *E. pauciflora* using competition indices derived from the general formula:

$$\mathrm{CI}_i = \sum_j \frac{S_j / S_i^a}{X_{ij}^b} \tag{4.3}$$

The index CI_i is calculated for a subject tree i of size S_i from competitor trees j of sizes S_j, at distances X_{ij} from i. Tree size is either expressed as DBH or basal area. This formula can be adapted to produce indices with a range of properties by varying values of the exponents a and b. Forms of this general index can accommodate three main features found in competition indices:

1. Distances to competing trees are commonly used to weight the their influence on the subject tree. All the indices described above include competitor distance except the stand density measures. In Eqn 4.3 the exponent b controls how much inter-tree distance affects the index. If b has a value of zero

the index is not weighted by inter-tree distance.

2. Size of competitor may be expressed relative to subject tree size, so a 20 cm DBH competitor will influence a 10 cm DBH subject tree more than a 30 cm DBH subject tree, for example. In Eqn 4.3 the exponent a controls the effect of competitor size, typically having the value 1. If a has a value of zero, competitor influence is not weighted by subject tree size.

3. The choice of competitor trees to include in the index involves size of 'search radius' and the choice of a 'one-sided' or 'two-sided' index. The search radius is that in which trees grow close enough to influence the subject tree.

   A one-sided index restricts competitors to trees within the search radius that are bigger than the subject tree (competitors j are chosen so that $S_j > S_i$, Eqn 4.3). Two-sided indices include all trees within the search radius. There is evidence for one-sided competition in forests (Lorimer 1983) and in general (Weiner and Thomas 1986). The r(AL) index in forest gap models (Shugart 1984) is clearly a one-sided index: tall trees acquire a disproportionate amount of the available light compared to short trees.

Lorimer (1983) constructed similar indices using data from three even-aged hardwood forests in Wisconsin, Massachusetts and New York State, U. S. A. Lorimer realised that as stands grow and become less dense through thinning, the number of trees within the search radius of a subject tree diminishes and values of the competition indices become artificially less. He introduced the idea that competition indices should be modified to be age-independent, where the size of search radius is adjusted according to some measure of stand development and the distance term DIST_{ij} weighted by the search radius. He found that use of 3.5 × the mean crown radius of overstorey trees on the stand (which he called mcr) provided the best specification of search radius in his indices.

Lorimer (1983) compared growth data with predictions made by polynomial functions of his indices and used the coefficient of determination (r^2) from the least-squares regression of the functions as a measure of fit. He found that indices without a term for distance between trees correlated as well as, and sometimes better than those with such a term. He suggested that indices without inter-tree distances provided information of local competitor density that was sufficient to predict variation in growth. He also found that indices that incorporated the size of the subject tree (where the competitor size was expressed relative to subject size) consistently produced better correlations with growth.

Lorimer also found that one-sided indices correlated better with growth than ones that considered all trees within the search radius as competitors. An index he suggested as useful is:

$$\sum_j \frac{\mathrm{DBH}_j}{\mathrm{DBH}_i}$$

summed over all competitors j of subject tree i within the 3.5 mcr search radius. It is simple to measure in the field and performed well in the correlations with growth (Lorimer 1983).

Correlation with tree-ring data

A selection of search-radius competition indices derived from the general formula shown in Eqn 4.3 were calculated for the cored trees from the tree position data collected in 1986 at sites 1 and 3, and all sites combined (site 2 had too few cored trees for correlations using data from only that site to be significant). Values of the indices were correlated with 1984–1985 DBH increment values using the Spearman rank correlation coefficient.

A search radius of 3.5 mcr (as suggested by Lorimer 1983) cannot be easily applied to these *E. pauciflora* stands because the crowns showed marked asymmetry. At site 1 projections of the crowns were mapped and their areas measured. These data enabled a search radius based on mean crown area to be calculated. Using only overstorey trees (as did Lorimer) the mean crown area was 17.3 m^2; a circle of that area has a radius of 2.3 m (equivalent to mcr), multiplication by 3.5 gives a search radius of 8.2 m. Figure 4.5 shows the values of r^2 for a range of search radii from 4 m to 12 m, correlating 1984–1985 DBH increments with index 1 (Table 4.6) at sites 1 and 3, and all sites combined. The only data points shown are those where the correlations were significant with $P < 0.05$.

Lorimer found that the fit of his indices (as measured by r^2) improved as the search radius was increased from 0 to 3.5 mcr, but beyond that distance the r^2 value changed little. The data here showed a decline beyond an optimal radius of *c.* 8 m. This decline is at least partly a result of edge effects in the sample sites, where with increasing search radius more trees were closer to the edge of the site than the search radius and were thus excluded from the analysis. Because a rank correlation coefficient was used, the loss of degrees of freedom produced a reduction in correlation.

The indices were compared using an 8 m search radius (Table 4.6). Pairs of indices 1 and 4 (one-sided), and 2 and 3 (two-sided) compare the effect of including the distance between trees. In both comparisons, the correlation of the index incorporating a distance term (1 and 2) was approximately the same as that without (4 and 3) at site 1, slightly worse at site 3 and similar overall. These results suggest that the addition of inter-tree distance to BRIND, which would add substantial complication to the model, will not result in a clear improvement in modelling of competition.

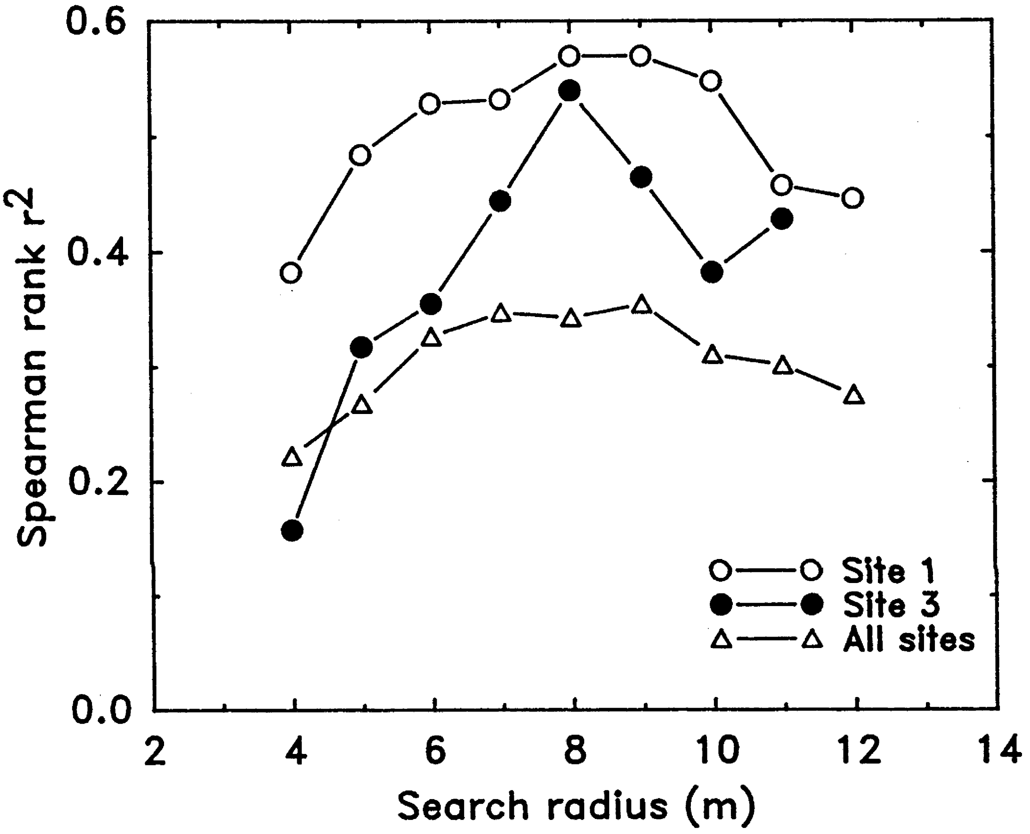

Figure 4.5: Spearman rank r^2 calculated over a range of search radii from competition index 1 (Table 4.6) and log-transformed 1984–5 DBH increments from sites 1 and 3, and all sites combined. Only significant values of r^2 ($P < 0.05$) are shown.

Table 4.6: Spearman rank r^2 values from correlations between competition indiced and log-transformed 1984–5 DBH increments. Indices of DBH or basal area (BA) are summed over all competitor trees j within a search radius of 8 metres of each subject tree i. $\mathrm{DIST}_{ij}$is the distance between the trees. All correlations significant with $P < 0.05$ unless marked ns for non-significant.

Index	Form of index	Competitors	Site			Description
			1	3	All	
1	$\sum_j \frac{\mathrm{DBH}_j/\mathrm{DBH}_i}{\mathrm{DIST}_{ij}}$	$\mathrm{DBH}_j \geq \mathrm{DBH}_i$	0.57	0.54	0.34	One-sided Distance term Subject tree size
2	$\sum_j \frac{\mathrm{DBH}_j/\mathrm{DBH}_i}{\mathrm{DIST}_{ij}}$	all	0.21	0.44	0.11	Two-sided Distance term Subject tree size
3	$\sum_j \mathrm{DBH}_j/\mathrm{DBH}_i$	all	0.18	0.50	0.12	Two-sided Subject tree size
4	$\sum_j \mathrm{DBH}_j/\mathrm{DBH}_i$	$\mathrm{DBH}_j \geq \mathrm{DBH}_i$	0.56	0.64	0.36	One-sided Subject tree size
5	$\sum_j \mathrm{DBH}_j$	$\mathrm{DBH}_j \geq \mathrm{DBH}_i$	0.44	0.77	0.38	One-sided
6	$\sum_j \mathrm{DBH}_j$	all	0.00^{ns}	0.08^{ns}	0.00^{ns}	Two-sided
7	DBH_i	none	0.47	0.46	0.17	Size–increment correlation

Competition indices that include subject tree size may show spurious correlation with growth because of its inclusion. Similarly, one-sided indices may also show spurious correlation because of the indirect inclusion of subject-tree size (competing trees are all bigger). Any index of competition that is to be used for prediction of growth should correlate with it better than a simple function of tree size. 'Index' 7 in Table 4.6 shows the strength of correlation between the subject trees and their DBH increments.

The pairs of indices 1 and 2, 3 and 4, and 5 and 6 can be compared to assess differences between one-sided and two-sided competition indices. The one-sided indices (1, 4, 5) correlated with DBH growth better than their equivalent two-sided indices (2, 3, 6) and mostly better than just subject-tree size (index 7). Inclusion of subject-tree size (DBH_i in indices 3, 4) produced better results in two-sided, but mixed results in one-sided indices.

In summary, there is evidence of effects of neighbouring trees on radial growth in *E. pauciflora* on these three sites. Comparisons of competition indices suggest that an appropriate search radius for these eucalypt stands is *c.* 8 m, inclusion of inter-tree distance is not necessary, that one-sided indices are better and that the inclusion of subject-tree size does not always improve correlation. The different correlations at sites 1 and 3 and the weaker correlation when data from the sites were pooled suggest that there are between-site differences in extent of influence of trees on others.

These results largely agree with those of Lorimer (1983). Other authors have also found one-sided indices to be better than two-sided ones for predicting growth (e.g., Weiner and Thomas 1986). Lorimer's suggested index includes subject-tree DBH because it correlated better than the equivalent one without, whereas in this study the difference between the two (indices 4 and 5) is not as clear. The application of these results to a new growth model for BRIND is discussed below.

4.2 A new growth submodel

The results of the analyses of growth and competition described in this chapter provide guidance for the construction of a new growth model for BRIND. I decided to retain the multiplicative structure from BRIND but add an index of soil moisture, change the temperature index and use a simple, one-sided competition index instead of the r(AL) shading index. The stand crowding index S(BAR) was kept. The new model was then tested against the 9-year growth data used in Section 4.1.5.

4.2.1 Soil moisture

The examination of climatic effects on *E. pauciflora* described above suggests that annual radial growth is affected by available soil moisture and that a simple moisture index can be used to model this variation. Based on the assumption that similar patterns of moisture response apply to growth in other eucalypt species in the Brindabella Range, a soil moisture index was added to the growth model in BRIND.

I chose to use the WATBAL model because its predictions correlated best with the tree-ring data and it was the simplest of the three models examined. For the WATBAL model to be incorporated into a new version of BRIND it was necessary to develop a climate model with monthly resolution for the Brindabella Range and to formulate the index that specifies how changes in soil moisture affect radial growth.

Climate model

The climate model is required to predict monthly values of temperature, rainfall and evaporation for any site in the Brindabella Range, given its altitude, aspect (azimuth) and slope. Relationships were established between the CSIRO climatic data and topographic variates to give monthly estimates of means and variances at any site. These parameters were then used to describe a distribution from which monthly estimates are randomly selected.

All three climatic variates showed clear responses to altitude (Section 4.1.4). Relationships between altitude (A) and each of temperature (T), rainfall (R) and evaporation (E) were established and used to adjust mean monthly values from selected recording stations ('base' stations, b) to give an expected value for month i:

$$\mathrm{E(R)}_i = \mathrm{E(R}_b)_i + 8.33 - \exp[2.12 - 2.07 \times 10^{-3}(\mathrm{A} - 1040)] \tag{4.4}$$

$$\mathrm{E(T)}_i = \mathrm{E(T}_b)_i + 5.80 \times 10^{-3}(1366 - \mathrm{A}) \tag{4.5}$$

$$\mathrm{E(E)}_i = \mathrm{E(E}_b)_i + 5.88 \times 10^{-2}(1366 - \mathrm{A}) \tag{4.6}$$

The relationship of altitude with rainfall is curved (Eqn 4.4), while that with the other climatic variates is linear (Eqns 4.5 and 4.6). Variance in monthly rainfall, temperature and evaporation was examined and found to not vary significantly with altitude: at all altitudes, $\mathrm{Var(R}_b)_i$, $\mathrm{Var(T}_b)_i$ and $\mathrm{Var(E}_b)_i$ were used as estimates of $\mathrm{Var(R)}_i$, $\mathrm{Var(T)}_i$ and $\mathrm{Var(E)}_i$, respectively (values of all the base station figures are given in the program listing in Appendix C, in the function `Climate`).

While the relationship between the CSIRO climatic data and altitude is clear, they provide little information about how temperature, rainfall and evaporation vary with aspect or slope. Many of the recording stations were situated on ridge tops, in gullies or flat clearings (J. Burns, personal communication). It is expected that rainfall varies little with aspect and slope on a small scale. On a larger scale, there are clear differences—the western side of the Brindabella Range as a whole has lower rainfall than the eastern side. The climatic data used in this study were all from the Cotter River catchment, to the east of the main range and hence are only applicable to climatic prediction in that region. The relationship between temperature and slope or aspect is probably quite complex, but temperature is less directly important to soil moisture calculation than evaporation (which is related to temperature).

Evaporation varies with aspect and slope and results from the heat load at a site, a product of incident radiation and diurnal temperature variation. Radiation is symmetrical about the north-south axis (i.e., total incident radiation on an east-facing slope during a day is the same as that on a west-facing slope at the same location on the same day) but heat load is not: heating of the air during the course of a day results in higher heat load on westerly slopes because the air temperature is higher when direct radiation falls (in the afternoon) (Tajchman and Lacey 1986).

Radiation load can be simply calculated from latitude, slope and aspect of any site on Earth. It is often expressed as a radiation index, the ratio of radiation on that slope to radiation falling on a level plane at the same latitude. Austin and co-workers (Austin 1971; Austin *et al.* 1983, 1984, 1990) used radiation index in the prediction of expected distributions of eucalypt species and Stage (1976) used a similar index to modify site index calculated for forest stands.

Swift (1976) provides equations for calculating radiation values and radiation index derived from the method of Lee (1963). Swift's method includes the calculation of an 'equivalent slope', a level plane at a different longitude to the slope of interest but subject to the same amount of radiation. The longitudinal offset between the subject slope and the equivalent slope can be expressed in hours and is approximately equivalent to the time difference (in hours) between maximum radiation and maximum heat load during the course of a day.

In the new climate model, a 'heat index' is calculated from the radiation index given by the Swift (1976) equations multiplied by a simple function of the offset between subject and equivalent slopes, assuming that an offset of 3 hours results in a change of *c.* 20% in heat load (P. M. Fleming, personal communication):

$$H = I(1 - 0.067o) \tag{4.7}$$

where H is heat index, I is radiation index and o is offset in hours (positive for easterly slopes). The equations provided by Swift (1976) calculate radiation index and offset for a specified day of the year, so monthly approximations were made by using the values for the middle day in each month. Figure 4.6 shows heat index for a 15° slope at 8 aspects for the months July, October, January and April. For steeper slopes, the range of variation is greater. The functions shown in Fig. 4.6 are of very similar shape to the trends of site index with aspect and slope shown by Stage (1976, his Fig. 2).

Prediction of moisture availability using the WATBAL model requires estimates of soil water storage capacity in millimetres. Because published figures of field capacities in the Brindabella Range vary greatly (Section 4.1.4), for initial testing of the moisture model added to BRIND, a water storage value of 65 mm was used (the value in the model that best fitted the tree-ring data). Values of the moisture index were found to be less sensitive to field capacity than to rainfall input.

Growth-modifying index

Pastor and Post (1986) used a linear index of drought-days to modify growth in an extension of the FORET model (see also the description by Shugart 1984). In their model, each species is described by a parameter of the maximum number of

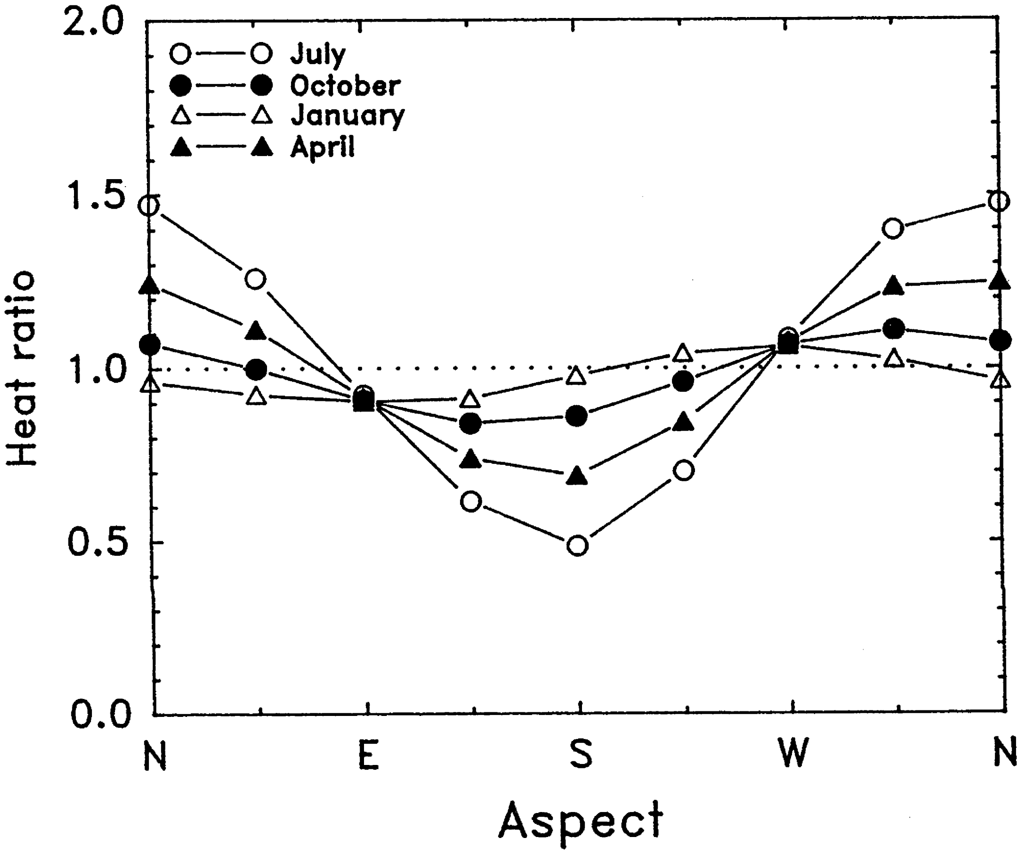

Figure 4.6: Values of the heat index calculated for a 15° slope on eight aspects at four times of the year. The index shows greater variation for steeper slopes.

drought-days it can tolerate. The index is linearly scaled from 1 when there are no drought days to 0 if the number of drought-days in a growth year is equal to or greater than that maximum. The values for the species' maximum drought-days parameters were taken from maps of their distribution, in a similar way to DEGD. An equivalent linear species growth-modifying index for the WATBAL moisture stress index (ratio of actual to potential evapotranspiration, called MI) would be linearly scaled from 0 at the minimum value the species can tolerate to 1 when the stress index is 1.

The single parameter needed for each species is the minimum value of MI it can withstand. Vegetation and site data collected for a land-use survey of the Queanbeyan-Shoalhaven area (Gunn *et al.* 1969, made available by M. P. Austin, CSIRO) were used to estimate values for *E. pauciflora.* These data comprise records of presence of the species in 261 sample quadrats (20 m × 20 m) and calculated mean annual values of MI from a model very similar to the WATBAL model used in this study (Austin 1971). From these data, the minimum mean MI at any site with *E. pauciflora* was 0.58.

Use of a linear growth-modifying index with the minimum MI value as a parameter assumes that the radial growth of a species is reduced to zero by moisture stress at the dry end of its range of occurrence. This assumption is equivalent to the assumption that growth is reduced to zero at the ends of species' ranges of DEGD occurrence (Section 2.2.3) and likewise may not be valid. Further, it is not certain that the response of growth to available moisture is linear. The required data to overcome these problems are not currently available for the species being modelled in the Brindabella Range.

The linear growth-modifying index that is described above would predict no growth of trees of *E. pauciflora* in years when MI is 0.58 or less (Fig. 4.7(a)). In

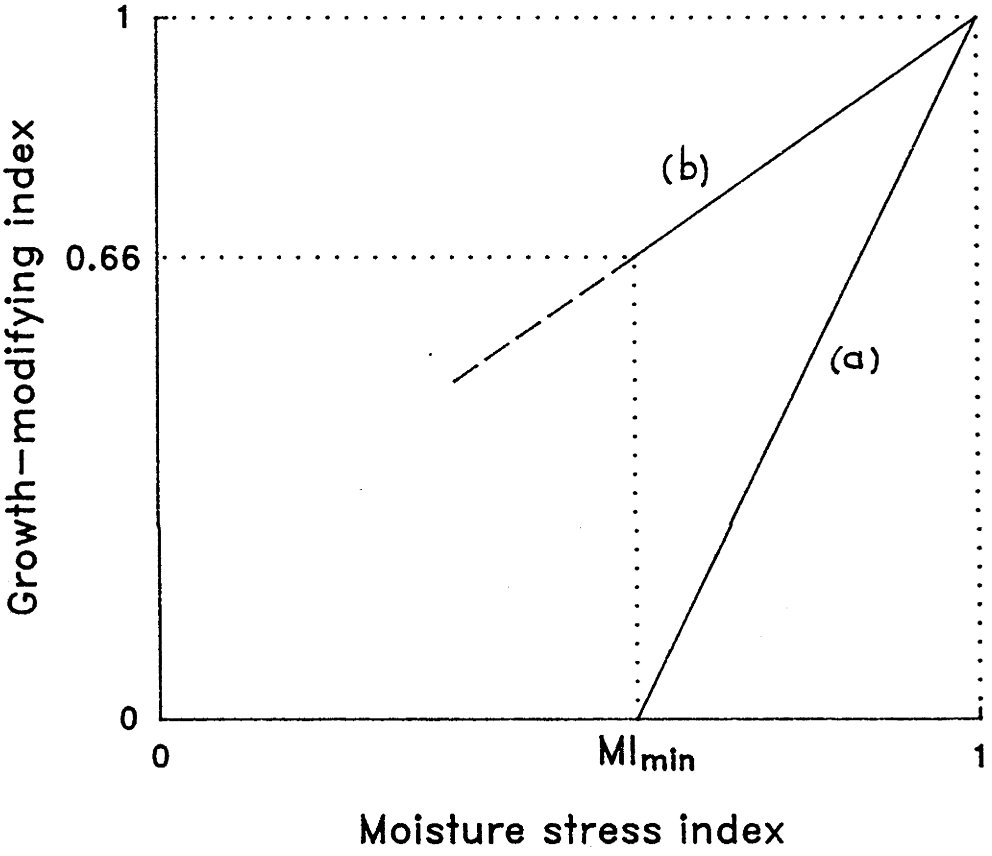

Figure 4.7: Values of growth modifying index M(MI) against mean moisture stress index MI. (a) Complete reduction of growth at MI = MI_{min}. (b) Reduction of growth to 0.66 at MI = MI_{min} (see text).

the growth-climate analysis described in Section 4.1.4, the lowest MI value calculated using the best-fitting soil moisture model (WATBAL model using 65 mm field capacity) was 0.48, but the trees' growth, while reduced, was not zero (season ending 1968, Fig. 4.2). The minimum values of MI from the Queanbeyan-Shoalhaven data are lowest *mean* values from 261 sites, not the lowest annual MI values at those sites. The mean value of MI from the 22 years of climatic data used above (and hence comparable with the Queanbeyan-Shoalhaven values) was 0.81.

Growth of *E. pauciflora* could be expected to be zero when MI is zero, but the equation of the best-fit regression model in Section 4.1.4 predicts non-zero growth for zero MI because a logarithmic transformation was used to stabilise variance and produce a multiplicative model (Eqn 4.2). Values of MI over the 22 years ranged from 0.48 to 1.0 so the regression model appears to be inapplicable outside this range of MI. The regression equation 4.2 can be used to estimate the extent of growth reduction at MI = 0.58 relative to optimal conditions of MI = 1. From Eqn 4.2, the ratio of ΔD at MI = 1.00 to ΔD at MI = 0.58 is constant for all years because the data were fitted to a log-transformed variate and has the value 0.66. This suggests that soil moisture stress reduces DBH growth in this species by approximately one third at the extremes of its range (i.e., at MI = 0.58).

For the first implementation of a soil moisture model as an addition to the growth model in BRIND, I decided to use a growth-modifying index with the value 0.66 at the minimum MI values taken from the Queanbeyan-Shoalhaven data for each species:

$$\mathrm{M(MI)} = \frac{0.34\mathrm{MI} + 0.66 - \mathrm{MI}_{\mathrm{min}}}{1.0 - \mathrm{MI}_{\mathrm{min}}} \tag{4.8}$$

where M(MI) is the growth-modifying index, MI is the moisture index in the current year and $\mathrm{MI}_{\mathrm{min}}$ is the minimum MI value for a species (0.58 for *E. pauciflora*).

This index is illustrated by the line marked (b) in Fig. 4.7.

4.2.2 Temperature

The analysis of climatic effects on growth in *E. pauciflora* provides little indication of how temperature affects radial growth in this species. The absence of any detectable effect of temperature on growth may have been because there was not enough temperature variation between the three sites where the tree-ring data were collected.

Temperature is important in determining whether cambial growth can occur and thus the length of the growth season (Green 1967, Banks 1982). There were significant periods of most years in the tree-ring analysis when the temperature was too cold for growth. The new growth model should at least incorporate an index of the length of the growing season each year. The growth parameter G expresses the annual growth rate for a species, but if the growth season is shortened G should be adjusted accordingly.

A simple index of the length of the growing season is:

$$\mathrm{T(GM)} = \frac{\mathrm{GM}}{12} \tag{4.9}$$

where GM is the number of 'growing months' in the year. Growing months were defined in the calculation of soil moisture indices as being those with mean temperature greater than 6°C, after Banks' (1982) finding that cambial growth in *E. pauciflora* stopped below this temperature (Section 4.1.1).

The index shown in Eqn 4.9 is imprecise because it can have only twelve discrete values between zero and one, which will result in sudden changes in value as altitude changes. I decided to use a more complex method of calculating GM.

Instead of counting the number of months in a growth season with mean temperature greater than 6°C the following algorithm, which estimates the proportion of each month when the temperature was too cold for growth, was used:

$$\mathrm{GM} = \sum_{i}^{12} \mathrm{GM}_i, \text{where} \tag{4.10}$$

$$\mathrm{MG}_i = \begin{cases} 0 & \text{for } \mathrm{T}_{i-1} < 6^\circ \text{ and } \mathrm{T}_i < 6^\circ \\ \dfrac{\mathrm{T}_i - 6}{\mathrm{T}_i - \mathrm{T}_{i-1}} & \text{for } \mathrm{T}_{i-1} < 6^\circ \text{ and } \mathrm{T}_i \geq 6^\circ \\ \dfrac{\mathrm{T}_{i-1} - 6}{\mathrm{T}_{i-1} - \mathrm{T}_i} & \text{for } \mathrm{T}_{i-1} \geq 6^\circ \text{ and } \mathrm{T}_i < 6^\circ \\ 1 & \text{for } \mathrm{T}_{i-1} \geq 6^\circ \text{ and } \mathrm{T}_i \geq 6^\circ \end{cases} \tag{4.11}$$

This method for calculating GM has also been incorporated into the soil moisture model.

4.2.3 Competition

The results of work with competition indices described above (Section 4.1.6) show a significant relationship between growth and measures of density of neighbouring trees within a small radius. The exact form of the index appears not to be critical and two of them, labelled 4 and 5 in Table 4.6, correlate as well or better with DBH increment than more complicated indices. Index 4 is the sum of diameters of larger neighbouring trees within an 8 m radius of the subject tree, divided by its diameter; index 5 is similar but does not include subject tree DBH.

The correlation tests described above were performed using the Spearman rank coefficient, which has the advantage that its use does not assume a particular statistical model (i.e., linear relationship, normal distribution of errors). Its disadvantage is that very little predictive value is gained from its use. Having reduced the number of possible indices from many to two using the rank coefficient, the two chosen were examined in more detail.

Index 5 is the summation of diameters of surrounding trees, but usually an additive index is constructed by summing basal areas. If Index 5 is constructed using basal area information its rank correlation is very similar to that with DBH, so this basal area index was compared to the tree-ring data. Figure 4.8 shows 1984–5 DBH increment for the cored trees plotted against (a) Index 4 calculated from DBH and (b) Index 5 calculated from basal area (in m^2) for the three field sites. The lines are linear regression equations fitted to the two sets of data. Both sets of data show the pattern of a spread of data enclosed by an apparent envelope which in these cases can be approximated by a straight line. This pattern of data conforms with the conceptual model of competition in BRIND, where neighbouring trees place a limit on growth, which in any tree may be further reduced by other factors.

For the first version of a new growth model I decided to use Index 5 calculated using basal area data. This index has no direct term of subject-tree size but has the indirect term resulting from the one-sided choice of competitors. The slope of the regression line in Fig. 4.8(b) approximates the apparent slope of the envelope line. When the same index values were plotted against DBH increments from the three previous years (1981–2, 1982–3 and 1983–4), the strength of correlation was less, but the slope of the fitted line was similar.

The competition index chosen has the formula:

$$\mathrm{CI} = \sum \mathrm{BA}_j \tag{4.12}$$

where BA_j is the basal area in m^2 of each competitor tree j. Competitors are

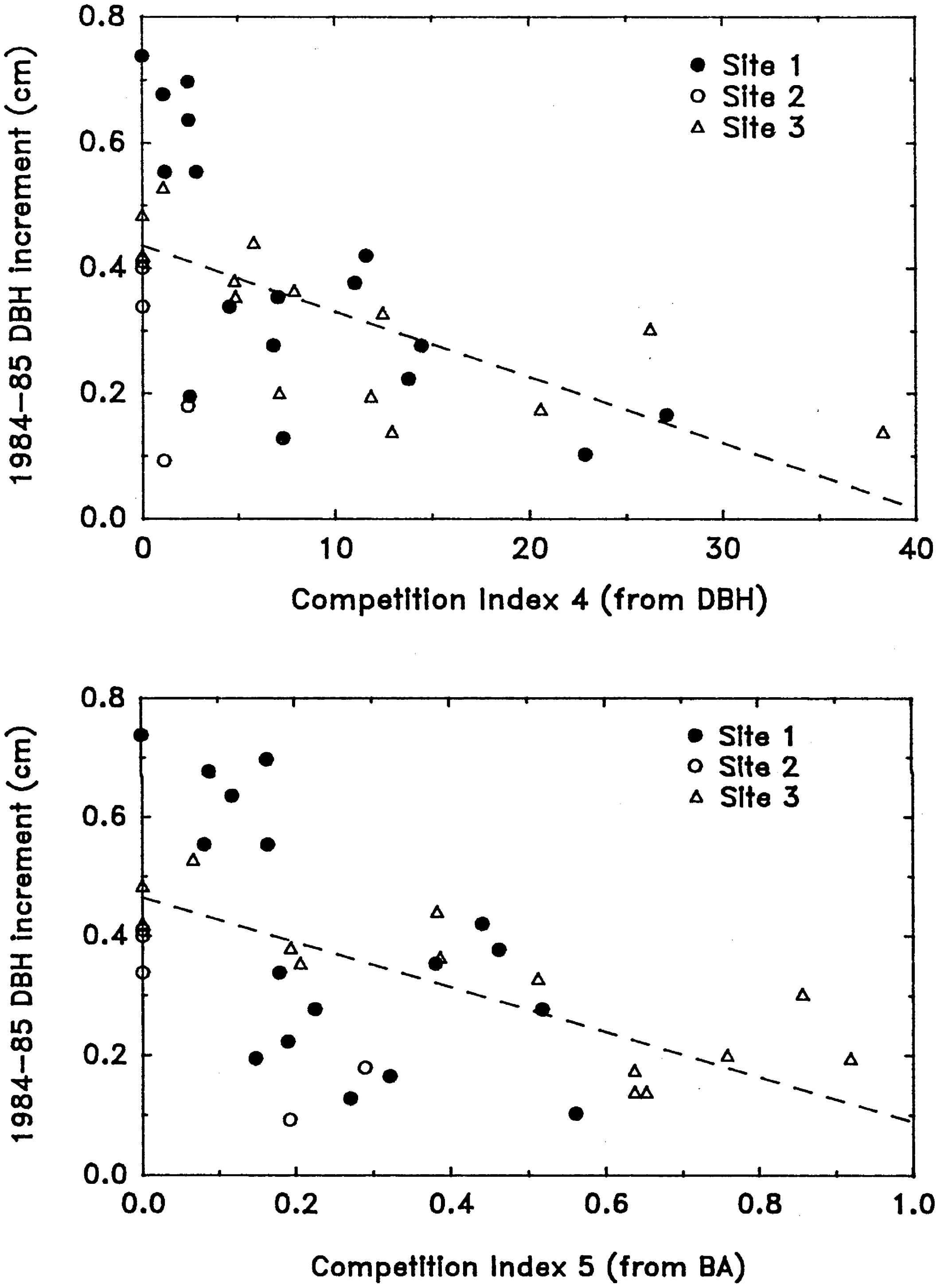

Figure 4.8: 1984–5 DBH increments of cored trees plotted against competition indices 4 and 5 from Table 4.6. (a) Index 4 calculated using DBH; and (b) Index 5 calculated using basal area. Lines are from regression fits to the data.

those with DBH greater than or equal to that of the subject tree within a radius of 8 m. The growth-modifying index calculated from CI is:

$$\mathrm{P(CI)} = 1 - \frac{k}{\mathrm{A}}\mathrm{CI} \tag{4.13}$$

where k is 75.2 and A (plot area) is 201 m^2; giving $k/\mathrm{A} = 0.374$, the mean slope of the regression lines for the 1984–5 growth season. The parameter k in Eqn 4.13 is functionally equivalent to k in the r(AL) index (hence the same symbol) in that it expresses intensity of competition between trees scaled to be independent of plot size.

The new growth submodel will include the 'stand crowding' index S(BAR) from BRIND to model total resource availability and retention of nutrients in standing biomass (Chapter 2). The soil moisture index M(MI) is a relative index only and does not control absolute moisture availability. Because of the one-sided competition index the largest tree on the plot does not have its growth limited in any other way by the biomass on the plot. For example, a model without a crowding index would predict the same growth of the dominant tree with no competitors as it would with a number of (smaller) competitors. In the second case, there must be less water available to the dominant tree because it will not completely stop the other trees from transpiring and growing.

A difficulty with S(BAR) is calculating meaningful values of the maximum biomass parameter SOILQ. Shugart and Noble (1981) used a value of 1000 t/ha, calculated using field data from mature stands of *E. delegatensis* in the Brindabella Range, but this species is found only on sheltered sites with deep soils. Most other sites are less productive and the field data described in Chapter 3 reflect this. As a first approximation, I will estimate SOILQ by using mean soil moisture index MI and mean growth season index T(GM) to scale an optimal value of 1000 t/ha. These two indices model effects of reduced moisture availability and shortened growth season on site productivity.

4.2.4 Potential growth increments

In Chapter 2 I argued that use of a single value of the growth rate parameter G for all trees of a species by gap models ignores the natural variability in growth between trees of a species and results in a loss of realism in modelling. Natural variability between trees at a site is a result of variation in genetic potential and microhabitat differences whose effects are practically impossible to separate from each other.

If a fixed value of G which is 'average' for a species is used, the model will be unable to predict the faster growth rates shown by some trees. Figure 4.9 shows all DBH increments from all cored trees at the three sites plotted against DBH before each increment was added and a line showing the potential DBH increments calculated using the equation from BRIND for *E. pauciflora.* The points above the line are from a few fast-growing trees: most of those at site 1 are from one tree and those at site 3 are from a single tree. Because final predicted increments are always less than those shown by the line in Fig. 4.9 the model cannot predict the growth of these fast-growing trees. If a larger value of G is used which accurately describes the maximum potential growth rate of any tree (i.e., encompasses all the points in Fig. 4.9), then predicted growth rates of many trees will be too high.

Variation in growth between trees could be modelled if a distribution of G values were specified for each species instead of a single value. Each tree would be assigned a value of G at establishment that represents the combination of its genetic potential for growth and the particular site where it established. To provide a distribution of G for a species, the shape of the distribution needs to be

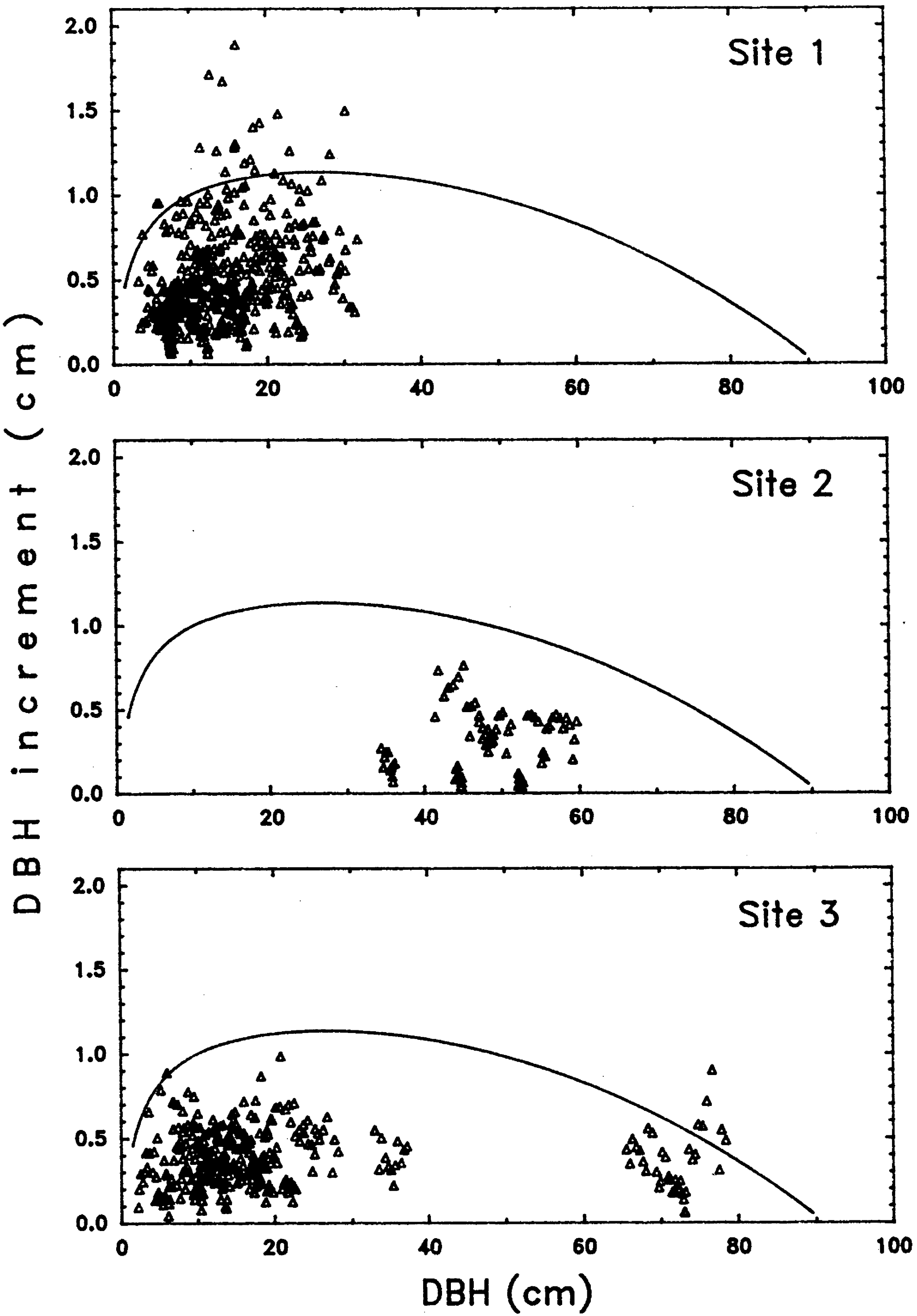

Figure 4.9: All DBH increments measured from cores at the three sites in the tree-ring study, plotted against DBH before each increment was added. Curve shows the maximum annual DBH increments calculated by BRIND for *E. pauciflora* (Eqn 2.5).

specified and then values for parameters that describe the distribution's expected value and spread.

Some of the tree-ring data described earlier in the chapter were used to investigate apparent G values in *E. pauciflora* at the three sites. The new growth model was fitted to measured DBH increments and used to estimate values of G for each tree. The growth model equation is:

$$\Delta D = f(D)\ \mathrm{T(GM)}\ \mathrm{M(MI)}\ \mathrm{P(CI)}\ \mathrm{S(BAR)} \tag{4.14}$$

with T(GM), M(MI) and P(CI) as described in the preceding sections and:

$$f(D) = \frac{D\,(1 - DH/D_{\max}H_{\max})}{274 + 3b_2D - 4b_3D^2} \tag{4.15}$$

as in BRIND. Equation 4.14 was rearranged to fit values of G to the tree data:

$$G = \frac{\Delta D}{f(D)\ \mathrm{T(GM)}\ \mathrm{M(MI)}\ \mathrm{P(CI)}\ \mathrm{S(BAR)}} \tag{4.16}$$

Potential DBH increments $f(D)$ were calculated using $D_{\max} = 91$ cm and $H_{\max} = 1829$ cm as in BRIND; T(GM) and M(MI) were calculated from the climatic estimates described in Section 4.1.4. For the S(BAR) index, SOILQ was calculated as described above to be 620 t/ha.

The increment and climatic data were available for up to 22 years, but calculation of P(CI) requires DBH measurements of all trees on the plots and these were only available from 1986. The diameters of cored trees in mid-1983 were found to be very closely correlated with their mid-1986 diameters (site 1, Pearson's r^2 0.994; site 2, r^2 0.997; site 3, r^2 0.999; all with $P < 0.001$) and regression equations between them used to estimate the mid-1983 diameters of other trees on the plots. The index P(CI) was calculated from the 1983 estimated diameters, and a close correlation was found between the 1983 and 1986 P(CI) values for each tree. Change in P(CI) during the three years was small and affected values of G in that time by only 0.3% on average.

Estimates of G were calculated for 36 cored trees based on their 1982–3, 1983–4 and 1984–5 DBH increments. There was considerable variation between years in G values, with a mean coefficient of variation of 42%. Calculated values of G were most sensitive to the measured DBH increments (ΔD, Eqn 4.16) that varied considerably during the three years, while climatic indices varied much less. In spite of variation between years, the ranking of trees' G values was consistent between the three years: the X^2 statistic calculated by Friedman's method (Sokal and Rohlf 1981) indicated that there was a significant difference ($P < 0.05$) between the trees according to their ranks. Kendall's coefficient of concordance had the value 0.81 ($P < 0.05$), which shows good correlation between the three values of G for each tree. Thus, while there was variation between years in each tree's DBH growth, some trees grew consistently faster and others consistently slower.

Figure 4.10 shows the distribution of mean values of G for the 36 trees. The overall mean value was 125, similar to the value of 135 used in BRIND. The tree with much higher calculated G than the others was the large tree at site 3 with large increments seen in Fig. 4.9(c). The estimate of G for this tree may be unrealistically large because the $f(D)$ equation in the growth model (Eqn 4.15) predicts only small increments for trees nearing $D_{\max}$. This result suggests either that the shape of the equation may not be appropriate for *E. pauciflora* or that a better value of $D_{\max}$ for this species is greater than 91 cm, or perhaps both. The distribution of G shown in Fig. 4.10 can be modelled by a normal distribution using the parameters calculated from it: $\mu = 125$ and $\sigma = 80$.

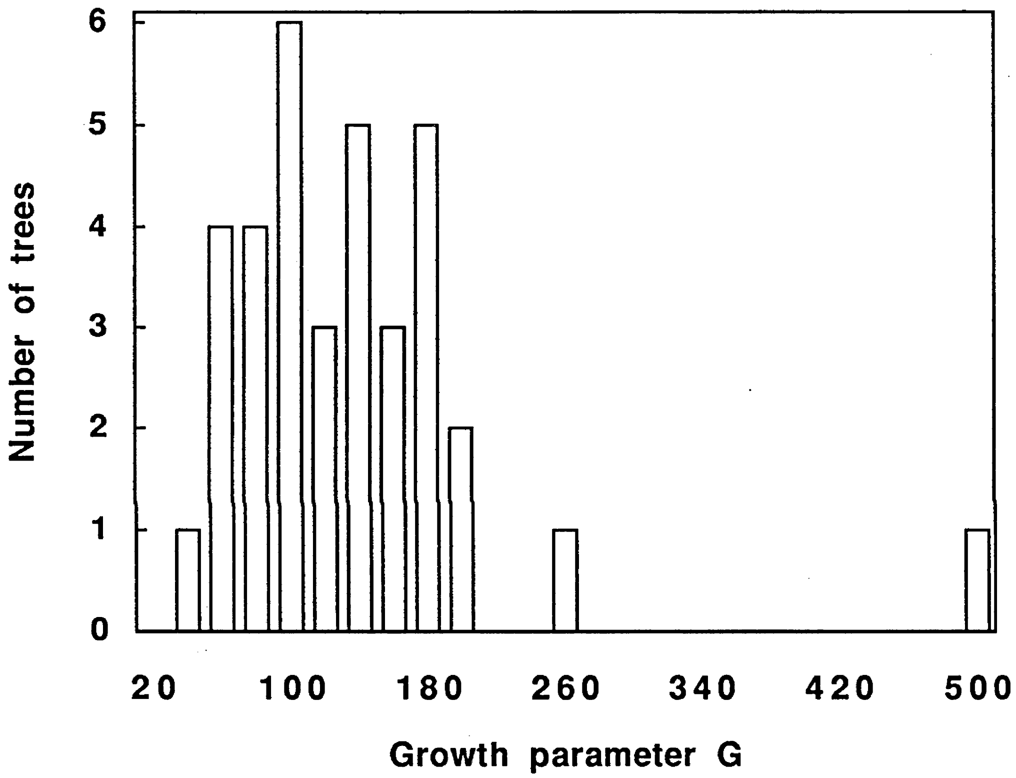

Figure 4.10: Distribution of mean values of G calculated over three years from 36 cored trees.

4.2.5 Some tests

As a test of the new growth model, simulations were run from 1976–1985 in the same manner as described in Section 4.1.5 for BRIND, and predicted 9-year DBH growth compared with growth during the same period recorded in the field data. The model was run without establishment or mortality, and the field climatic data (monthly temperature, rainfall and pan evaporation) were used in the calculation of growth-modifying indices.

The growth model requires separate values of G were required for each tree. For the cored trees, the mean of the three values calculated in Section 4.2.4 were used. For the other trees, values of G were chosen randomly from the normal distribution described above.

The competition index P(CI) is based on a smaller plot size than the shading index in BRIND, so the simulations were performed for plots with an area of *c.* 1/50 ha (and thus fewer trees than BRIND). Four separate 1/50 ha plots were simulated within each of the three field plots; the 12 plots included 32 of the cored trees. The smaller plot size necessitated adjustment of coefficients in the equation used for calculation of biomass (Table 2.2). With a plot size of 0.02 ha, the equation for biomass becomes:

$$B = 3.543 \times 10^{-3} D^{2.58} \tag{4.17}$$

where B is a tree's contribution to biomass, scaled by plot size (cf. Eqn 2.12, Table 2.2).

Figure 4.11 shows predicted and actual 9-year DBH growth for the cored trees on the 12 plots. Prediction of individual tree growth by the new growth model is much better than that by BRIND (cf. Fig. 4.3). Much of the improvement in prediction came from the use of individual G values for each tree that were

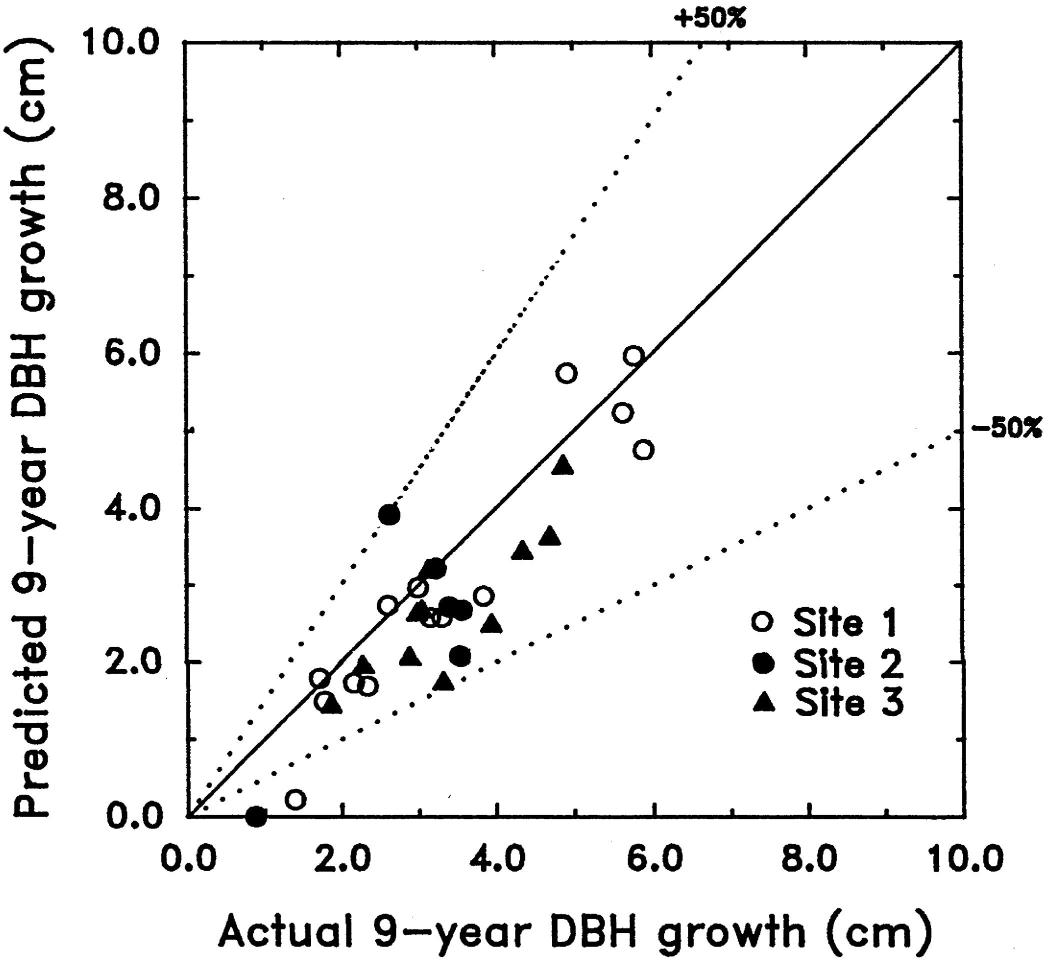

Figure 4.11: DBH growth over 9 years predicted by the new growth model vs. actual DBH growth during that period for cored trees on twelve 1/50 ha plots.

Table 4.7: Biomass estimates in tonnes per hectare on twelve 1/50 ha plots simulated by the new growth model in 1976 (beginning of simulation) and 1985 (predicted by the model); and 1986 field biomass values of the same.

		Biomass (t/ha)		
Plot	Number of trees	starting 1976	predicted 1985	calculated 1986
1a	11	48	73	79
b	24	90	156	160
c	14	41	65	73
d	7	134	138	234
2a	4	391	450	465
b	3	116	134	137
c	4	246	269	289
d	3	140	169	166
3a	6	429	485	525
b	28	140	201	187
c	25	152	120	190
d	14	316	382	395

generally lower than the fixed value in BRIND and which replicated some of the variability in growth between trees. Plot biomass values predicted by the model at the end of the 9-year period (1985) were similar to the field biomasses for 1986 (Table 4.7, cf. Table 4.5).

The 9-year simulation test is a verification of the growth model (*sensu* Mankin *et al.* 1977) because it can reproduce data very similar to those used to parameterise it. Further testing of the model requires data collected independently of the data described in this chapter—collected from other locations. Such testing may indicate that the distribution of G for *E. pauciflora* should be different (e.g., broader) to adequately model variation in the whole range of the species.

4.3 Conclusions

Examination of growth in *E. pauciflora* has suggested that the multiplicative structure of the growth model in BRIND is adequate but the terms used in it are not. New components for the growth model have been constructed on the basis of the work on *E. pauciflora* which differ significantly from those in BRIND.

The environmental growth-modifying indices of temperature and soil moisture in the new growth model do not predict cessation of growth at the ends of a species' range along those gradients. This behaviour is in keeping with my desire to change the emphasis of predicted environmental response from growth to establishment.

The maximum growth increment equation (Eqn. 4.15) is the same as in BRIND but values of G for trees of a species are drawn from a distribution, rather than being all the same. This change models variation in genetic potential and microhabitat differences evident in field data.

The index of competition P(CI) reflects effects of neighbouring trees apparent in growth of *E. pauciflora* and is a simple, one-sided index which appears to describe variation in growth as well as some more complicated indices. While P(CI) does not model an explicit mechanism of competition, it is perhaps more suitable than the r(AL) index in BRIND which models the wrong mechanism at sites (such as these) where light is not the primary resource limiting growth.

Changes to the growth submodel will have important ramifications for the remaining components of the whole model and these are taken into account in the development of a new submodel for establishment, described in the next chapter. Effects of these changes on whole model behaviour will not be known until it is assembled and tested; this is described in Chapter 8.

Chapter 5

Establishment

This chapter describes the construction of statistical regression models of species responses to environmental gradients and examination of sensitivity of BRIND to establishment rates. The regression models are to be used to predict species establishment probabilities along the gradients of temperature and soil moisture in a new establishment submodel for BRIND. Establishment will be important in the new model because growth rates of trees will vary less along environmental gradients (Chapter 4).

In Chapter 3 I showed that removal of the SPROUT subroutine resulted in improved prediction by BRIND of total plot biomass and numbers of trees. It is clear that SPROUT was causing the addition of too many trees of eucalypt species that sprout (i.e., most of them) to the simulated plot. It is not clear how sensitive model behaviour is to rates of establishment. A series of simulations were carried out with varying rates of establishment and their effects on size distribution and plot biomass examined.

5.1 Sensitivity to rate of establishment

A series of simulations were performed using BRIND without SPROUT to examine sensitivity to establishment rate. Except for the absence of SPROUT, the model was configured identically to BRIND: a value for SOILQ of 1000 t/ha was used for the S(BAR) index and the light-extinction coefficient k was set to 0.5 (Chapter 2).

The BIRTH subroutine uses three parameters to determine the number of sites for establishment during each simulated year. These parameters set limits to the number of sites for establishment: in each case, a number between one and that limit is randomly chosen for the actual number of sites. In 90% of non-fire years the limit is 4 but in the remaining 10% a limit of 30 sites is used, with the years of peak establishment chosen randomly. In fire years, a limit of 200 sites is used (Chapter 2).

Sensitivity to variation in non-fire-year limits to establishment was examined first. Sets of 50 simulations were performed at altitude classes 4 and 6 with the two non-fire limits set to: 4 and 4 (no 10-year peak); 2 and 15 (half the values in BRIND); 4 and 30 (as in BRIND); 4 and 60 (double the 10-year peak); and 8 and 60 (double the values in BRIND). The model was run without fire and sampled at 550 years after the beginning of each simulation (as in Chapter 3).

Variation in numbers of trees was largely confined to size classes less than 20 cm DBH. Figure 5.1 shows mean values and ranges of numbers of trees in the 0–10 cm and 10–20 cm size classes found in the field (F) and predicted by the model for each set of simulations (4, 4; 2, 15; 4, 30; 4, 60; 8, 60). Variation in numbers of larger trees was less, with no differences in mean numbers of trees with DBH greater than 40 cm across all sets of simulations.

Table 5.1 shows mean biomass values from each set of 50 simulations and from the field plots in each altitude class. There is an effect on predicted biomass of

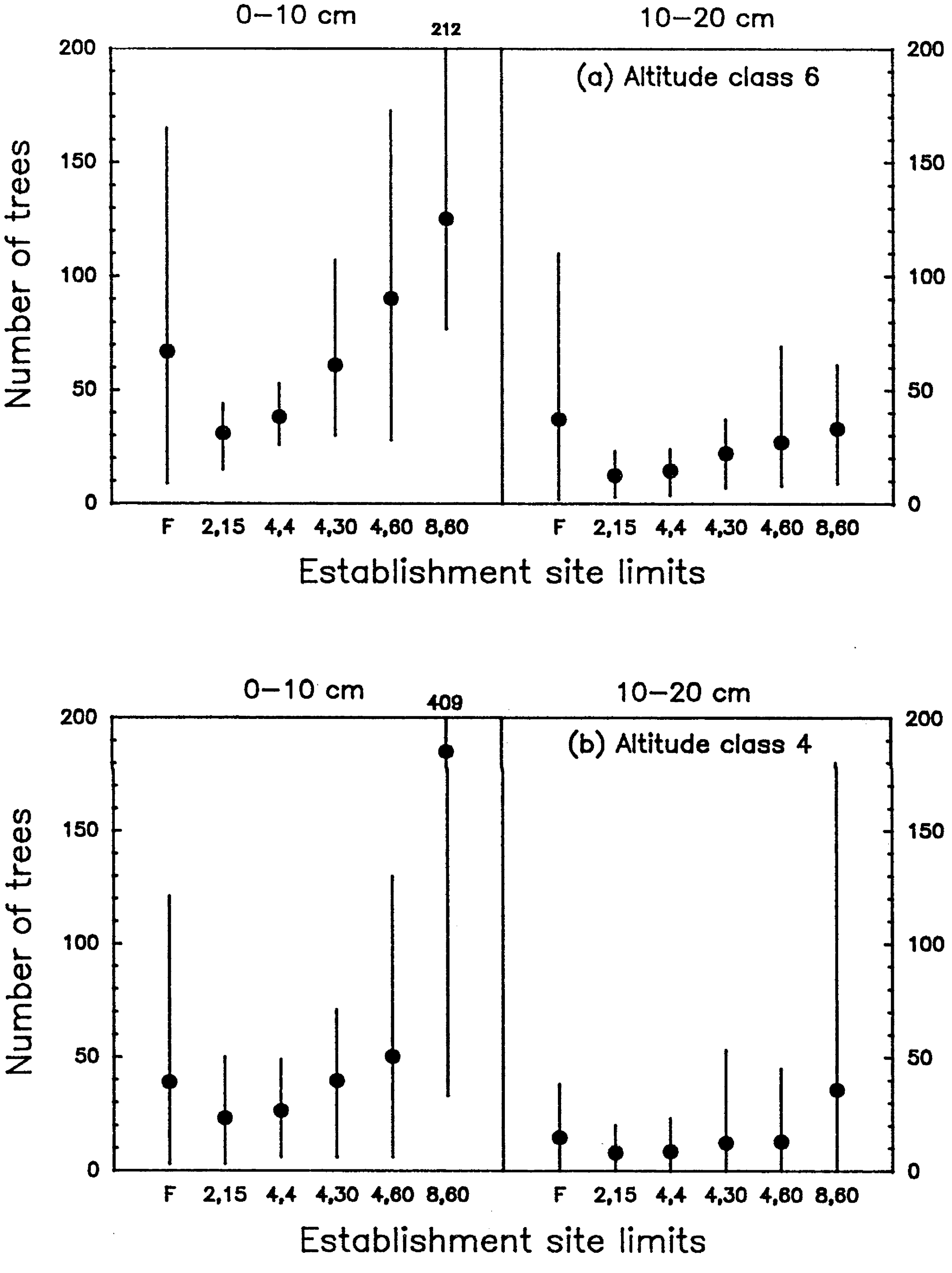

Figure 5.1: Mean and range of number of trees in 0–10 cm and 10–20 cm DBH classes in field sites (F) and predicted by BRIND without SPROUT using varying limits for establishment in years without fire (2, 15; 4, 4; 4, 30; 4, 60; 8, 60). Sets of 50 replicate simulations performed without fire in (a) altitude class 6 (> 1500 m) and (b) altitude class 4 (1101–1300 m).

Table 5.1: Mean predicted biomass values in tonnes per hectare from sets of 50 simulations with varying limits to numbers of sites for establishment (BRIND without SPROUT, no fire) in altitude classes 4 and 6. Also shown are mean field biomass values in the same altitude classes.

Altitude class	Field data	Site limits 2, 15	 4, 4	 4, 30	 4, 60	 8, 60
6	133	201	223	234	289	293
4	309	282	314	323	337	393

variation in numbers of trees establishing. When predicted numbers of small trees were closer to numbers in the field plots, biomass values were too high, especially in altitude class 6. This is because the model predicts growth of trees to sizes larger than found in the field (Section 3.4).

Sensitivity of model predictions to changes in the number of trees establishing after fire was also examined. Sets of simulations with fire (annual probability of 0.02) were performed at altitude classes 4 and 6, using BRIND without SPROUT. Non-fire-year site limits were 4 and 30 (as in BRIND) but fire-year site limits were varied: 4 (no increase with fire), 50, 100, 200 (as in BRIND) and 300. In each simulation the model was run for 500 years, then sampled at the end of the first 50-year fire-free period after that (as for the simulations described in Chapter 3). Figure 5.2 shows means and ranges of numbers of trees in the 0–10 cm and 10–20 cm DBH classes for the field sites (F) and the five simulation sets.

The number of small trees is insensitive to great variation in establishment in fire years. This phenomenon is a result of the design of the simulations, where each is sampled 50 years after fire. The 50-year sampling was chosen as a suitable method for comparison with the field data in the light of the limited fire-history data available (Chapter 3). Plot biomass increased with increasing fire-year establishment but is less sensitive than to establishment in non-fire-years (Table 5.2).

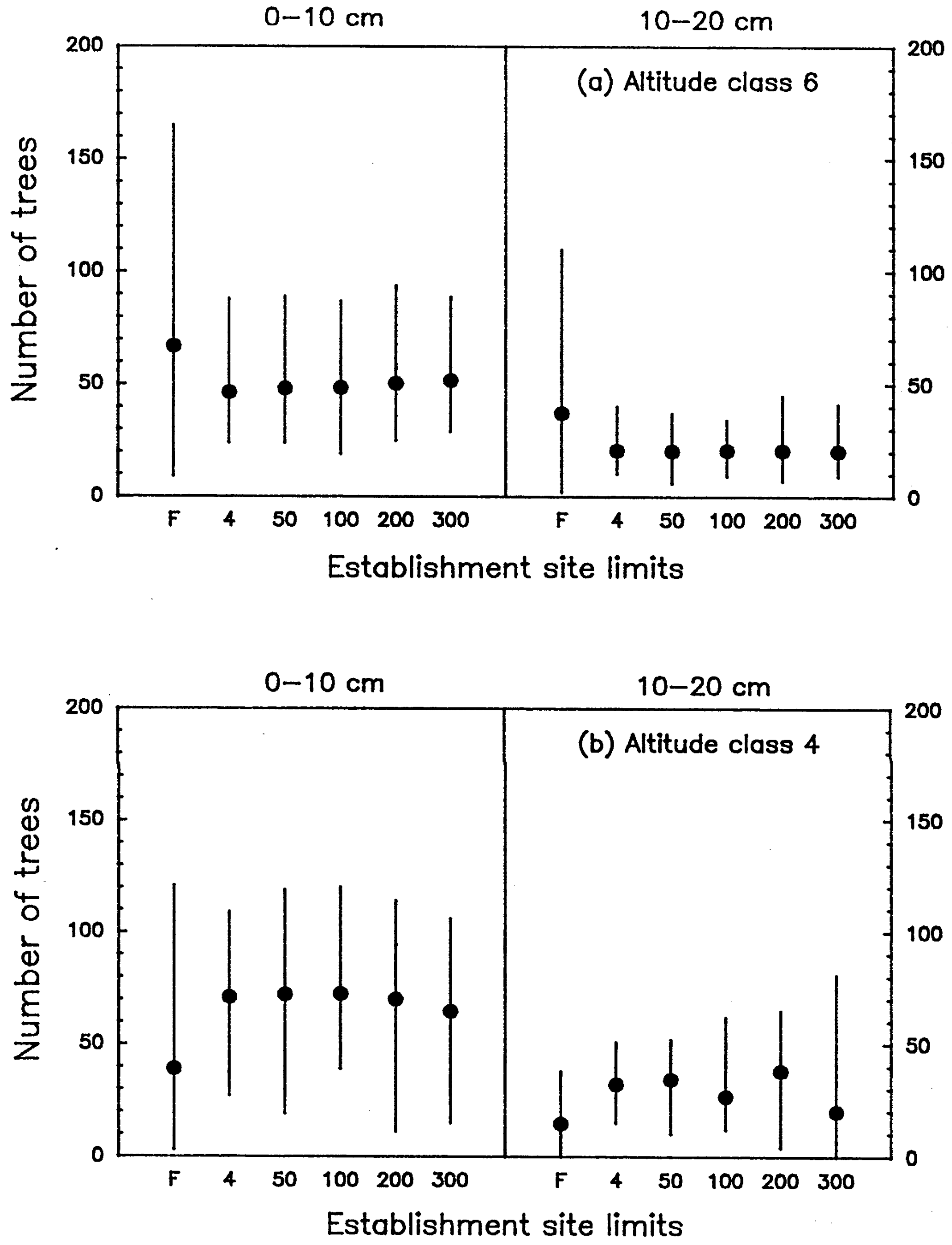

Figure 5.2: Mean and range of number of trees in 0–10 cm and 10–20 cm DBH classes in field sites (F) and predicted by BRIND without SPROUT using varying limits for establishment in years with fire (4, 50, 100, 200, 300). Sets of 50 replicate simulations performed with annual probability of fire of 0.02 in (a) altitude class 6 (> 1500 m) and (b) altitude class 4 (1101–1300 m).

Table 5.2: Mean predicted biomass values in tonnes per hectare from sets of 50 simulations with varying limits to numbers of sites for establishment after fire (BRIND without SPROUT, annual fire probability of 0.02) in altitude classes 4 and 6. Also shown are mean field biomass values in the same altitude classes.

Altitude	Field	Site limit				
class	data	4	50	100	200	300
6	133	168	197	183	190	210
4	309	149	163	178	220	220

Note that predicted biomass was lower in simulations with fire than those without (cf. Table 5.1).

In summary, predicted numbers of small trees are sensitive to rates of establishment in the immediately preceding years. Modelling occasional 'good' years with more sites for establishment appears to be important for reproducing variation in numbers found in the field. Increased establishment in fire years makes a comparatively small difference to predicted plot characteristics 50 years after fire.

5.2 Response to environmental gradients

The major addition to the establishment submodel, statistical models of species responses to environmental gradients, is discussed in the following sections.

Assuming that the observed distribution of a species largely reflects its establishment ability, a quantitative description of its distribution can be used to predict its likelihood of establishing under various environmental conditions. Austin and co-workers (Austin and Cunningham 1981; Austin *et al.* 1983, 1984; Austin 1987; Austin *et al.* 1990) have developed statistical descriptions of environmental responses of eucalypt species using generalised linear regression models (GLMs), which predict probability of occurrence of eucalypt species over a number of environmental gradients. GLMs of this type were developed to be used in the new establishment model to weight species' probabilities of establishment.

Austin *et al.* (1990) assembled data from 6080 plots throughout south-eastern New South Wales and fitted GLMs for *Eucalyptus rossii*, *E. muellerana*, *E. sieberi*, *E. pauciflora* and *E. maculata*. Environmental data for each plot were estimates of mean annual temperature, annual rainfall and radiation index calculated from broad-scale climatic surfaces (e.g., Adomeit *et al.* 1984), and geology type (see Austin *et al.* 1990 for details). They found that geology type had a significant effect on the distributions of all species and fitted separate models on the two most common geology types (sediments and granites) for each species.

5.2.1 Species models

Generalised linear models with binomial error distributions were fitted to an augmented version of the data set used by Austin *et al.* (1990) with a total of 6609 plots (additional data provided by M. P. Austin). The 6609 plots include the 106 plots described in Chapter 3 and another 100 (of 0.1 ha) from the Brindabella Range where only presence or absence of species was recorded. Fourteen species were chosen from the complete data set for model fitting. These species include the major species modelled by BRIND and other important species found in the Brindabella Range (Table 5.3).

Geology type was fitted as a factor with five levels (two of which, sediments and granites, included 82% of the 6609 plots), and temperature and moisture index were fitted as continuous variates (polynomials of degree up to 3). Temperature was found by Austin *et al.* (1990) to be the single most important environmental factor affecting species distributions; and soil moisture index, shown

Table 5.3: The 14 species for which GLMs were fitted from 6609 plots in southeastern New South Wales and coefficients for significant terms in the best models. Fitted were mean annual temperature (T, T^2, T^3), mean annual moisture stress index (MI, MI^2), simple interaction of T and M (TM), and geology type as sediments or granites.

Species	Constant	T	T^2	T^3	MI	MI^2	TM	Sed.	Gran.
Acacia melanoxylon	−28.71	3.099	−0.1489		24.65	−14.36		−0.332	−0.531
Bedfordia arborescens	−39.96	4.660	−0.1993		11.23			−0.245	−0.730
Eucalyptus dalrympleana	−43.76	5.502	−0.3302		51.15	−29.47		0.778	−1.588
Eucalyptus delegatensis	−68.7	11.43	−0.5530		35.20		−2.730	−0.779	−0.987
Eucalyptus dives	−29.73	4.786	−0.2911		14.86	−26.32	1.798	1.644	−0.213
Eucalyptus fastigata	−47.79	4.837	−0.4310		48.50	−53.38	4.594	−0.0320	0.604
Eucalyptus macrorhyncha	−14.52	−12.8	1.957	−0.07300	63.94	−24.21	−2.633	2.028	−0.046
Eucalyptus mannifera	9.02	−11.85	1.549	−0.05836	31.87	−24.41		0.945	−1.919
Eucalyptus pauciflora	−65.41	16.86	−1.523	0.04504	30.32	−9.34	−1.786	−0.935	−1.278
Eucalyptus polyanthemos	34.87	−18.57	2.271	−0.08070	−3.993			1.120	−0.635
Eucalyptus radiata	−230.2	42.51	−3.245	0.08399	128.6	−58.63	−2.560	0.688	0.905
Eucalyptus rossii	17.50	−18.55	2.462	−0.09016	40.93	−24.54	−1.317	1.180	−1.667
Eucalyptus rubida	−30.73	9.560	−0.8560	0.02322	−8.160	3.290		0.061	−0.308
Eucalyptus viminalis	−60.12	14.72	−1.192	0.03053				0.365	1.351

to be related to growth (Chapter 4), includes information of rainfall and radiation (and some of temperature). A simple interaction of temperature and MI was fitted as well, evaluated as the product of temperature and MI at each site.

Temperature and moisture index data were collated for the 6609 sites to obtain the best environmental description possible from available data. For the 206 plots from the Brindabella Range, mean annual temperature and moisture index were calculated using the climate model described in Chapter 4. These 206 plots include the 106 described in Chapter 3 and another 100 (of 0.1 ha) where only presence or absence of species was recorded. For the remaining 6403 plots, temperature estimates from the CSIRO broad-scale surfaces were used (provided with the vegetation data by M. P. Austin).

The remaining 6403 plots include the 261 Queanbeyan-Shoalhaven plots described in Chapter 4 for which MI values are available. For these plots, mean annual moisture stress index was fitted to mean annual temperature (T, °C), rainfall (R, mm) and radiation index (I, dimensionless), and the model used to estimate MI for the other 6142 plots. Because moisture index is constrained to have values between zero and one, a GLM with a binomial error distribution was fitted to estimate it. The best model for moisture index had linear terms for rainfall and radiation index ($P < 0.001$), with the formula:

$$\mathrm{LP} = 1.564 + 0.00395\ \mathrm{R} - 3.668\ \mathrm{I} \tag{5.1}$$

where LP is the linear predictor of the model. Predicted moisture index is given by the antilogit of the LP:

$$\mathrm{MI}' = \frac{e^{\mathrm{LP}}}{1 + e^{\mathrm{LP}}} \tag{5.2}$$

Figure 5.3 shows data and fitted lines for three ranges of radiation index.

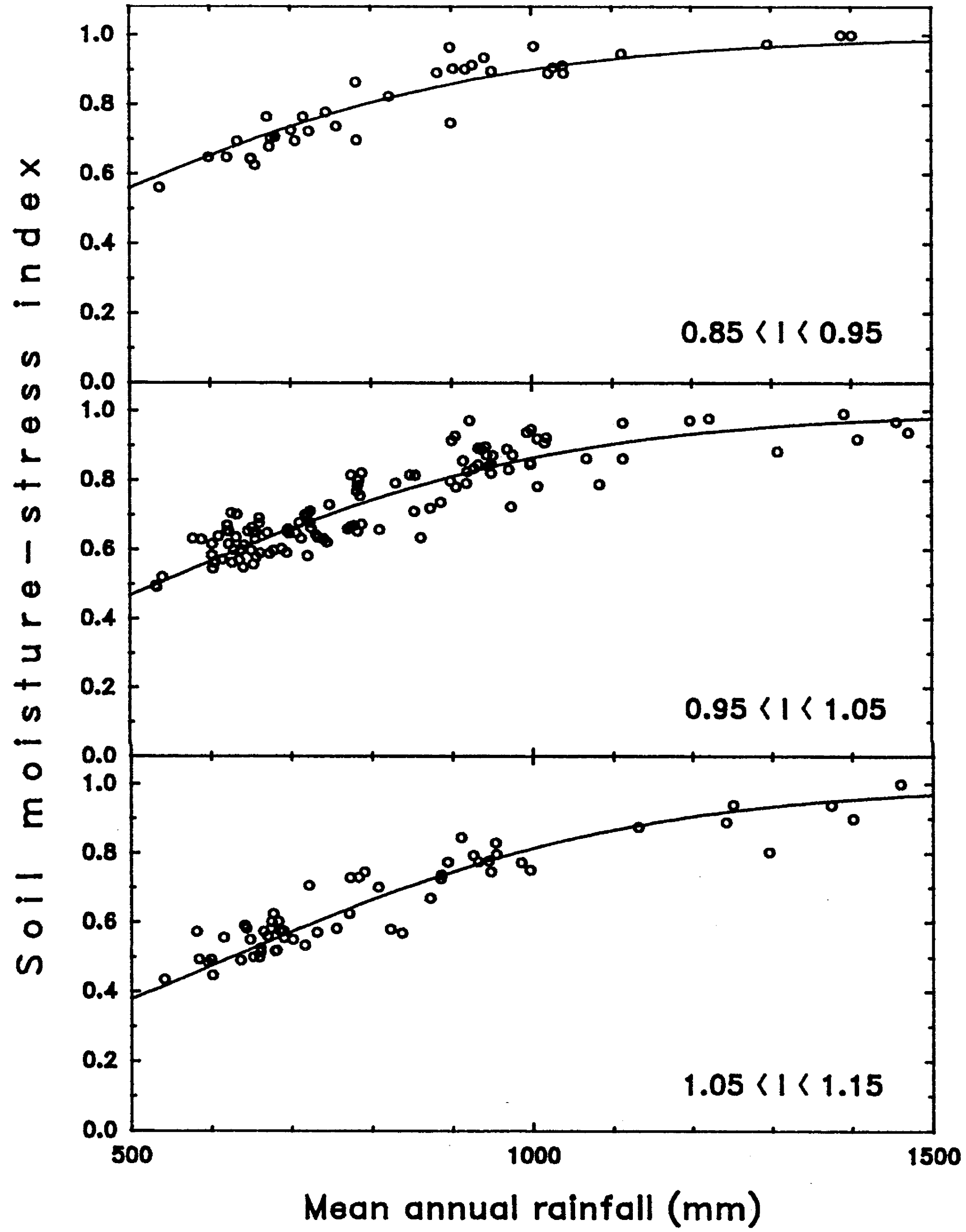

Figure 5.3: Soil moisture-stress index values from Queanbeyan-Shoalhaven data set (Gunn *et al.* 1969) plotted against mean annual rainfall for three ranges of radiation index (I). Lines are predictions of GLM shown in Eqn 5.1, for radiation index values of 0.90, 1.00 and 1.10.

5.2.2 Results

Temperature and geology type were significant terms in the chosen GLMs for all 14 species and MI was significant in all species except *E. viminalis.* A cubic term for temperature was significant in eight species GLMs and a cubic term for moisture index in four. Seven species GLMs included a significant temperature–MI interaction term.

Cubic terms for MI—where they were significant—produced nonsensical response shapes, even though they fitted the data better in terms of explanation of deviance in the GLM. In the four species with significant cubic MI terms (*E. dalrympleana*, *E.melliodora*, *E. pauciflora* and *E. rubida*), the GLM predicted a bimodal response to MI with a primary peak at a lower value of MI and an additional peak as MI approached 1.0. Bimodal response to MI was not seen in the field data for any of these species. Quartic MI terms were tested in the GLMs but found to not be significant. For the new establishment model, GLMs were chosen without cubic MI terms; the change in deviance from dropping those terms was small. Cubic temperature terms did not predict a bimodal response to temperature in any of the species GLMs where they were significant.

Table 5.3 lists coefficients of significant terms in the chosen GLMs for occurrence of the 14 species in the 6609 plots. For example, the GLM for *E. dives* is:

$$\begin{aligned} \mathrm{LP} = -29.73 + 4.786\mathrm{T} & -0.2911\mathrm{T}^2 + 14.86\mathrm{MI} - 26.32\mathrm{MI}^2 \\ & +1.798\mathrm{TM} \begin{cases} +1.644 \ \text{Sediments} \\ -0.213 \ \text{Granites} \end{cases} \end{aligned} \tag{5.3}$$

to evaluate the linear predictor (LP); probability of occurrence is given by antilogit(LP) (Eqn 5.2).

The interaction between linear temperature and moisture index terms was found to be significant in seven GLMs (Table 5.3). The interaction term changes the nature of a species' response to the environmental variates. For example, *E. dives* is more likely to occur on wetter sites at higher temperatures than at lower ones and *E. pauciflora* is more likely to occur on wetter sites at lower temperatures than at higher ones.

5.2.3 Application to BRIND

To assess how useful the GLMs for the 14 species will be for predicting probability of establishment, they were used to calculate probabilities of occurrence for the 206 field plots in the Brindabella Range out of the complete data set (6609 plots). Table 5.4 shows summaries of GLM predictions for 24 altitude–aspect cells in the Brindabella Range.

If predicted probabilities of occurrence are to be used to weight contributions to the 'seed pool' in the new establishment model, it is the relative contribution made by a species that is important. To calculate relative probabilities of occurrence, predicted probabilities were normalised by the summed probabilities of all species at each plot:

$$p_i' = \frac{p_i}{\sum p} \tag{5.4}$$

In each cell in Table 5.4, species are listed in decreasing order of probability of occurrence and sufficient listed so that the cumulative normalised probability is 0.8 or more. Also shown are mean temperatures for each altitude and mean MI values for each cell—the climate values used in GLM calculation.

Table 5.5 lists species occurrences in the 206 field plots in the Brindabella Range, summarised into the 24 altitude–aspect cells. In each cell are shown the proportions of plots in which each species occurred, normalised by the total of proportions. Species are listed in decreasing order of proportion, with sufficient

Table 5.4: Summary of predictions of probability of occurrence from GLMs fitted to 14 species in south-eastern New South Wales (6609 plots), normalised across all species in each of 24 altitude–aspect cells. Predictions made using temperature and MI values from the climate model described in Chapter 4. Sufficient species are listed in each cell so that cumulative normalised probability is at least 0.8. Also shown are mean values of temperature for the sites in each altitude class and mean values of soil moisture index (MI) for the sites in each cell (values used in calculation of the GLMs). Species names are abbreviations of those given in Table 5.3.

Altitude	Aspect							
	45°		135°		225°		315°	
1600 m	MI = 0.89		MI = 0.91		MI = 0.86		MI = 0.85	
7.4°C	E.pau	0.43	E.dal	0.44	E.pau	0.45	E.dal	0.44
	E.dal	0.41	E.pau	0.39	E.dal	0.42	E.pau	0.37
1400 m	MI = 0.87		MI = 0.91		MI = 0.85		MI = 0.83	
8.5°C	E.dal	0.37	E.dal	0.37	E.dal	0.37	E.dal	0.37
	E.pau	0.29	E.pau	0.30	E.pau	0.29	E.pau	0.30
	E.rub	0.07	E.del	0.09	E.rub	0.07	E.rub	0.08
	E.del	0.05	E.rub	0.07	E.div	0.06	E.div	0.07
	E.fas	0.05			E.fas	0.05		
1200 m	MI = 0.84		MI = 0.89		MI = 0.84		MI = 0.81	
9.5°C	E.dal	0.23	E.dal	0.31	E.dal	0.27	E.dal	0.28
	E.rad	0.19	E.rad	0.16	E.rad	0.16	E.pau	0.16
	E.pau	0.14	E.pau	0.15	E.pau	0.15	E.div	0.13
	E.fas	0.13	E.fas	0.10	E.fas	0.11	E.rad	0.13
	E.vim	0.11	E.vim	0.08	E.div	0.10	E.fas	0.09
	E.div	0.08	E.div	0.07	E.vim	0.09	E.vim	0.08
1000 m	MI = 0.82		MI = 0.86		MI = 0.84		MI = 0.79	
10.8°C	E.rad	0.20	E.rad	0.23	E.rad	0.20	E.rad	0.19
	E.div	0.17	E.dal	0.18	E.dal	0.17	E.div	0.18
	E.dal	0.14	E.div	0.13	E.div	0.17	E.dal	0.17
	E.man	0.10	E.fas	0.12	E.man	0.09	E.vim	0.08
	E.fas	0.08	E.vim	0.08	E.fas	0.08	E.man	0.07
	E.mac	0.07	E.man	0.05	E.mac	0.07	E.fas	0.07
	E.vim	0.07			E.vim	0.06	E.mac	0.05
800 m	MI = 0.78		MI = 0.85		MI = 0.80		MI = 0.75	
12.0°C	E.mac	0.18	E.mac	0.19	E.mac	0.18	E.mac	0.18
	E.man	0.17	E.man	0.17	E.man	0.17	E.man	0.18
	E.div	0.16	E.div	0.16	E.div	0.16	E.div	0.16
	E.ros	0.13	E.rad	0.14	E.ros	0.12	E.ros	0.15
	E.rad	0.11	E.ros	0.08	E.rad	0.12	E.rad	0.09
			E.vim	0.04				
660 m	MI = 0.77		MI = 0.87		MI = 0.77		MI = 0.69	
12.6°C	E.mac	0.23	E.mac	0.23	E.mac	0.23	E.mac	0.22
	E.man	0.19	E.man	0.19	E.man	0.19	E.ros	0.22
	E.ros	0.17	E.div	0.14	E.ros	0.17	E.man	0.19
	E.div	0.13	E.rad	0.11	E.div	0.13	E.div	0.11
	E.rad	0.07	E.ros	0.08	E.rad	0.06		
	E.pol	0.05	E.pol	0.06	E.pol	0.05		

Table 5.5: Proportional occurrences of 14 species in field plots in each of 24 altitude–aspect cells in the Brindabella Range on sediments and granites (206 plots). Figure shown for each species is the proportion of plots in that cell where it occurred, weighted by the total of all proportions of all species. Sufficient species are listed in each cell so that the cumulative proportion of their occurrence is at least 0.8. Numbers of plots in each cell are shown in parentheses. Species abbreviations as for Table 5.4.

Altitude	Aspect							
	0°–89°		90°–179°		180°–269°		270°–359°	
> 1500 m	(4)		(3)		(4)		(3)	
	E.pau	1.00	E.pau	1.00	E.pau	1.00	E.pau	0.75
							E.dal	0.25
1301–1500 m	(12)		(8)		(9)		(11)	
	E.pau	0.55	E.pau	0.62	E.pau	0.60	E.pau	0.55
	E.dal	0.27	A.mel	0.15	E.dal	0.27	E.dal	0.25
			E.del	0.15			A.mel	0.15
1101–1300 m	(9)		(14)		(10)		(8)	
	E.dal	0.24	E.dal	0.24	E.dal	0.30	E.dal	0.35
	A.mel	0.18	E.pau	0.24	E.del	0.26	E.pau	0.26
	E.pau	0.18	E.del	0.22	E.pau	0.26	E.div	0.26
	E.del	0.15	A.mel	0.20				
	E.rad	0.12						
901–1100 m	(11)		(12)		(9)		(8)	
	E.rad	0.21	E.rad	0.19	E.div	0.23	E.div	0.38
	E.div	0.17	E.fas	0.19	E.rad	0.23	E.man	0.24
	E.vim	0.14	Bedf	0.14	E.man	0.19	E.dal	0.19
	A.mel	0.12	E.dal	0.14	E.fas	0.16		
	E.fas	0.12	E.div	0.14				
	E.man	0.10						
701–900 m	(16)		(13)		(12)		(13)	
	E.man	0.18	E.man	0.24	E.man	0.24	E.man	0.23
	E.div	0.16	E.div	0.24	E.div	0.21	E.div	0.23
	E.mac	0.14	E.mac	0.15	E.mac	0.15	E.ros	0.16
	E.rad	0.10	A.mel	0.09	E.rad	0.15	E.mac	0.12
	E.ros	0.10	E.rad	0.09	A.mel	0.09	A.mel	0.09
	A.mel	0.08						
	E.vim	0.06						
≤ 700 m	(5)		(3)		(5)		(4)	
	E.mac	0.31	E.mac	0.25	E.mac	0.24	E.mac	0.27
	E.ros	0.23	E.man	0.25	E.man	0.24	E.ros	0.27
	E.man	0.15	E.div	0.25	E.ros	0.19	E.pol	0.18
	E.pol	0.15	E.rad	0.25	E.div	0.14	E.man	0.09

listed so that the cumulative normalised proportion is 0.8 or more. Numbers of plots in each cell are shown in parentheses.

The environmental ranges predicted by the GLMs for most species were much wider than their ranges in the field data (Tables 5.4 and 5.5). One effect of wider ranges is that in many cells more species are predicted to occur than were found in the field. With more species there were consequently lower (normalised) probabilities predicted for the species in each cell.

A number of species are predicted by the GLMs to occur at higher altitudes (lower temperatures) than in the field data. Some of these species (including *E. dalrympleana* and *E. dives*) are common in the field data at lower altitudes while others (such as *E. rubida* and *E. fastigata*) are less common in the study area but are found elsewhere in the Brindabella Range. *Eucalyptus rubida* is widespread but uncommon in the Brindabella Range (ANU Forestry 1973). *Eucalyptus fastigata* is more common in the northern part of the Brindabella Range on volcanic soils and *E. viminalis* is most commonly found in the northern part of the Range and in sheltered gullies (ANU Forestry 1973).

It is apparent for all these species that mean annual temperature does not adequately describe temperature conditions for tree establishment and growth. For example, a site in the Brindabella Range with a mean annual temperature of 7.5°C may be unsuitable for survival and growth of *E. dalrympleana* because of winter temperatures or frosts, while another site elsewhere with the same mean temperature may have milder winter conditions or higher rainfall and *E. dalrympleana* can survive and grow well.

5.2.4 Local species models

The GLMs fitted to species distribution data from a wider area are poor predictors of distributions within the Brindabella Range, using mean annual temperature and mean soil moisture index. The new establishment submodel for BRIND (in which the GLMs will be used) needs to make accurate predictions of species distributions because the growth submodel predicts less environmental effects on growth (Chapter 4), in accordance with desired changes to the structure of BRIND (Chapter 2).

More GLMs were fitted to data from only the 206 sites located in the Brindabella Range, using mean annual temperature, mean soil moisture index and geology type as before. No terms were significant ($P < 0.05$) in GLMs for four of the 14 species because of insufficient field data. They are *Acacia melanoxylon*, *Bedfordia salicina*, *Eucalyptus fastigata* and *E. rubida*.

Coefficients for the 10 GLMs fitted are shown in Table 5.6 and can be compared with Table 5.3. There are fewer terms in the new GLMs (Table 5.6): all species had temperature terms and all but one had MI terms in the old GLMs (Table 5.3) but a number are without significant MI terms and most are without geology terms in the new ones. There are also no significant temperature–MI interaction terms in the new GLMs. Fewer significant terms are largely a result of the smaller sample size (206 vs. 6609 field plots).

Two species without new GLMs, *Acacia melanoxylon* and *E. fastigata*, are important in the study area and should also be included in the new model. *Acacia melanoxylon* is a widespread understorey species and *E. fastigata* is not widespread but personal observations showed that it usually dominates on sites where it occurs. For these two species, GLMs fitted to the large data set were included with the new GLMs for the other ten.

Table 5.6: Coefficients of significant terms in the best GLMs fitted to species occurrence data from 206 plots in the Brindabella Range. Fitted were mean annual temperature (T, T^2, T^3), mean annual moisture stress index (MI, MI^2) and geology type as sediments or granites. Also shown is the number of plots in which each species occurred. Compare with Table 5.3 (GLMs fitted to 6609 plots throughout south-eastern New South Wales).

Species	Constant	T	T^2	T^3	MI	MI^2	Sed.	Gran.	No.
Eucalyptus dalrympleana	−89.29	21.82	−1.150		−13.59				69
Eucalyptus delegatensis	−437.6	13.99	−0.7561		768.4	−395.6			29
Eucalyptus dives	−86.67	15.00	−0.6595				2.722	1.386	83
Eucalyptus macrorhyncha	−50.42	4.205							46
Eucalyptus mannifera	−113.1	18.41	−0.7515				1.834	−5.776	68
Eucalyptus pauciflora	46.23	−4.764							83
Eucalyptus polyanthemos	−12.99	2.004			−15.74				12
Eucalyptus radiata	−987.8	251.0	−21.44	0.6077	14.32				57
Eucalyptus rossii	−8.371	2.395			−24.61				33
Eucalyptus viminalis	−86.14	15.66	−0.7198						22

Table 5.7 shows predicted probabilities of occurrence of the 12 species, normalised across all species in 24 altitude–aspect cells in the Brindabella Range, and can be compared with Tables 5.4 and 5.5. The new GLMs predict ranges of species occurrence much closer to the field ranges than the old GLMs. *Eucalyptus delegatensis* does not appear in Table 5.7 because its predicted probability is low (maximum value 0.08).

Low maximum predicted probabilities like that for *E. delegatensis* result from low numbers of occurrences in the field data and because the two environmental dimensions used (temperature and soil moisture index) are inadequate to describe the occurrence of a species with any more certainty. Austin *et al.* (1990) discussed the problem of incomplete environmental descriptions and stated that topographic position (slope, ridge, gully etc.) or disturbance history can have important effects on probability of finding a species. They also stated that environmental variates that have a more direct effect on tree survival and growth processes (e.g., seasonal temperature instead of mean annual temperature) may need to be used to predict occurrence with more certainty.

5.3 Conclusions

Rates of establishment in BRIND appear to be adequate and predict plausible numbers of small trees. Predicted plot biomass is too high when numbers of establishment sites are correct, but it is also affected by a number of other model components (Chapter 3).

The GLMs fitted to the large data set were poor predictors of distribution in the Brindabella Range because the two environmental dimensions of mean annual temperature and soil moisture index are inadequate to predict occurrence. The GLMs from the Brindabella Range data fit better, but predict low maximum

Table 5.7: Summary of predictions of probability of occurrence from GLMs fitted to 12 species in the Brindabella Range (206 plots), normalised across all species in 24 altitude–aspect cells in the Brindabella Range. Predictions made using temperature and MI values from the climate model described in Chapter 4. Sufficient species are listed in each cell so that cumulative normalised probability is at least 0.8. Species names are abbreviations of those given in Table 5.3.

Altitude	Aspect							
	45°		135°		225°		315°	
1600 m	E.pau	0.91	E.pau	0.91	E.pau	0.89	E.pau	0.89
1400 m	E.pau	0.50	E.pau	0.50	E.pau	0.49	E.pau	0.48
	E.dal	0.39	E.dal	0.34	E.dal	0.41	E.dal	0.42
1200 m	E.dal	0.39	E.dal	0.37	E.dal	0.39	E.dal	0.40
	E.pau	0.30	E.pau	0.30	E.pau	0.30	E.pau	0.30
	E.div	0.11	E.div	0.11	E.div	0.11	E.div	0.11
			E.fas	0.11				
1000 m	E.div	0.25	E.div	0.24	E.div	0.25	E.dal	0.27
	E.dal	0.25	E.dal	0.20	E.dal	0.23	E.div	0.25
	E.man	0.16	E.rad	0.18	E.man	0.16	E.man	0.16
	E.rad	0.13	E.man	0.15	E.rad	0.15	E.rad	0.10
	E.vim	0.09	E.vim	0.09	E.vim	0.09	E.vim	0.09
800 m	E.man	0.23	E.man	0.26	E.man	0.24	E.ros	0.24
	E.ros	0.22	E.div	0.24	E.div	0.22	E.man	0.22
	E.div	0.22	E.mac	0.16	E.ros	0.20	E.div	0.21
	E.mac	0.15	E.ros	0.12	E.mac	0.15	E.mac	0.14
			E.rad	0.10				
660 m	E.ros	0.24	E.mac	0.26	E.ros	0.24	E.ros	0.24
	E.mac	0.23	E.man	0.23	E.mac	0.23	E.mac	0.22
	E.man	0.20	E.ros	0.17	E.man	0.20	E.man	0.19
	E.div	0.15	E.div	0.17	E.div	0.15	E.pol	0.19

probabilities for some species. It is not certain how predictions of establishment by the GLMs will interact with those of the new growth submodel described in Chapter 4. Chapter 8 describes tests of the new model with both components.

Chapter 6

Mortality

This chapter describes an investigation of the effects of reducing rates of random background mortality from the high levels in BRIND (Chapter 2) to more realistic levels. Sensitivity of suppression mortality (which increased when background mortality was reduced) to variation in selected model parameters was also examined.

Examination of mortality described here begins with a discussion of theoretical models of tree mortality that provide a basis for desired predictions by forest models.

6.1 Mortality rate curves

It has been observed that patterns of life-history phenomena in trees (including death) correlate better with size than with age, and that size and age of trees are often not closely correlated (White 1980; Hughes 1984). Mortality rates of trees should thus be expressed in relation to tree size rather than tree age but are still usually expressed as annual rates of (size-related) mortality.

In a recent review, Harcombe (1987) concluded that the most common pattern of mortality in trees is described by a concave curve, where the annual mortality rate of small trees is high, falling away rapidly to a low level for trees larger than a certain size. Smaller trees are subject to suppression which results in reduced growth and higher mortality; intermediate-sized trees may be suppressed but do not show increased mortality as a result. Their rate of mortality is low and remains apparently constant ('background' mortality): it is usually attributed to random effects which are essentially unpredictable.

There have also been suggestions that 'U'-shaped mortality rate curves are appropriate to trees, where very large individuals show higher mortality than intermediate-sized trees. The higher rate of mortality in large trees is associated with slower growth. It is assumed to result from increased susceptibility to disease and wind damage resulting from senescence, and from accumulated damage by insects or fire. Evidence for U-shaped mortality rate curves is equivocal, and Harcombe (1987) argued that such curves should not be expected on grounds that there is no evolutionary advantage in increased mortality with large size. Nevertheless, U-shaped mortality curves for trees seem intuitively likely: mortality of intermediate-sized trees (which are not killed by suppression) must be very low and there are limits to the sizes trees can actually attain, so the size-dependent mortality rate of very large trees is expected to increase. Convincing evidence for existence of U-shaped curves is difficult to find because of the rarity of mortality in large trees. While there may not be any evolutionary advantage in trees being subject to higher mortality rates when large, such mortality may be unavoidable for physical reasons.

Harcombe (1987) summarised information about the size of trees at which the initial high rate of mortality falls to the lower rate of background mortality, i.e., the size above which trees are no longer subject to mortality from suppression. The sizes he gave are all between 10 and 20 cm DBH. Harcombe also described mortality rates of trees in the 'flatter' part of the curve and noted some consistent differences between species. Annual mortality rates presented varied between and within studies (rates calculated in some studies included mortality of suppressed trees) but for unsuppressed trees ranged from approximately 0.0009 to 0.02.

Mortality in BRIND is modelled in three components: random, 'background' mortality with low probability (size-independent mortality); increased mortality from fire damage; and increased mortality from reduced DBH growth (suppression and old age) (Chapter 2).

Annual probabilities of size-independent mortality in BRIND for eucalypts range from 0.011 (*E. dalrympleana*, 400 years) to 0.036 (*E. pauciflora*, 125), which I argued in Chapter 2 to be unrealistically high. It is not certain what is a realistic rate of background mortality for eucalypts in the Brindabella Range. From the range of rates given by Harcombe (1987), a background annual mortality rate of *c.* 0.005 appears to be more realistic. A figure of 0.005 would result in predicted loss of 5.0% of trees (of all sizes) on average in any 10-year period. In the three plots used in the tree-ring work described in Chapter 4, 160 trees are certain to have survived at least 10 years but it is not certain how many of the 17 dead trees found on those plots died during that period (the dead trees varied in size from 0.5 to 45 cm DBH). Background mortality at an annual rate of 0.005 would be expected to reduce 168 trees to 160 in 10 years.

Calculation of rates of mortality from field data requires recording of tree size and status (dead/alive) of individual trees over a period of time, often with a 5- or 10-year period between records. These data are used to construct a static life table for the trees, where the numbers in each size class are recorded according to three fates: remaining in the same size class during that period; growing to the next size class; and death (Harcombe 1987). Annual mortality information was collected from BRIND by tabulating the fates of all trees each simulated year, according to size class. Every year, each tree was recorded as having been killed by fire, suppression or random size-independent mortality, having grown into the next size-class, or having stayed in the same one. Figures tabulated over a number of years allow mean annual mortality rates during that time to be calculated.

Simulations were performed with BRIND to investigate mortality rate curves predicted by the model. The version of BRIND used had the SPROUT subroutine removed because it contributes to unrealistic prediction of biomass and establishment (Chapter 3). The $\mathrm{Age}_{\mathrm{max}}$ parameter for *Eucalyptus pauciflora* was set to 250 years: double the value used by Shugart and Noble (1981) in BRIND and a better estimate (Banks 1982). In each simulation, after a 1000-year lead time, the model was run for a 2000-year period during which mortality information for each species was tabulated as described above. Simulations were performed without fire in altitude classes 4 and 6 (Chapter 3); plots were dominated by *E. delegatensis* and *E. pauciflora* respectively.

Figure 6.1 shows annual mortality in 5 cm DBH classes of trees of *E. pauciflora* in altitude class 6 (Fig. 6.1(a)) and *E. delegatensis* in altitude class 4 (Fig. 6.1(b)). Shaded portions of the bars show the contribution of size-independent mortality, while open portions show that from slow growth (suppression in small trees, growth reduction with size in large trees). The dotted lines show probabilities of size-independent mortality for each species: the expected proportion of trees dying in that way.

6.2 Size-independent mortality

The patterns of mortality shown in Fig. 6.1 conform approximately to those described by Harcombe (1987) except that the rate of size-independent mortality

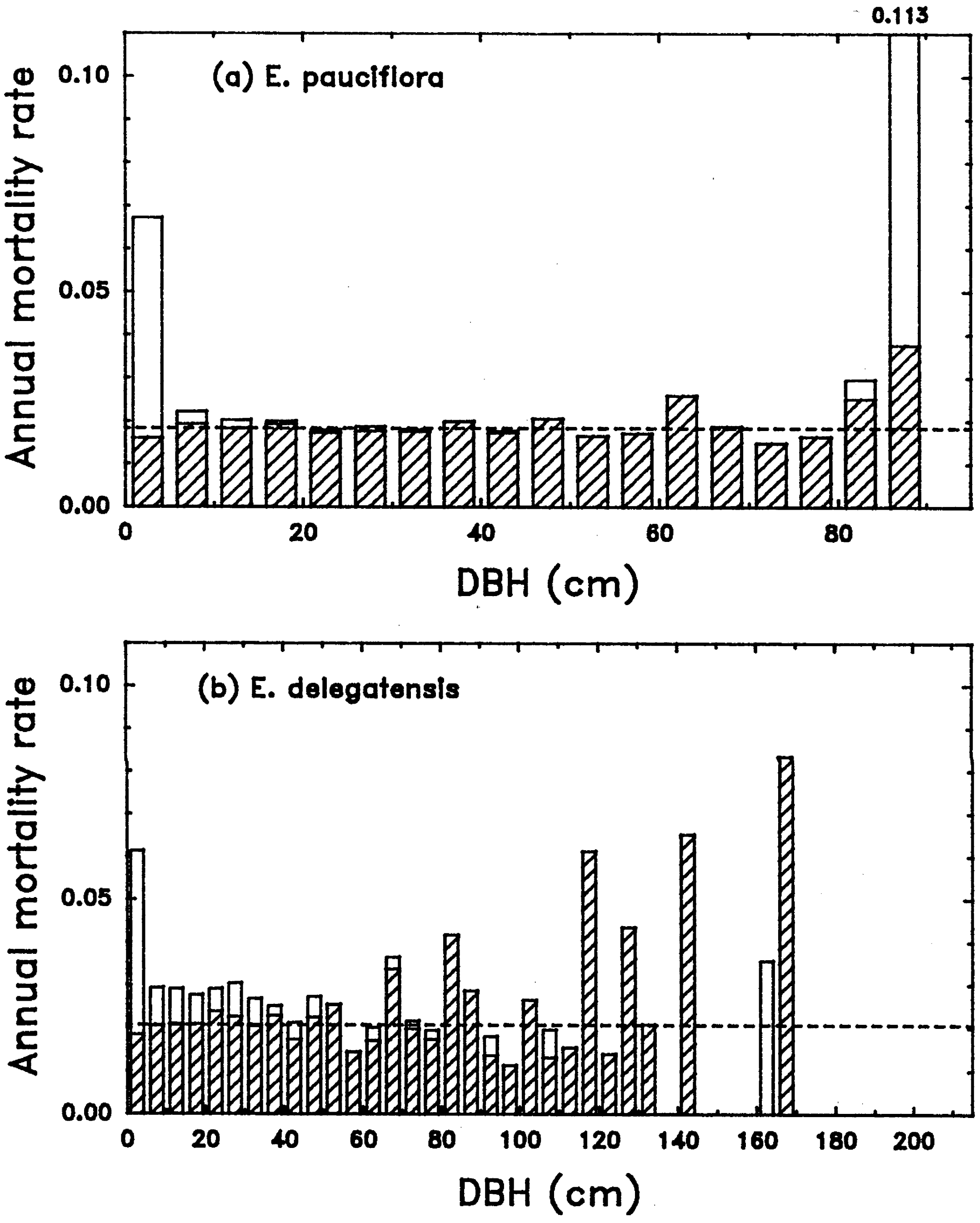

Figure 6.1: Annual mortality rates in 5 cm DBH classes predicted by BRIND without the SPROUT subroutine in 2000-year simulations without fire. Mortality in (a) *Eucalyptus pauciflora* from simulations in altitude class 6 (Chapter 3) and (b) *E. delegatensis* from simulations in class 4. Shaded bars show size-independent mortality, open bars suppression mortality. Dashed line shows expected mortality rate from size-independent mortality.

is unrealistically high. Calculation of size-independent mortality was investigated with the aim of improving the model's prediction of background mortality rates. In BRIND annual size-independent mortality is calculated from each species' Age_{max} parameter using:

$$p = 1 - \exp(\ -4.605/\text{Age}_{\text{max}}\ )$$

or

$$p = 1 - (0.01)^{1/\text{Age}_{\text{max}}} \tag{6.1}$$

based on the idea that 1% of trees are expected to survive to Age_{max}. If size-independent mortality were the only cause of death then approximately 1% of trees would reach Age_{max}. Because of death from suppression (and fire in BRIND), the actual percentage of trees surviving to Age_{max} is lower: 0.24% of all trees in altitude class 4 and 0.34% of all trees in class 6 survived to Age_{max} in the simulations whose results are shown in Fig. 6.1.

As an experiment, mortality rates were calculated using an expected proportion of trees surviving to Age_{max} of 20% (i.e., with 0.2 instead of 0.01 in Eqn 6.1). For *E. pauciflora* with Age_{max} set to 250 years the annual rate is 0.0064 and for *E. delegatensis* (220 years) it is 0.0073. These annual rates are similar to those described by Harcombe (1987) and appear to be consistent with the information available from the tree-ring study plots discussed above. The final survival of trees to Age_{max} will be less than 20% because of the additional suppression mortality.

Figure 6.2 shows annual mortality in size classes from the simulations with size-independent mortality calculated on the basis of 20% survival to Age_{max}. Overall rates of size-independent mortality are more realistic, as expected, but there was higher suppression mortality in both species. Suppression mortality

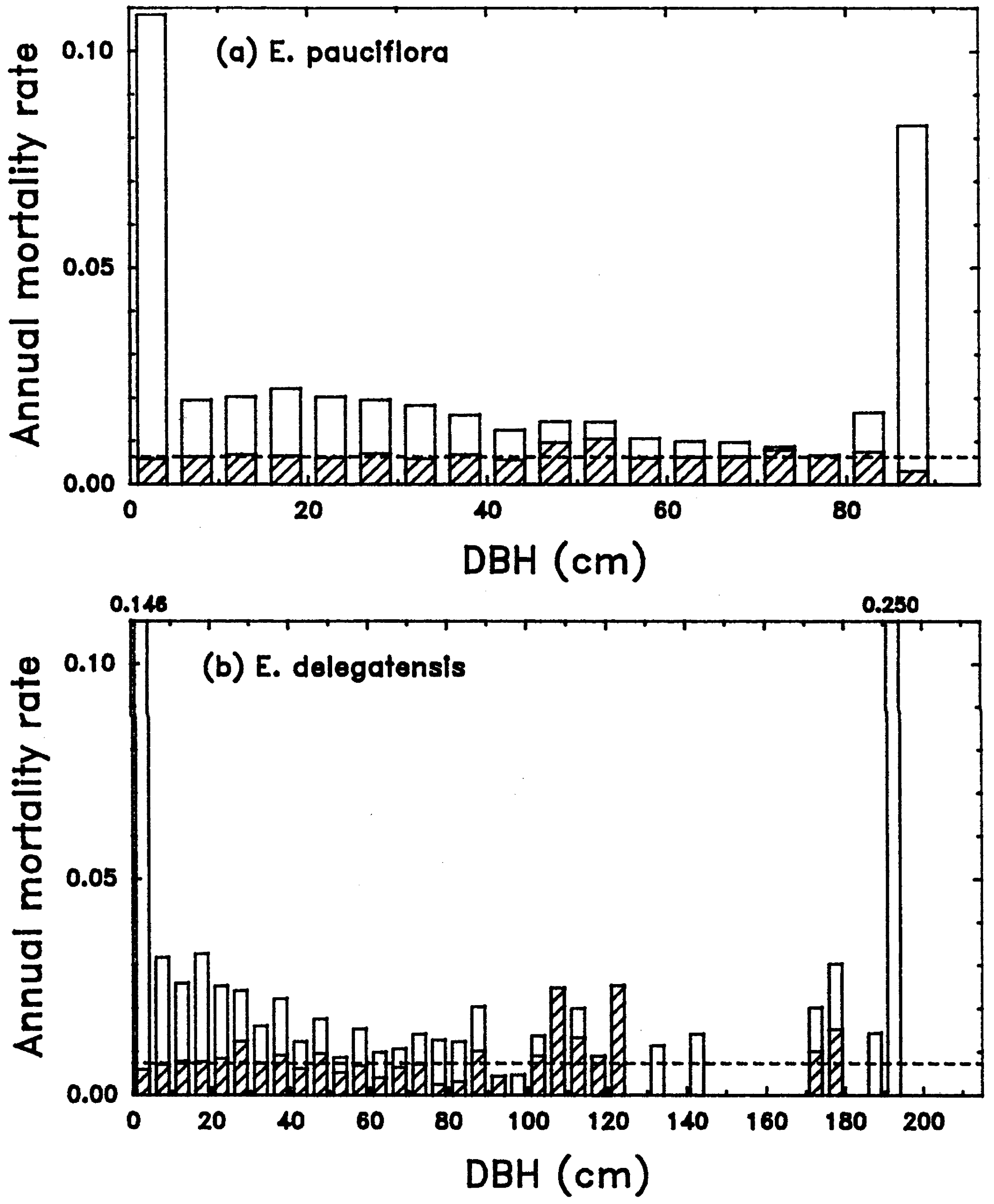

Figure 6.2: Annual mortality rates in 5 cm DBH classes predicted by the model as for Fig. 6.1 but with size-independent mortality probabilities based on 20% final survival to Age_{max} (see text), in 2000-year simulations without fire. Mortality in (a) *Eucalyptus pauciflora* from simulations in altitude class 6 and (b) *E. delegantensis* from simulations in class 4. Shaded bars show size-independent mortality. open bars suppression mortality. Dashed line shows expected mortality rate from size-independent mortality.

increased to the extent that the pattern of overall mortality was similar (Fig. 6.2, cf. Fig. 6.1). The final percentage of trees surviving to $\mathrm{Age}_{\mathrm{max}}$ increased in altitude class 4 (from 0.24% to 1.40%) but did not change in class 6 (0.34% and 0.30%).

Suppression was higher because more trees survived to grow bigger, resulting in the prediction of more biomass. Mean values over 2000 years increased from 353 to 390 t/ha for the *E. pauciflora* plots in altitude class 6 and from 326 to 395 t/ha for the *E. delegatensis* plots at class 4. Mean biomass of field plots in the two altitude classes was 133 t/ha in class 6 and 309 t/ha in class 4.

6.3 Suppression mortality

Suppression mortality predicted by BRIND in the simulations shown in Fig. 6.2 appears too high: trees which have attained a certain size may be suppressed but are not killed (Harcombe 1987). Harcombe collated evidence that trees will survive suppression once they have a DBH of 15–20 cm. Jacobs (1955) described an individual of *E. delegatensis* of DBH 48 cm as being suppressed but still alive. Banks (1982) also found chronic suppression in *E. pauciflora* to be common in long-lived trees.

In the tree-ring data described in Chapter 4 there was evidence of suppression in trees of a range of sizes: some of the cored trees had growth increments of less than 0.1 cm in some years, especially at site 2 (Fig. 4.9). There were also six trees which were cored but excluded from the analyses because they had many very narrow rings with unclear boundaries—indicative of chronic suppression—but were not dead. The DBH of five of these trees was between 6 and 12 cm while that of the sixth—which had been hollowed out by fire—was 54.5 cm. It is clear that some trees in these stands were surviving chronic suppression, even when quite small.

In spite of these data, there is evidence that thinning in even-aged stands results in mortality of intermediate-sized trees. The yield table of unthinned *E. delegatensis* published by Borough *et al.* (1978) indicates that stocking is still reduced by density-dependent mortality when trees are as large as 40–50 cm DBH. Their data suggest total annual mortality rates of *c.* 0.063 for 20 cm DBH trees and of *c.* 0.025 for 45 cm DBH trees. These rates are higher than would be expected from density-independent causes alone. Borough *et al.* also provided similar data for unthinned stands of other eucalypt species.

It appears that forest structure has a bearing on mortality of intermediate-sized trees. While these trees may survive suppression in mixed-age (mixed-size) stands, in even-aged (even-sized) stands with many competitors of similar size they may not be able to survive suppression.

A series of simulations were performed to investigate how variation in certain model parameters affected patterns of suppression mortality predicted by the model. Parameters investigated were numbers of trees establishing, growth rates of the species and intensity of shading competition.

To investigate the effect of establishment numbers on suppression mortality rates, simulations were performed as in the previous section, with lower size-independent mortality rates (Fig. 6.2) but with the limits of sites for establishment halved, from 4 and 30 to 2 and 15 (Chapter 5). Figure 6.3 shows mortality rates resulting from those simulations and it can be seen that halving the rate of establishment had very little effect on them: mortality of the smallest trees was lower—fewer trees entered the plot, so fewer were killed; and predicted mean biomass was slightly lower (Table 6.1). These results are consistent with previous observations that variation in establishment numbers mainly affects numbers of trees with DBH less than 40 cm and is only a partial contributor to excessive biomass (Chapter 5).

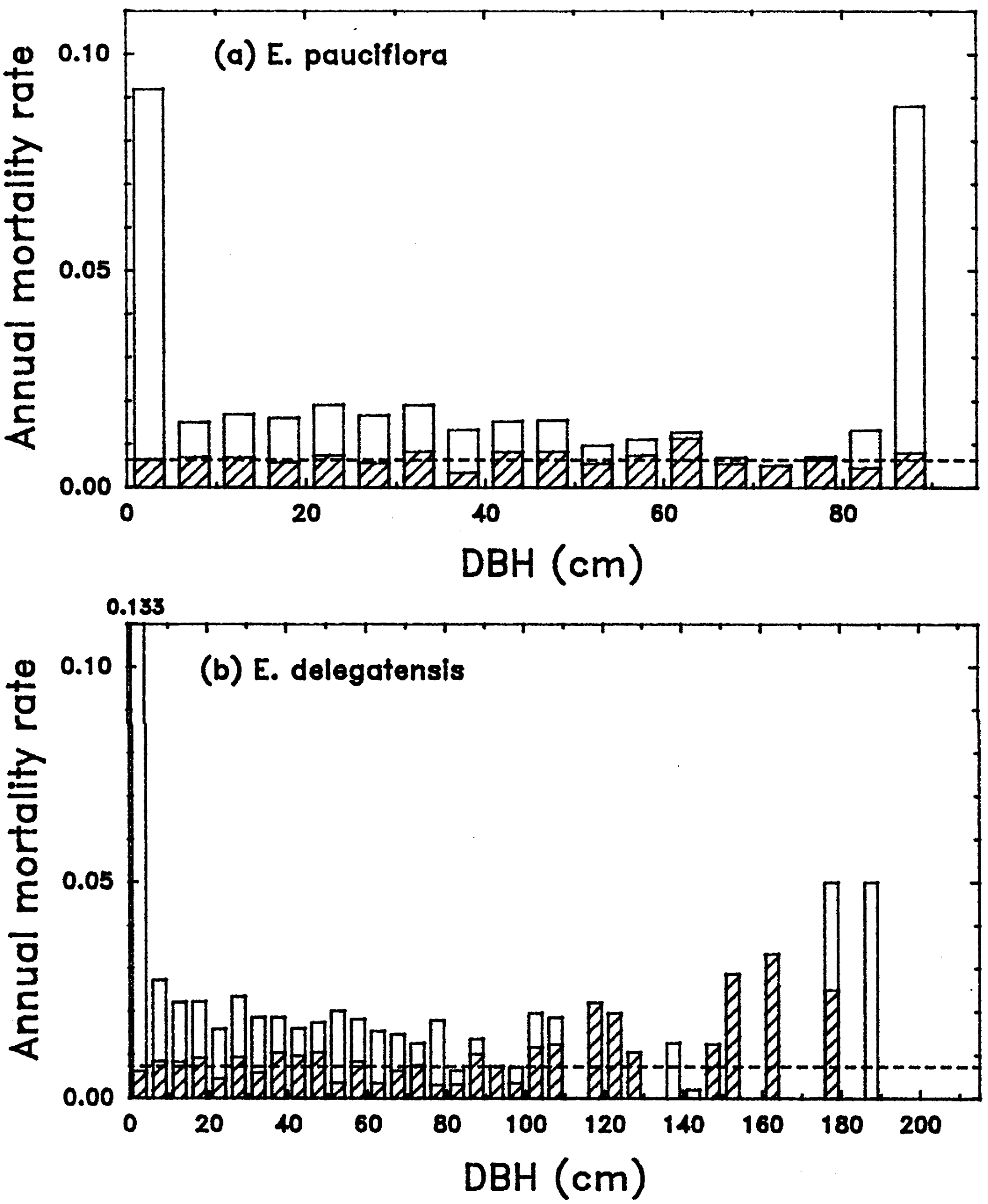

Figure 6.3: Annual mortality rates in 5 cm DBH classes predicted by the model as for Fig. 6.2 but with limits to numbers of sites for establishment reduced from 4 and 30 to 2 and 15 (see text), in 2000-year simulations without fire. Mortality in (a) *Eucalyptus pauciflora* from simulations in altitude class 6 and (b) *E. delegatensis* from simulations in class 4. Shaded bars show size-independent mortality, open bars suppression mortality. Dashed line shows expected mortality rate from size-independent mortality.

Table 6.1: Mean biomass values in tonnes per hectare in field plots and predicted by BRIND without SPROUT and without fire, for altitude classes 4 and 6. Simulations were run for 2000 years after 1000-year lead times. $\mathrm{Age}_{\max}$ for *Eucalyptus pauciflora* was set to 250 years (cf. 125 years in BRIND). Also shown is the number of the figure which shows mortality rates in the simulations.

Source	Altitude class 4	Altitude class 6	Figure
Field plots	309	133	
BRIND without SPROUT	326	353	6.1
Size-independent mortality from 20% survival (see text)	395	390	6.2
20% survival and:			
Establishment site limits halved	377	376	6.3
Species growth rates halved	311	296	6.4
Increased competition	407	314	6.5

Simulations were also performed with the growth rate parameters (G) of all the species halved. Figure 6.4 shows predicted mortality rates: suppression mortality was slightly reduced in intermediate–large trees and the largest size attained by trees was less (cf. Fig. 6.2). The largest size reached by trees in the model is a function not only of $D_{\max}$ but also G and mortality rate (cf. Figs 6.1 and 6.4). Mean biomass was lower (Table 6.1) but still in excess of biomass on the field plots.

In Chapter 3 I showed that plot biomass is sensitive to variation in the light extinction coefficient used to calculate shading, k. Larger values of k result in stronger suppression of smaller trees by larger ones and effect a reduction in plot biomass (Figs 3.11 and 3.12). Figure 6.5 shows mortality rates from simulations performed with k set to 0.7. Mortality of trees with DBH less than 5 cm increased

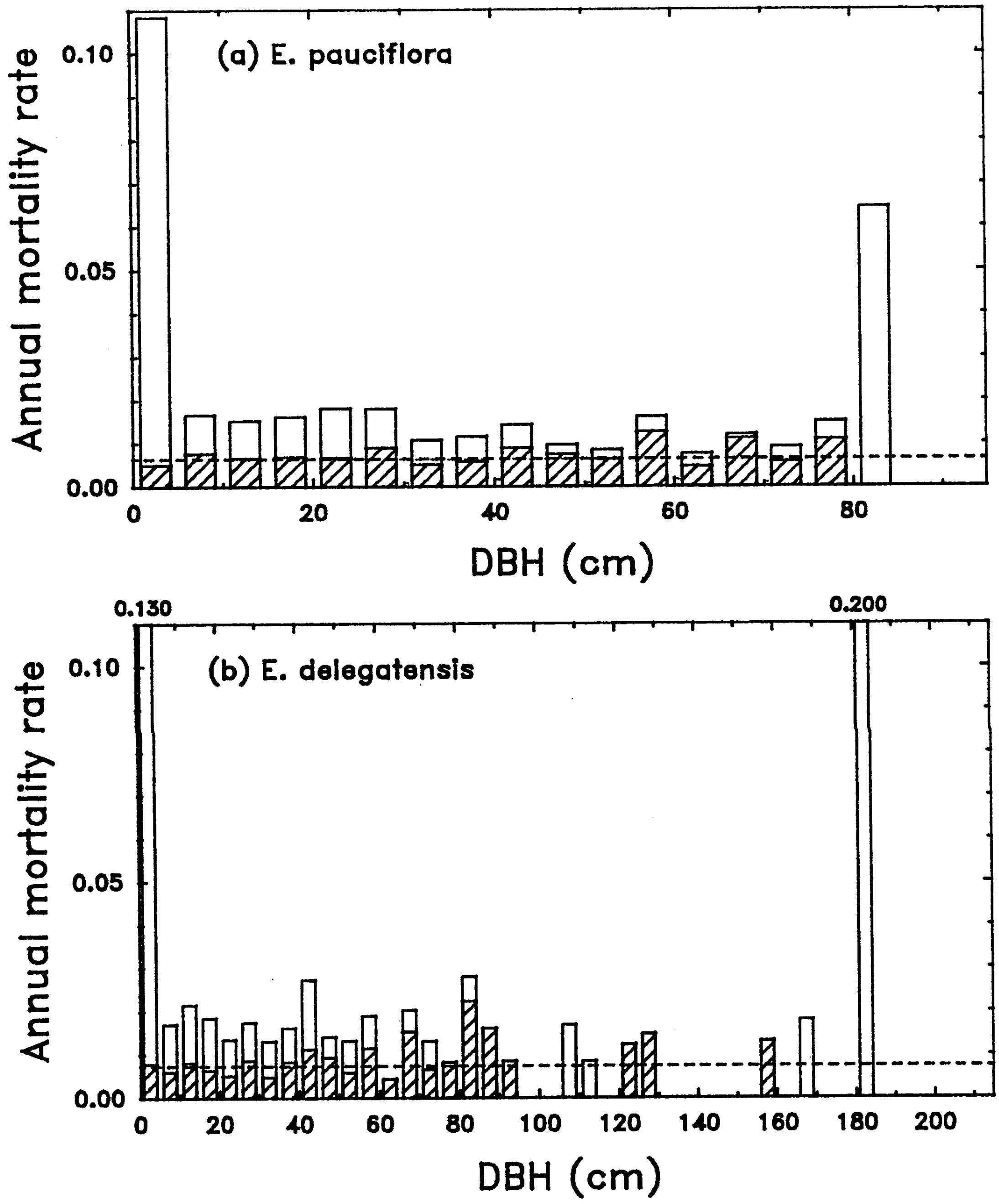

Figure 6.4: Annual mortality rates in 5 cm DBH classes predicted by the model as for Fig. 6.2 but with the growth parameters (G) of all species halved (see text), in 2000-year simulations without fire. Mortality in (a) *Eucalyptus pauciflora* from simulations in altitude class 6 and (b) *E. delegantensis* from simulations in class 4. Shaded bars show size-independent mortality, open bars suppression mortality. Dashed line shows expected mortality rate from size-independent mortality.

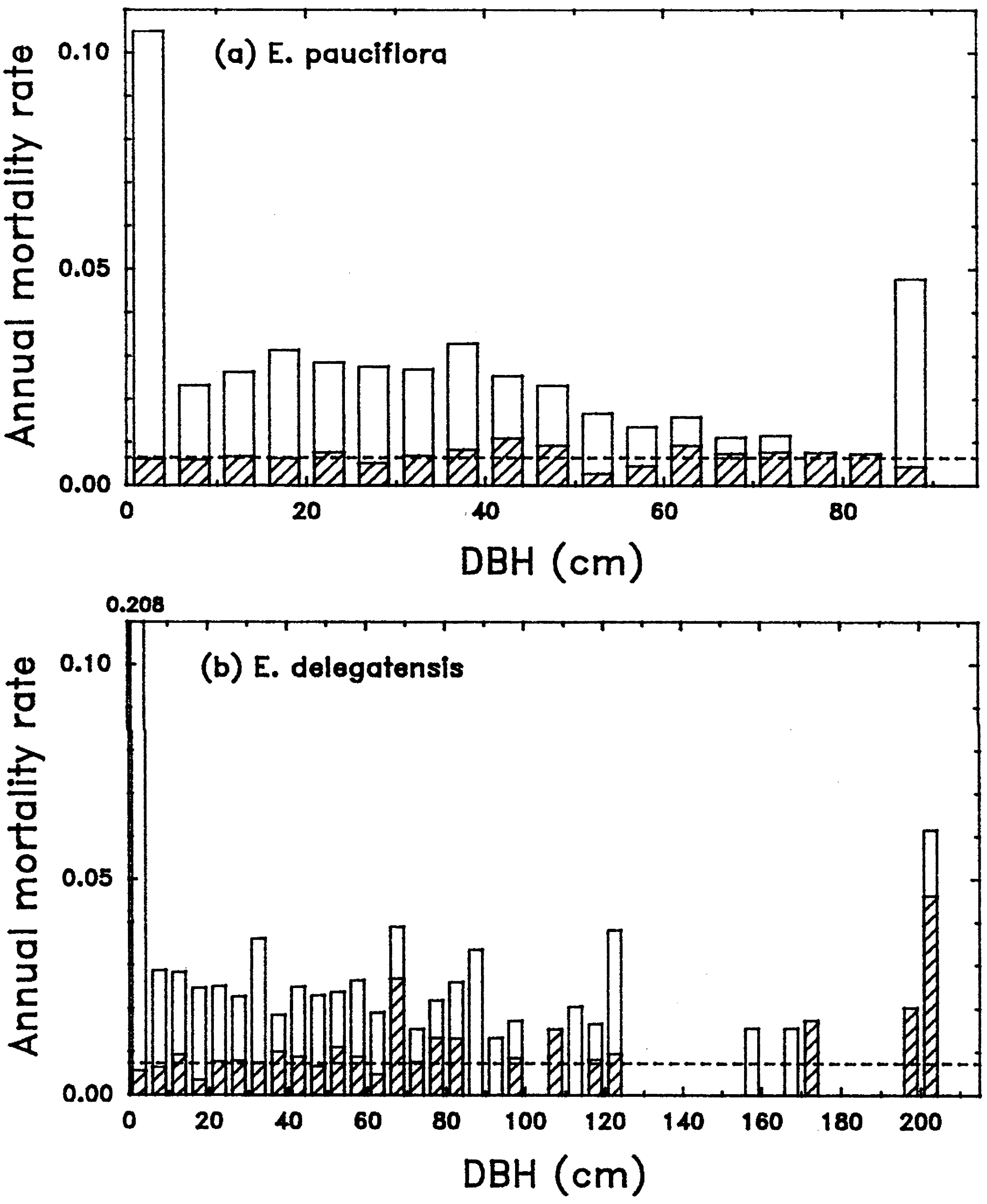

Figure 6.5: Annual mortality rates in 5 cm DBH classes predicted by the model as for Fig. 6.2 but with the light extinction coefficient k increased from 0.5 to 0.7 (see text), in 2000-year simulations without fire. Mortality in (a) *Eucalyptus pauciflora* from simulations in altitude class 6 and (b) *E. delegantensis* from simulations in class 4. Shaded bars show size-independent mortality, open bars suppression mortality. Dashed line shows expected mortality rate from size-independent mortality.

in altitude class 4 but did not change in class 6, while mortality of intermediate-sized trees in both altitude classes increased (cf. Fig. 6.2). Mean plot biomass decreased in class 6 but increased slightly in class 4 (Table 6.1, relative to simulations with $k = 0.5$).

It appears strange that suppression of intermediate-sized trees of *E. pauciflora* should increase with increased k, while that of the smallest trees remain unchanged. The explanation is that trees with DBH less than 5 cm were already completely suppressed with $k = 0.5$, so increasing k made no difference to their rate of mortality from that cause. The smallest trees of *E. delegatensis* were not completely suppressed with $k = 0.5$, so the increase in k had an effect on them.

Increased mortality of intermediate-sized trees of both species with increased k is a product of the dynamic (gap-phase replacement) behaviour of the model. For much of the time the plot is dominated by one or a few large trees: the only other trees are very small, are suppressed and few escape to grow bigger. After the loss of a large tree, leaf area index on the plot falls and the small trees are released from suppression. These trees form an even-sized cohort (all growing at similar rates) whose growth is not checked until leaf area index on the plot builds up again and shading becomes significant. At this stage self-thinning of the cohort begins, most of it taking place when the trees have DBH greater than 20 cm. This model behaviour appears to mimic that of the even-aged stands described by Borough *et al.* (1978) where self-thinning occurs in trees up to quite large sizes.

6.4 Conclusions

Reduction of rates of size-independent mortality in BRIND to more realistic levels had two effects on model behaviour. More biomass is predicted as a simple result of lower overall mortality: more trees survive to bigger sizes. Suppression mortality increased when size-independent mortality was less so that mortality overall rates of intermediate-sized trees (especially those with DBH $<$ 40 cm) were similar.

Suppression mortality rates were not greatly affected by establishment or growth rates were halved. When the intensity of competition (the effect of larger trees on smaller ones) was increased mortality of medium-sized trees increased as a result. Increased competition had little effect on mortality rates in the smallest trees which were already suppressed.

Higher rates of mortality from suppression in medium-sized trees should not be expected according to the data presented by Harcombe (1987) but self-thinning data from eucalypt forests suggest that higher rates occur in even-aged stands. It appears that stand structure affects mortality rates of these trees, so that it is higher when trees are all of similar size. The model behaves in this way: after disturbance, cohorts of trees grow up and thin together at larger sizes (20–40 cm DBH) than individual trees growing under the canopy of one or few dominants.

Chapter 7

Fire response

This chapter describes the development of a new submodel of tree response to fire from data collected after a wildfire in eucalypt forests. In Chapter 2 I described the fire-response model in BRIND and criticised its treatment of fire response as being only immediate or delayed mortality, without modelling damage to surviving trees or sprouting. BRIND treats sprouting only as a method of replacing dead trees, not as a response to fire. The new fire-response submodel was developed for fire-resistant eucalypt species to predict mortality and re-sprouting responses to fire.

The fire-behaviour component of BRIND, which predicts excessive fire intensity values (Chapter 2), was adjusted to calculate realistic intensities so the new response model could be tested.

7.1 The Gudgenby fire

In January and February 1983, a fire burnt extensive areas of the Gudgenby region of the southern Australian Capital Territory. A total area of *c.* 36 000 ha of forest was burnt in the altitudinal range *c.* 1000 m–1700 m. There was much

Table 7.1: Site information of the ten sites where fire-response data were collected at Gudgenby. Grid references are to the nearest 50 m on the Australian Metric Grid. Estimated scorch heights are shown: those in italics are from sites where all sites where all trees were fully scorched and are equal to the height of the tallest tree (Fig. 7.2).

Grid E (m)	Grid N (m)	Altitude (m)	Aspect (°)	Slope (°)	Estimated scorch height (m)
680850	6039600	1160	245	13	*12.6*
680650	6041550	1180	105	20	13.0
680400	6035100	1380	320	9	17.5
676950	6036650	1150	330	3	*24.0*
678650	6040700	1055	340	7	12.4
680900	6042500	1080	300	12	9.2
676450	6046450	1070	80	12	*28.0*
676850	6054100	955	80	18	17.6
676750	6054600	980	95	18	7.9
681350	6032100	1310	260	7	9.4

variation in fire intensity throughout the burnt area, ranging from leaf scorch of only the lowest leaves on trees, to apparent 'fire-storm' conditions where all material with a diameter less than *c.* 5 cm had been removed.

In May–July 1983 ten sites were selected in the burnt area to sample the range of fire intensities, judged visually from the extent of leaf scorch (Table 7.1). At each site, a sample plot 20 m wide and 50–75 m long was marked out. All trees judged to have been alive before the fire were tagged with numbered, galvanised iron tags. A total of 1111 trees were tagged, mostly of four species: *Eucalyptus dives* (336 trees), *E. rubida* (227), *E. viminalis* (107) and *E. pauciflora* (397). In April–May 1985 the sites were revisited and the extent of recovery by the trees recorded.

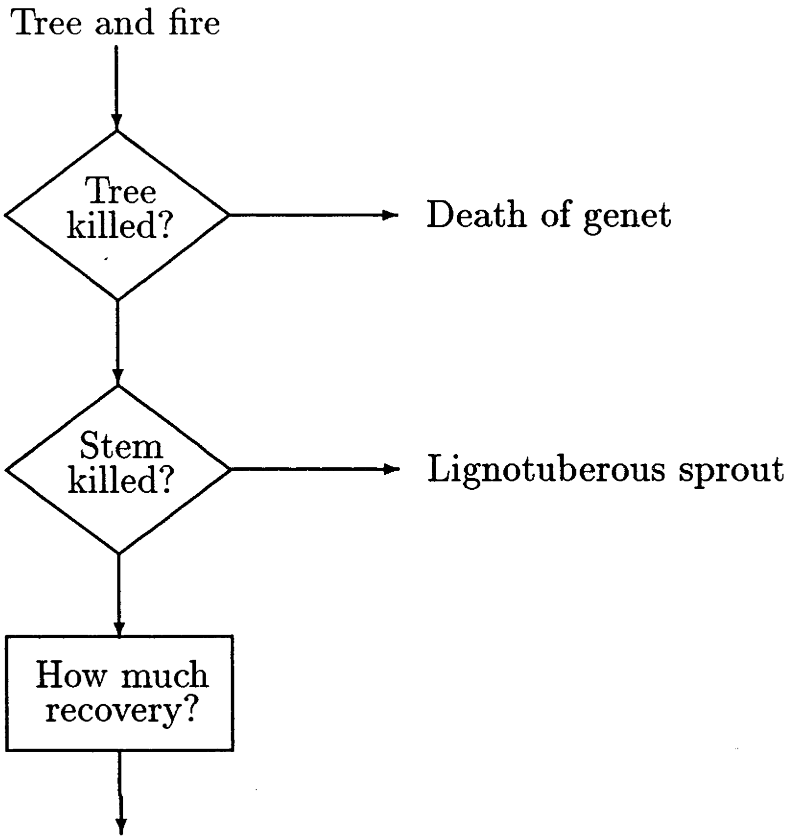

Figure 7.1: Flowchart of three-stage fire response model for BRIND. The sequence of operations will be carried out for all trees on the simulated plot.

7.2 Three-stage response model

The four common species recorded at Gudgenby are all 'sprouting' species (Gill 1981b)—they can survive complete canopy scorch and show sprouting from epicormic buds in response to fire. Responses to fire recorded at Gudgenby suggested a 3-stage model for sprouting species whose structure is illustrated by the flowchart in Fig. 7.1.

The first stage of the response model is concerned with mortality: whether or not a tree dies as a direct result of being burnt. The second stage, for the surviving trees, predicts whether or not their above-ground parts survive the fire: the stem can be killed but the tree continue to grow from lignotuberous sprouts. The third stage estimates the amount of damage sustained by trees whose stems have survived the fire. These trees grow new leaves from epicormic buds (Gill 1981b).

The new model, like that in BRIND, will be applied to each simulated tree when a fire occurs.

7.3 Data from Gudgenby

Data were collected from Gudgenby to describe size of the trees and to estimate intensity of the fire at each site. Gill (1981a) suggested that extent of bark loss may be used as an indicator of fire intensity but the Gudgenby sites were first visited after the fire so pre-fire bark depths were not available. Intensity was estimated from height of highest leaf-scorch on the trees.

In 1983, the following data were recorded for each tree:

1. DBH overbark in cm;
2. Height in metres, measured trigonometrically for a sub-sample of trees and estimated for the remainder based on the measurements;
3. Estimated scorch height: if a tree was fully scorched, scorch height was recorded as unknown but greater than tree height.

In 1985, these data were recorded for each tree:

1. Status (dead/alive);
2. Estimated height of the tip of the tallest living shoot.

7.3.1 Scorch height

Only 125 of the 1111 trees were not fully scorched and at three sites all trees were fully scorched (Table 7.1). An attempt was made to estimate an average fire intensity for each of the sites to be applied to all trees, including those that were fully scorched. While some information is lost by averaging scorch heights over each site, the estimated values are more appropriately applied to BRIND, where scorch is assumed constant throughout the simulated plot.

Figure 7.2 describes the algorithm used, which minimises least-squared differences between individual tree scorch heights and the estimated site value. Because the error for a fully-scorched tree is zero when the tree is shorter than the estimated site scorch (tree *d*, Fig. 7.2) these fully-scorched trees contributed nothing to the estimates, so at the three sites where all trees were fully scorched S was evaluated as the height of the tallest tree. At these three sites scorch may be significantly underestimated, with the possible result that no effect of scorch height will be apparent in the response of the trees when fire intensity was actually important.

7.3.2 Vigour index

Considerable variation in height within most DBH classes was apparent for trees at Gudgenby (Fig. 7.3). This variation may indicate tree vigour, so a 'vigour index' was constructed based on the height of a tree relative to an expected maximum height for a tree of its DBH in that species. Height was divided by expected maximum height for each tree to give an index value between zero and one.

For calculation of vigour index, curves of maximum expected height were calculated for each of the four common species recorded at Gudgenby, using a

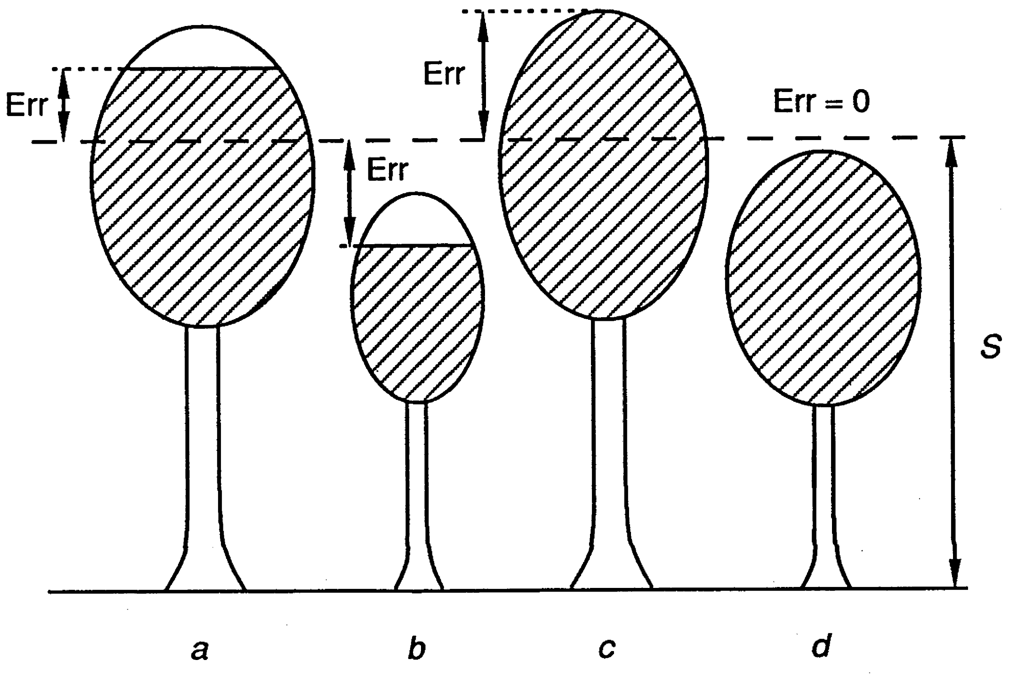

Figure 7.2: Diagram illustrating method for estimation of 'true' scorch height (S) at a site. A value of S is calculated to minimise $\sum \text{Err}^2$, where Err is defined as: the difference between an estimate of S and recorded scorch height of a tree which is not fully scorched (trees *a* and *b*); the difference between an estimate of S and the height of a fully-scorched tree greater than S (tree *c*); or zero for a fully-scorched tree shorter than the estimate of S (tree *d*).

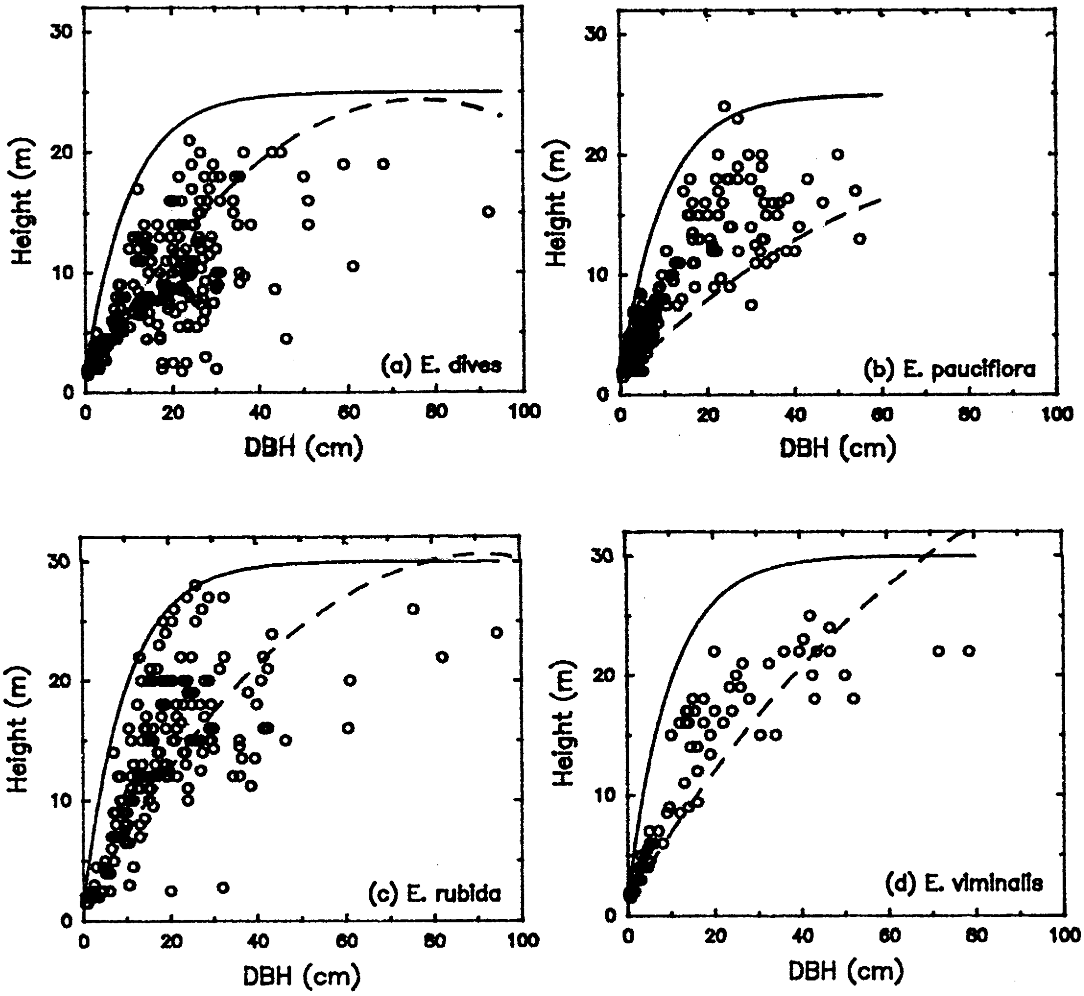

Figure 7.3: Height vs. DBH of trees recorded at Gudgenby in the four major species. Shown are exponential 'envelope' curves calculated using equations of the form shown in Eqn 7.1 (solid lines) and the DBH–height curves used in BRIND (dashed lines).

negative exponential equation of the form:

$$H_e = H_{\max} - e^{b-cD} \tag{7.1}$$

to give expected height H_e from $H_{\max}$, the maximum height parameter for a species (Chapter 2); b sets the intercept of the height axis for zero DBH at 137 cm and is therefore given by $b = \ln(H_{\max} - 137)$ and c determines the initial slope of the line. Equation 7.1 has the same form as the DBH–height equation preferred by Leemans and Prentice (1987) to the quadratic equation (Eqn 2.4) in their gap model FORSKA.

Figure 7.3 shows DBH and height data for the four common species at Gudgenby and curves for each species from Eqn 7.1. Values of $H_{\max}$ used were 25, 25, 30 and 30 m for *E. dives*, *E. pauciflora*, *E. rubida* and *E. viminalis*, respectively. The $H_{\max}$ values for *E. dives* and *E. rubida* are similar to values for those species used by BRIND, but that for *E. pauciflora* greater (BRIND uses 18.3 m) and that for *E. viminalis* less (36.6 m in BRIND). According to Boland *et al.* (1984)—a later edition of the source used for many species parameter values in BRIND (Hall *et al.* 1970, see Shugart and Noble 1981)—it is uncommon for *E. pauciflora* to grow taller than 20 m and with a straight bole but some of the trees found at Gudgenby belong to this category. Boland *et al.* (1984) also describe *E. viminalis* as commonly growing to heights of 30–50 m but at Gudgenby was only found growing to *c.* 25 m (Fig. 7.3). A value for c of 0.1 was found to give a suitable 'envelope' curve for each of the four species at Gudgenby. Vigour index calculated from these curves had values between zero and one for most trees, with three exceptions whose values were slightly greater than one. Inaccuracy in height estimation could account for these three discrepancies.

7.3.3 Summary of response

Table 7.2 summarises tree mortality results at Gudgenby. Of the four most common species, mortality was highest in *E. viminalis* and lowest in *E. pauciflora*, but their 95% confidence intervals overlap.

Table 7.3 summarises stem death results for surviving trees at Gudgenby. Stem death was highest in *E. pauciflora* and lowest in *E. rubida* with significant differences between the species. The higher stem death in *E. pauciflora* is in agreement with Banks' (1982) finding of a high rate of mortality of above-ground tissue in this species after fire. Banks attributed this behaviour of *E. pauciflora* to its being a member of the ash group of eucalypts which includes the fire-sensitive (but non-lignotuberous) species *E. regnans* and *E. delegatensis.*

Height to the tips of the tallest living shoots on trees with surviving stems was used as an indication of recovery after fire. Many trees were observed to have full crown recovery and be as tall as before the fire, or taller. With the exception of *E. viminalis* with 29%, 50–60% of all trees showed full canopy recovery.

7.4 Statistical models

Generalised linear models (GLMs) were fitted to the tree data to be used as predictive equations of fire response. The first two stages of the fire-response model have binary response variates—trees and stems were either killed or they survived—so GLMs with binomial error distribution and logit link function were fitted. Height recovery from fire was expressed as a proportion of pre-fire height and a binomial GLM fitted to predict that as well.

Table 7.2: Mortality of trees recorded at Gudgenby. Shown are total numbers of trees, proportion dead two years after the fire and 95% confidence intervals for mortality in the four most common species (normal approximation to binomial).

Species	Total number	Mortality proportion	95% CI
E. dalrympleana	33	0.030	
E. dives	336	0.080	0.051–0.109
E. pauciflora	397	0.053	0.031–0.075
E. radiata	3	0.000	
E. rubida	227	0.062	0.046–0.078
E. viminalis	107	0.121	0.060–0.183
unknown	8	0.750	
Total	1111	0.075	

Table 7.3: Stem death in surviving trees recorded at Gudgenby. Shown are total numbers of trees, proportion with dead stems and 95% confidence intervals for stem death in the five most common species (normal approximation to binomial).

Species	Total number	Stem death proportion	95% CI
E. dalrympleana	32	0.469	0.381–0.557
E. dives	309	0.479	0.451–0.507
E. pauciflora	376	0.819	0.799–0.839
E. radiata	3	0.333	
E. rubida	213	0.371	0.338–0.404
E. viminalis	94	0.553	0.502–0.604
unknown	2	0.500	
Total	1028	0.587	

7.4.1 Mortality

A GLM was fitted to predict mortality, using data from four species: DBH, height, scorch height, vigour index and a species factor term were used. Vigour index was found to explain most deviance ($P < 0.001$), then DBH ($P < 0.01$), the interaction between vigour index and DBH ($P < 0.05$) and species factor ($P < 0.05$). The GLM without the species factor term was chosen for use in the fire-response model. Even though the species term was significant, only one species (*E. viminalis*, with higher mortality) was predicted to be significantly different from any others, but this species had the fewest trees of the four. Equation 7.2 shows the equation for the linear predictor of the chosen GLM:

$$\mathrm{LP} = -2.518 - 1.715v + 0.1185D - 0.1603vD \tag{7.2}$$

where LP is the linear predictor, v is vigour index, D is DBH and vD is the product of v and D which expresses the interaction of the two.

Figure 7.4 shows (a) proportion mortality in DBH and vigour index classes and (b) contours of probability of mortality predicted by the GLM for all species with DBH, vigour and interaction terms (Eqn 7.2). This model predicts higher mortality for less vigorous trees and for larger ones. From Fig. 7.4 it appears that the mortality model may predict too much mortality of larger, more vigorous trees—this is partly a problem of fitting a regression model to a rare event.

7.4.2 Stem death

Initially, a GLM predicting stem death was fitted for all species combined. The species factor term was significant and predicted significant differences in stem death between the species, so separate models were then fitted for each of the

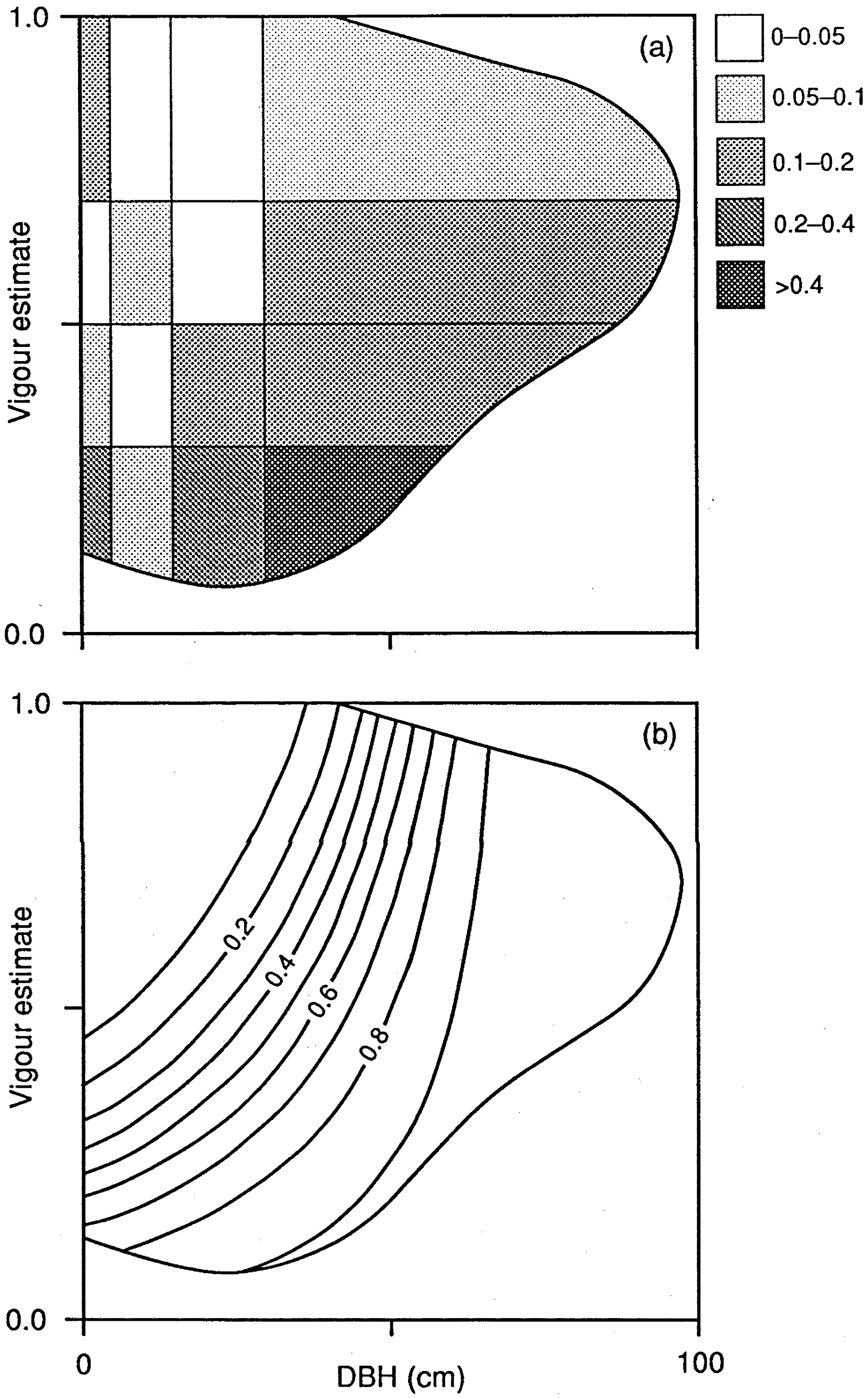

Figure 7.4: (a) Proportion mortality in 16 DBH–vigour estimate classes of the field data from Gudgenby and (b) Contour plot of predicted probability of mortality from GLM for all species with DBH–vigour interaction term. Curved line in both plots shows approximate boundary of DBH and vigour values found in the trees.

Table 7.4: Coefficients in final GLMs for predicting stem death in the four major species at Gudgenby. Two models are shown for *E. pauciflora*: (a) with and (b) without scorch height term.

Species	No. of trees	Constant	DBH (cm)	Scorch height (m)
E. dives	278	1.641	−0.1603	
E. pauciflora (a)	347	0.2998	−0.2614	0.2453
E. pauciflora (b)	347	3.455	−0.1916	
E. rubida	194	3.645	−0.3234	
E. viminalis	91	4.541	−0.4653	

four major species. In the best GLM (that which explains most deviance in the dependent covariate) for each of the species, tree size was significant: height in *E. dives* and DBH in the other three. In the GLM for *E. dives* DBH was significant but explained less deviance than height. It was decided to use the GLM with DBH for *E. dives* in preference to that with height, to have uniform model structure across all species.

In the best GLM for *E. dives* a DBH^2 term was significant as well. While this model explained more deviance than that with only a linear DBH term, its prediction of higher stem death in trees of large DBH does not match the response found in the field data (Fig. 7.5).

Scorch height was significant only in the GLM for *E. pauciflora* but its effect may be overestimated in that model because of the limited information about scorch available at Gudgenby. Table 7.4 lists values of coefficients in the GLMs fitted, showing two models for *E. pauciflora*: with and without a scorch height term. The models are all biologically reasonable: coefficients of the DBH terms are all negative so the model predicts lower probabilities of stem death for larger trees and the scorch term for *E. pauciflora* predicts more stem death with higher

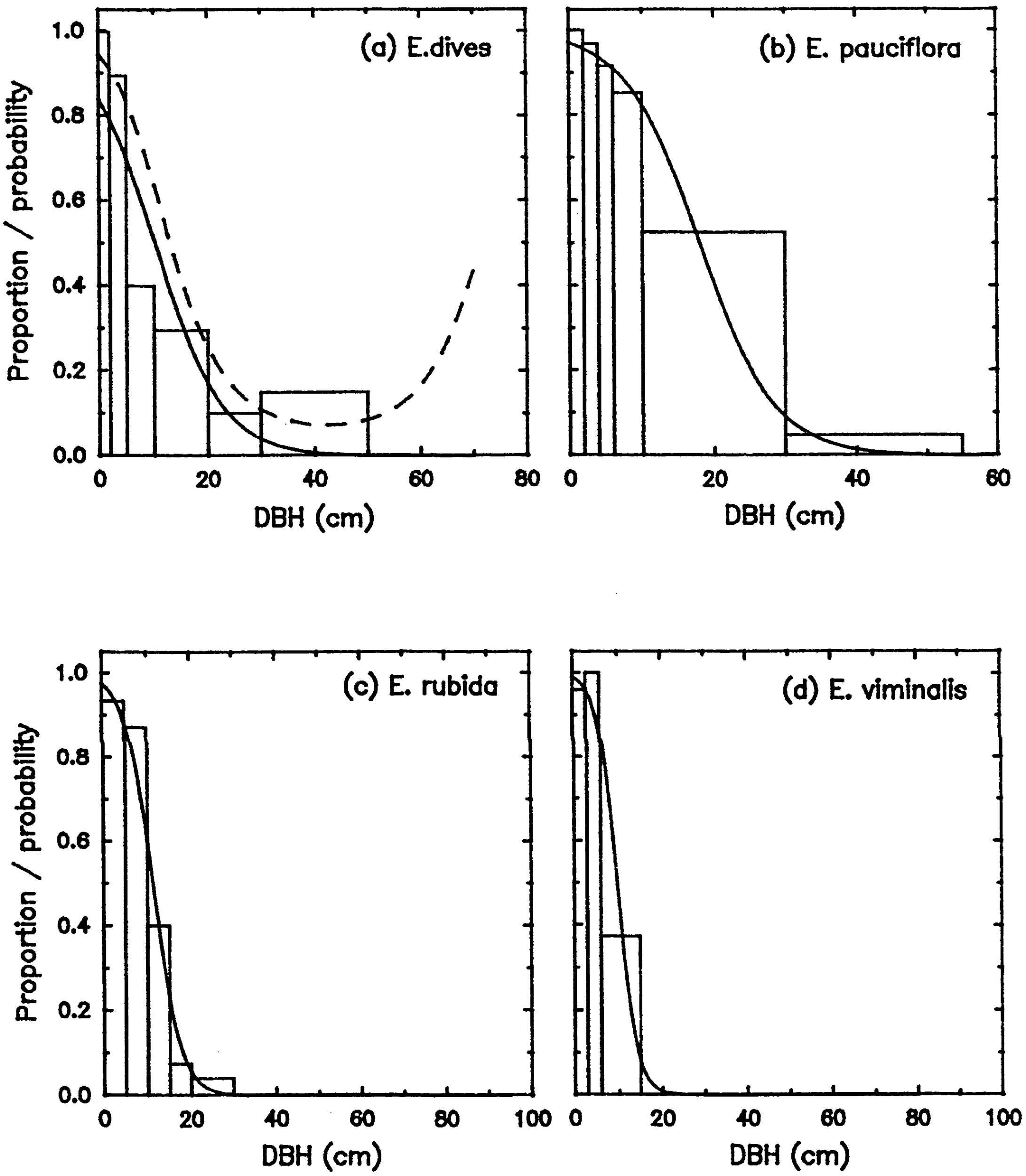

Figure 7.5: Proportion of stem death vs. DBH at Gudgenby in the four major species (histograms); uneven class widths are used because of low numbers of larger trees. Also shown are predictions of the second stage of the fire-response models: dashed line for *E. dives* is from GLM with quadratic DBH term; solid lines (all species) from GLMs with linear DBH terms. The range of DBH shown for each species indicates the range of sizes of trees in the field data.

scorch.

7.4.3 Growth recovery

To model growth recovery, post-fire height was expressed as a proportion of pre-fire height and a GLM with binomial error distribution fitted for the 424 trees with living stems (all species combined). DBH was the most significant term in the model and scorch height was also significant. Equation for the GLM with scorch height is:

$$\mathrm{LP} = 1.174 + 0.0484D - 0.0430S \tag{7.3}$$

where LP is linear predictor, D is DBH in cm and S is scorch height in metres. This model is biologically meaningful: larger trees recover better and higher scorch results in less recovery.

The uncertainty about scorch heights estimated at Gudgenby (Section 7.3.1) and their calculation in BRIND (Chapter 2 and Section 7.5 below) suggests that the GLM with scorch (Eqn 7.3) may over-predict the effects of scorch on growth recovery. The GLM without scorch has the equation:

$$\mathrm{LP} = 0.513 + 0.0477D \tag{7.4}$$

This model will predict increasing recovery for trees with bigger DBH. Its prediction is shown in Fig. 7.6, with values from the field data.

7.5 Testing of new model

There are few data available to test the new fire-response model. Noble (unpublished data) collected tree information before and after an experimental fire in the catchment of Bushrangers Creek in the Brindabella Range (O'Loughlin *et al.*

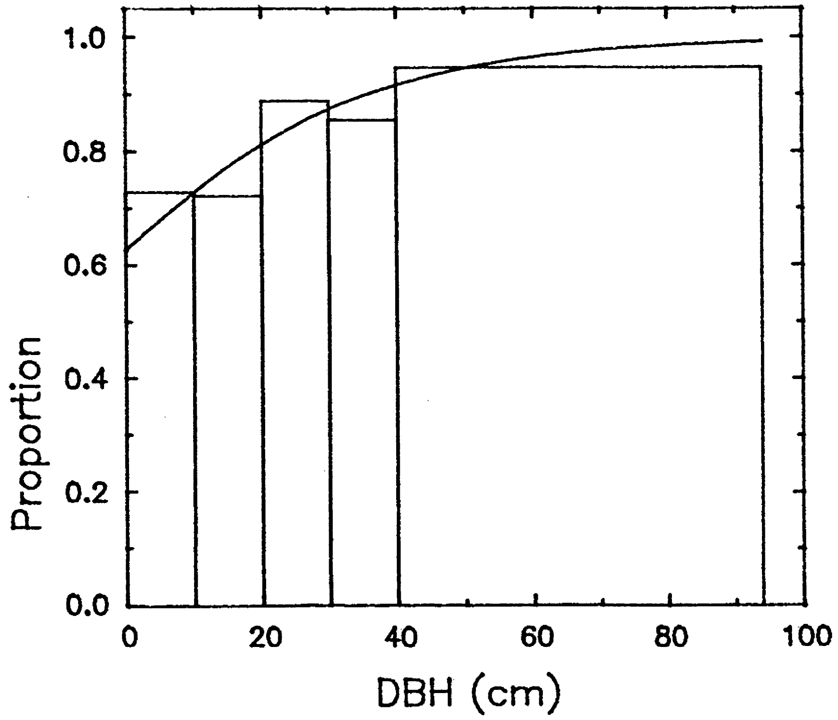

Figure 7.6: Post-fire height of trees expressed as a proportion of pre-fire height vs. DBH in trees with surviving stems at Gudgenby (histogram) and proportion predicted by the GLM for the third stage of the new fire-response model (solid line).

1982). Heights of the trees were not recorded (only DBH), so the mortality and growth recovery predictions of the response model could not be tested. The second stage of the model (prediction of stem death) was tested for the three species in Noble's data: *E. dives*, *E. dalrympleana* and *E. pauciflora.*

There were too few trees of *E. dalrympleana* recorded at Gudgenby to develop a separate fire-response model for that species, so the models for the taxonomically- and morphologically-similar species *E. viminalis* and *E. rubida* (Boland *et al.* 1984) were both used and their predictions compared. Using DBH measurements of the 2627 trees that survived the Bushrangers fire, probabilities were calculated of stem death for each.

One of the models for *E. pauciflora* also requires scorch height. I described in Chapter 2 how the fire-behaviour equations in BRIND calculate excessive flame height. While it is out of the scope of this study to provide detailed fire-behaviour equations, I need some realistic estimates of fire intensities in the Brindabella Range. Fine fuel in the fire was 22.5 t/ha, forest fire-danger index was 24 on the day of the fire (O'Loughlin *et al.* 1982) and the average slope of the area was calculated from a topographic map to be *c.* 20°. From these data the McArthur equations (Noble *et al.* 1980b) predict fire characteristics as rate of spread 2.6 km/h and flame height 37 m, which are clearly too high.

If the rate of spread value is divided by 7, following the findings of Kessell (1990) for eucalypt communities of this type (see Chapter 2), fire characteristics are calculated as rate of spread 0.37 km/h and flame height 8.2 m. Assuming that leaf scorch height is up to six times flame height (Luke and McArthur 1978), it is calculated as up to 50 m using the adjusted flame height (otherwise, up to 221 m). O'Loughlin *et al.* report tree heights up to 43 m and near-complete crown scorch and Noble (1984) reports 95% scorch from the same fire, so it appears that the adjusted estimates are closer to the real values.

Table 7.5: Summary of stem death predictions for Bushrangers trees. The range of probabilities of stem death predicted for each species is listed and the percentage of those probabilities > 0.5 to compare with the percentage stem death in the field data. Also listed is the percentage error in predicting the fate of trees on average. The two results shown for *E. pauciflora* are: (a) with scorch height term; and (b) without scorch height term; and for *E. dalrympleana*: using GLM fitted to (a) *E. viminalis* and (b) *E. rubida* from Gudgenby.

Species	No. of trees	Range of probabilities	% Probs > 0.5	% Stem death	% Error
E. dives	808	0.000-0.815	52.6	62.9	16.7
E. pauciflora (a)	1451	0.005–1.000	99.2	89.5	9.7
E. pauciflora (b)	1451	0.000–0.963	93.3	89.5	6.5
E. dalrympleana (a)	368	0.000–0.983	34.5	36.0	6.0
E. dalrympleana (b)	368	0.000–0.965	39.9	36.0	8.4

Table 7.5 summarises results from the tests of the fire-response model with the Bushrangers data. The range of probabilities of stem death predicted for the trees is listed for each model and the percentage of probabilities greater than 0.5. This latter figure indicates the expected percentage of trees whose stems will be killed in the whole model. Also shown are the rates of error in prediction: these are calculated as the number of 'wrong' predictions (probability > 0.5 when the tree's stem survived and probability ≤ 0.5 when it was killed) divided by the total number of trees. For *E. pauciflora*, results are shown for the models (a) with, and (b) without a scorch height term; and for *E. dalrympleana*, results are shown for (a) the *E. viminalis* and (b) the *E. rubida* models.

Prediction of stem death was similar to field results for all three species, with the ranking of the species responses predicted correctly. Stem death was under-predicted in *E. dives* and over-predicted in *E. pauciflora*. Height was a better predictor of stem death in *E. dives* but its GLM with a DBH term was chosen so the models would have a consistent form across the species. The *E. pauciflora* model with the scorch term over-predicted stem death more than that without. Experimentation with different estimates of scorch height (40 m, 30 m and 25 m) showed that this GLM still predicts more stem death than that without a scorch term. Given the uncertainty about calculating scorch heights expressed above, it is preferable to use the GLM without the scorch term (*E. pauciflora* model (a), Table 7.5). Of the two models for *E. dalrympleana*, the *E. viminalis* model ((a) in Table 7.5) made better predictions compared to the field data, although both models had lower rates of erroneous prediction than the models for the other two species.

7.6 Conclusions

The three-stage fire-response model is designed to simulate responses of fire-resistant eucalypt species more realistically than that in BRIND. The data from Gudgenby enabled the model to be implemented with biologically-meaningful equations but it is limited by the lack of a reliable measure of fire intensity.

In the new model, a tree whose stem survives a fire will be subject to increased probability of mortality when it is burnt again. This behaviour results from recovery of stems being modelled as reduction in height and thus vigour. It effectively simulates the hollowing of old trees by fire scars, which may then be killed by a fire which burns up through the hollow centre like a chimney (Luke and McArthur 1978).

Chapter 8

A new model

This chapter describes the construction of a new gap model called NEWBRIND that combines the changes to BRIND described in Chapters 4 to 7. NEWBRIND has been implemented in the C programming language on a variety of computer operating systems. A full listing of NEWBRIND is given in Appendix C.

NEWBRIND was run to simulate forest dynamics across a range of altitudes and aspects on the eastern side of the Brindabella Range. It is restricted to making predictions for sites on slope positions (i.e., not in gullies or on ridges) on soils of sedimentary and granitic origin. Its predictions were compared with field data as for BRIND in Chapter 3.

8.1 Construction of NEWBRIND

NEWBRIND has the same general structure as BRIND and other gap models: establishment, growth and death of individual trees on a small forest plot are simulated using an annual time-step.

8.1.1 Site description

The site to be simulated in NEWBRIND is described by its altitude, aspect and slope, and the geological parent material as sediments or granite. Altitude, aspect and slope are used to calculate the climatic conditions mean annual temperature, soil moisture index and number of growing months (Chapter 4). Site descriptors in NEWBRIND differ from those in BRIND by specifying the site itself rather than specifying the climatic measure (DEGD) directly.

A method for calculating the number of frosts occurring each simulated year was also added to NEWBRIND (in BRIND this value is supplied by the user). Daily surface climate records from the Bureau of Meteorology for the stations of Canberra, Cooma and Khancoban were used to calculate a relationship between monthly mean of minimum daily temperature (MinT) and number of frost events per month. The equation, obtained by regression using data from all three stations ($r^2 = 0.75$, $P < 0.001$), is $\sum_{i=1}^{12} F_i$ for 12 months, where

$$F_i = \begin{cases} 0 & \text{for } \mathrm{MinT}_i \geq 8.32°\mathrm{C} \\ 20.3 - 2.44\ \mathrm{MinT}_i & \text{for } \mathrm{MinT}_i < 8.32°\mathrm{C} \end{cases} \qquad (8.1)$$

Use of surface climate data with frost events was preferred to the commonly-used alternative method where frost days are estimated as the number of days when the minimum screen temperature is less than 2.2°C. When added to the climate model in NEWBRIND, mean annual numbers of frosts calculated by Eqn 8.1 for use in simulations with NEWBRIND are shown in Table 8.1.

Annual probability of wildfire is specified to NEWBRIND as it is in BRIND.

The species included in NEWBRIND are the 12 whose distributions were modelled by GLMs in Chapter 5. Species removed from the set modelled by BRIND

Table 8.1: Mean number of annual frosts predicted by Eqn 8.1 at each of the six altitudes used in the test simulations by NEWBRIND.

Altitude	Mean frosts
1600 m	163
1400 m	144
1200 m	115
1000 m	90
800 m	70
660 m	57

were small species of *Acacia*, *Bedfordia salicina*, *Exocarpos cupressiformis*, *Banksia marginata*, *Olearia argophylla*, *Eucalyptus stellulata*, the low cline-form of *E. pauciflora* and *E. rubida*. Shugart and Noble (1981) used a second set of DEGD parameters for low-altitude *E. pauciflora* to simulate the broad distribution of this species. This species was only found at high altitudes in the present study, which is restricted to slope positions and excludes the cold-air drainage valleys (ANU Forestry 1973). Species added were *E. macrorhyncha*, *E. mannifera*, *E. polyanthemos* and *E. rossii*.

Shugart and West (1979) showed that the correct plot size for a gap model is determined by characteristics of the dominant species. If the plot is too small an individual dominant tree cannot achieve its potential maximum size because its growth is limited by the stand crowding index S(BAR) (Chapter 2).

Results of the investigation of competition in *E. pauciflora* stands suggested that the radius of influence of this species is *c.* 8 m, which equates to a plot size of 1/50 ha (smaller than the 1/12 plot size in BRIND). While this plot size may be appropriate to forests dominated by the smaller *E. pauciflora*, it may be too small for those dominated by larger species, especially *E. delegatensis*.

Minimum plot size for a species can be calculated from its maximum DBH parameter $D_{\max}$ (Table 8.2). The biomass of a single tree is calculated from its DBH:

$$B = 7.123 \times 10^{-5} D^{2.58} \tag{8.2}$$

where B is biomass in tonnes (Chapter 2; Eqn 2.12 gives values in t/ha, adjusted for the 1/12 ha plot size of BRIND). A tree of *E. pauciflora* at $D_{\max}$ (91 cm) weighs 8.1 t, and one of *E. delegatensis* at $D_{\max}$ (213 cm) weighs 72.4 t. If SOILQ is 1000 t/ha then on a 1/50 ha plot (maximum 20 t) a tree of *E. pauciflora* can grow to its maximum size but one of *E. delegatensis* cannot. On a plot of 1/12 ha (83 t), *E. delegatensis* can grow to its maximum size. To ensure that trees of any species could grow to their maximum size, plot size in NEWBRIND was set to 1/12 ha.

The following sections describe construction of the components in NEWBRIND, elaborating on descriptions in previous chapters where necessary. Species parameters used in NEWBRIND are given in Table 8.2 and their derivations described below.

8.1.2 Growth

NEWBRIND includes the growth model described in Chapter 4, for which it is necessary to calculate parameter values for species other than *E. pauciflora*.

Species parameters used in the growth model are $D_{\max}$, $H_{\max}$, G and $\mathrm{MI}_{\min}$. Parameters $D_{\max}$ and $H_{\max}$ have the same values as in BRIND for most species, except for the adjustments to $H_{\max}$ described in Chapter 7 for *E. pauciflora* and *E. viminalis*. For species not in BRIND, values of $D_{\max}$ and $H_{\max}$ were taken from Boland *et al.* (1984).

Calculation of values for G was more complicated. I used the results of the growth analysis in Chapter 4 and values of G in BRIND to estimate distributions of G for species in NEWBRIND. The normal distribution of G for *E. pauciflora*

Table 8.2: Species parameters used in submodels of growth, establishment and mortality in NEWBRIND. The distribution of growth parameter G for the species is specified by μ and σ; $\mathrm{MI}_{\mathrm{min}}$ is used in the moisture stress index (Eqn 4.4); SdD is the minimum DBH for trees of the species to produce seeds; SdL is seed and lignotuberous seedling longevity (see text); FrMx is maximum frosts/year for establishment; flags are F=requires fire for germination, E=enhanced germination after fire, L=produces lignotubers and is fire-resistant; SImort is size-independent annual mortality rate, calculated from $\mathrm{Age}_{\mathrm{max}}$ (Chapter 6).

Species	D_{max} (cm)	H_{max} (cm)	G		$\mathrm{MI}_{\mathrm{min}}$	SdD (cm)	SdL (yr)	FrMx (yr^{-1})	Flags			$\mathrm{Age}_{\mathrm{max}}$ (yr)	SImort (yr^{-1})
			μ	σ					F	E	L		
Acacia melanoxylon	61	2133	106	68	0.55	6	50	70	1	1	0	100	0.0160
Eucalyptus dalrympleana	122	3658	44	28	0.57	12	45	100	0	0	1	400	0.0040
Eucalyptus delegatensis	213	6096	131	84	0.70	21	5	100	0	1	0	220	0.0073
Eucalyptus dives	76	2438	39	25	0.48	8	45	80	0	0	1	300	0.0054
Eucalyptus fastigata	183	4572	81	52	0.70	18	5	100	0	0	0	275	0.0058
Eucalyptus macrorhyncha	100	3500	85	54	0.44	10	45	70	0	0	1	200	0.0080
Eucalytpus mannifera	100	2500	69	44	0.49	10	45	70	0	0	1	180	0.0089
Eucalyptus pauciflora	91	2500	75	48	0.58	9	45	150	0	0	1	250	0.0064
Eucalyptus polyanthemos	100	2500	69	44	0.45	10	45	60	0	0	1	180	0.0089
Eucalyptus radiata	153	3976	86	55	0.52	15	45	100	0	0	1	220	0.0073
Eucalyptus rossii	100	2500	69	44	0.45	10	45	50	0	0	1	180	0.0089
Eucalyptus viminalis	122	3000	56	36	0.63	12	45	60	0	0	1	325	0.0049

that I derived in Chapter 4 from field growth data has mean 125 and standard deviation 80. The distribution's mean value is 0.93 times the value of 135 used for G in BRIND, and its coefficient of variation is 64%. Since no data were available for other species it was decided to calculate distributions of G for them using their G values in BRIND as a base: each was given a mean value of 0.93 times the value of G in BRIND and CV of 64%. For species not in BRIND a value of G was calculated from $D_{\max}$, $H_{\max}$ and $\mathrm{Age}_{\max}$ using the equation given by Botkin *et al.* (1972; shown in Table 2.2), then multiplied by 0.93 to give a value for μ_G.

Preliminary simulations using these parameters showed that slower-growing trees of each species (with lower G) were dying from suppression in the model, leaving only a group of faster-growing survivors (with higher G). For example, a simulation of stands of *E. pauciflora* at a site similar to those used for the tree-ring study in Chapter 4 (altitude 1400 m, 315° aspect) was performed using the parameters for G calculated in Chapter 4 ($\mu_G = 125$, $\sigma_G = 80$). In 2000 years, mean G of 409 trees that survived to grow larger than 5 cm DBH was 212, much higher than the mean value of 125 of trees at establishment. The CV of G in surviving trees was also severely reduced from 64% to 28%. Note that it is these growth rates of *surviving* trees that should be compared with the rates calculated from trees in the field in Chapter 4.

Experimental simulations showed that a value for μ_G of *c.* 75 for *E. pauciflora* is needed for NEWBRIND to predict a mean growth rate of *c.* 125 of surviving trees. A similar result was obtained for other, faster-growing species and suggests that μ_G values calculated using the method described above need to be adjusted by a factor of 0.6 for NEWBRIND to predict plausible growth rates of surviving trees.

Values of $\mathrm{MI}_{\min}$ (Table 8.2) for 13 of the 14 species were calculated using the CSIRO Queanbeyan-Shoalhaven data described in Chapter 4. The value for the remaining species, *E. delegatensis*, was estimated to be the same as for the

Table 8.3: Values of SOILQ in t/ha at 24 altitude–aspect locations in the Brindabella Range, to be used in simulations with NEWBRIND. Values were calculated from an optimum of 1000 t/ha multiplied by mean soil moisture index MI and growing months index T(GM) for each site.

Altitude	Aspect			
	45°	135°	225°	315°
1600 m	580	580	564	563
1400 m	633	640	600	590
1200 m	687	707	662	644
1000 m	715	751	696	672
800 m	744	791	726	690
660 m	755	825	776	687

ecologically-similar *E. fastigata*.

The other growth indices in the new growth submodel—T(GM), P(CI) and S(BAR)—have the same value for trees of all species. The competition intensity parameter k in P(CI) was set to 75.2 initially (see Chapter 4).

In Chapter 4 I argued that the S(BAR) index models total resource availability on the simulated plot. Its parameter SOILQ needs to be specified for each site, reflecting soil moisture and temperature limits to growth. For preliminary testing, SOILQ values for each site were set to an optimum value of 1000 t/ha scaled by mean soil moisture index MI and mean growing months index T(GM). Table 8.3 shows SOILQ values calculated for 24 altitude–aspect locations in the Brindabella Range used in the simulations. Maximum biomass is restricted by soil moisture availability at lower altitudes, by growing season length at higher altitudes, is greater on sheltered slopes (aspect 135°) and less on exposed slopes (aspect 315°).

8.1.3 Establishment

The establishment submodel in NEWBRIND is based on that in BRIND, except that species' probabilities of establishment are weighted by the GLMs that predict their occurrence, described in Chapter 5. Table 5.6 shows the GLM coefficients for the species.

Species' contributions to the 'seed pool' in NEWBRIND are also weighted by the biomass contribution of mature trees on the simulated plot. This behaviour models the much higher seed production by larger, dominant eucalypts than smaller trees (Jacobs 1955).

The pool of lignotuberous seedlings is modelled by giving lignotuberous species greater persistence in the simulated plot. Noble (1984) described the annual survival rate of lignotubers in the absence of fire to be 95% or more. An annual survival rate of 0.95 is modelled in NEWBRIND by setting the SDLIFE parameter (Chapter 2) of these species to 45 years. The vigorous growth of lignotuberous seedlings when released from overstorey influence is modelled by adding them to the simulated plot at a larger size than seedlings of non-lignotuberous species (1.7–2.2 cm vs. 1.2–1.7 cm DBH).

NEWBRIND has flags like those in BRIND (Chapter 2) that influence species' seed pool contributions (Table 8.2). Two flags are the same as those in BRIND: one specifying species that require fire for germination and one specifying those that show enhanced germination after fire (Shugart and Noble 1981). The lignotuber flag is new: all eucalypts in NEWBRIND except *E. delegatensis* and *E. fastigata* are lignotuberous. Lignotuberous species are also those that are resistant to fire (survive 100% canopy scorch, Chapter 7). The minimum DBH for seed production was set to one-tenth of $D_{\max}$, as in BRIND.

A low rate of seed dispersal is simulated in NEWBRIND by a small, constant addition to species' seed pool contributions (as in BRIND). The species set for NEWBRIND does not include any that are considered to be well-dispersed (such as *Bedfordia salicina* in BRIND), so all species have the same rate of seed input from dispersal.

Note that the environmental adjustment to a species' seed pool contribution using its GLM is calculated after weighting for mature trees, lignotuberous seedlings, fire effects and dispersal.

Values for the parameter that specifies the maximum number of frosts per year that each species can tolerate for establishment (FrMx in Table 8.2) were taken from species range information given by Boland *et al.* (1984). While the mechanisms of frost tolerance are complex (Tibbits and Reid 1987) and there is intraspecific variation in hardiness (Harwood 1980), the use of a single species parameter is appropriate to the annual time scale used in gap models and no more detailed information is available.

8.1.4 Mortality

In Chapter 6 I suggested the use of lower rates of size-independent mortality than those in BRIND by assuming that 20% of trees not killed by other causes (suppression, fire) will reach $\mathrm{Age}_{\mathrm{max}}$. This has been incorporated into NEWBRIND and Table 8.2 shows values of $\mathrm{Age}_{\mathrm{max}}$ and size-independent mortality (SImort) for the 14 species in the new model. Criteria for suppression mortality are the same as in BRIND: a species that grows by less than 0.1 cm in a year is marked as suppressed and has a probability of 0.368 of being killed that year (Chapter 2).

8.1.5 Fire response

Species in NEWBRIND are classified in their response to fire as being either fire-resistant or fire-sensitive. Fire-resistant species are those that can survive 100% canopy scorch and have lignotubers (Table 8.2); fire-sensitive species are killed by 100% scorch and do not have lignotubers. Fire-sensitive species are treated as in BRIND, where a tree will survive 50% canopy scorch but is killed by 100% scorch (Chapter 2).

To incorporate the new fire-response submodel for fire-resistant species described in Chapter 7 into NEWBRIND, it is necessary for the model to keep track of tree height and vigour index to predict mortality from fire and height recovery afterwards. An expected height, equivalent to its height if not burnt, is calculated for each tree from its DBH. A tree's vigour is its actual height divided by this expected height and remains at 1.0 until it is burnt. A tree whose stem survives a fire has its height (and therefore vigour) reduced (Chapter 7).

For fire-resistant species not found at Gudgenby, parameter values for modelling stem death (Chapter 7) were estimated from those available on the basis of taxonomic and morphological similarities between species. The assignments were: *E. macrorhyncha*, *E. polyanthemos* and *E. radiata* using values for *E. dives*; *E. dalrympleana* and *E. mannifera* using values for *E. viminalis*; and *E. rossii* using values for *E. rubida*. When a tree's stem is killed it is replaced by a lignotuberous 'sprout' with DBH uniformly distributed from 2 to 3 cm.

Fire intensity is calculated using the McArthur meter equations (Noble *et al.* 1980b) modified as described in Chapter 7. This simple modification (reducing calculated rate of spread by a factor of 7) provides plausible intensity values but fire behaviour should ideally be calculated using a better model, perhaps as part of a geographical information and modelling system, like that described by Kessell

(1990).

8.2 Preliminary model tests

Initial testing of NEWBRIND was designed to assess its performance at predicting general attributes of forests over the range of environments in the Brindabella Range. Its predictions of species distributions and plot biomass are compared with the same field data used in the comparisons described in Chapter 3.

These tests do not constitute a validation of the model (Mankin *et al.* 1977) because its predictions are being compared with some of the data used to parameterise it. They will however provide an indication of whether the changed model structure is capable of predicting plausible species distributions in the Brindabella Range.

Sets of 50 simulations were performed to predict forest composition at 24 sites throughout the Brindabella Range on sedimentary soil type. The sites were at 660, 800, 1000, 1200, 1400 and 1600 m altitude, and with aspects of 45°, 135°, 225° and 315°. Appendix B shows the site parameters used for each of the sites. Like those described in Chapter 3, each simulation was run for a 500-year lead time and then sampled 50 years after that in simulations without fire, or after the end of the first 50-year fire-free period after the lead time in simulations with fire.

Each simulation was begun from a bare plot into which the species can disperse with equal probability that is adjusted by values of their establishment GLM (Section 8.1.3).

8.2.1 Species distributions

Table 8.4 shows predictions by NEWBRIND of proportional biomass contributions of the most common species at the 24 sites in the Brindabella Range, from simulations performed without fire. These predictions can be compared with field distribution data shown in Table 8.5.

The predicted dominance of *E. radiata* is greater than in the field data. While this species is an important component of forests between 700 m and 1100 m altitude in the Brindabella Range it does not dominate forests over that range as extensively as NEWBRIND predicts. It can be seen from a comparison with Table 5.7 that the predicted distribution of *E. radiata* is much greater than would be expected from its establishment probabilities alone. This model behaviour results from the faster growth rate of this species ($\mu_G = 86$, $D_{\max} = 150$ cm) than for competing species (e.g., *E. mannifera* $\mu_G = 69$, $D_{\max} = 100$ cm; *E. dives* $\mu_G = 39$, $D_{\max} = 76$ cm).

NEWBRIND is more sensitive to rate of DBH growth than BRIND. This sensitivity occurs because climate has less effect on growth rates in the new model (Chapter 4) than in BRIND, where growth of species is severely restricted by T(DEGD) towards the extremes of their field ranges (Chapter 2). In reality there is variation in growth rates and maximum sizes between tree species (and within species) but such differences do not always result in the largest, fastest-growing species dominating all the time, as predicted by the model. The extreme asymmetry of competition predicted by NEWBRIND appears to be unrealistic.

Further simulations were performed at the 24 altitude–aspect sites in which the values of μ_G for *E. radiata*, *E. dalrympleana* and *E. dives* were all set to 70. Table 8.6 shows predictions of NEWBRIND using the revised values of μ_G, and can be compared with Table 8.4. Distributions predicted by NEWBRIND are now

Table 8.4: Mean values of proportion biomass predicted from fifty simulations without fire by NEWBRIND at each of 24 altitude–aspect sites on sediments in the Brindabella Range. Site parameters used in the simulations are given in Appendix B. Sufficient species are listed at each site so the cumulative mean proportion is at least 0.8. Species names are abbreviations of those given in Table 8.2.

Altitude	Aspect							
	45°		135°		225°		315°	
1600 m	E.pau	0.99	E.pau	0.98	E.pau	0.99	E.pau	0.98
1400 m	E.pau	0.94	E.pau	0.93	E.pau	0.97	E.pau	0.97
1200 m	E.pau	0.46	E.del	0.57	E.pau	0.58	E.pau	0.81
	E.del	0.39	E.pau	0.25	E.del	0.20		
					E.fas	0.19		
1000 m	E.rad	0.91	E.rad	0.75	E.rad	0.90	E.rad	0.90
			E.fas	0.19				
800 m	E.rad	0.93	E.rad	0.84	E.rad	0.95	E.rad	0.88
660 m	E.rad	0.68	E.rad	0.85	E.rad	0.78	E.rad	0.50
	E.mac	0.15			E.mac	0.12	E.mac	0.23
							E.man	0.14

Table 8.5: Mean values of proportion biomass from 87 field plots on sediments in each of 24 altitude–aspect cells in the Brindabella Range. Sufficient species are listed in each cell so that cumulative mean proportion is at least 0.8. Species names are abbreviations of those given in Table 8.2.

Altitude	Aspect							
	0°–89°		90°–179°		180°–269°		270°–359°	
> 1500 m	E.pau	1.00	E.pau	1.00	E.pau	1.00	E.pau	1.00
1301–1500 m	E.pau	0.95	E.pau	1.00	E.pau	0.91	E.pau	0.90
1101–1300 m	E.dal	0.56	E.del	0.62	E.del	0.34	E.pau	0.56
	E.pau	0.24	E.pau	0.26	E.dal	0.24	E.dal	0.34
					E.div	0.23		
901–1100 m	E.div	0.27	E.dal	0.26	E.fas	0.36	E.div	0.76
	E.rad	0.23	E.del	0.24	E.rad	0.24	E.dal	0.14
	E.dal	0.13	E.rad	0.24	E.div	0.18		
	E.vim	0.12	E.man	0.15	E.vim	0.17		
	E.man	0.11						
701–900 m	E.div	0.31	E.mac	0.34	E.rad	0.42	E.rad	0.28
	E.vim	0.21	E.rad	0.25	E.div	0.22	E.ros	0.23
	E.mac	0.12	E.man	0.21	E.mac	0.21	E.div	0.21
	E.rad	0.12					E.man	0.15
≤ 700 m	E.ros	0.41	E.mac	0.39	E.ros	0.42	E.mac	0.40
	E.mac	0.31	E.rad	0.26	E.mac	0.27	E.div	0.26
	E.pol	0.10	E.div	0.17	E.div	0.21	E.ros	0.23

closer to those in the field data. In general, differences are minor and the small sample size of the field data precludes detailed examination but some differences can be seen that are attributable to values of model parameters.

The model predicts *E. fastigata* to be more common and occur over a wider altitudinal range than its field distribution. The higher growth rate of this species ($\mu_G = 86$, $D_{\max} = 183$ cm) causes it to dominate more than expected from its low probability of establishment. The GLM used for *E. fastigata* was calculated using the large regional data set and is not a good predictor of the distribution of this species in the study area. It was chosen because this species was not common enough for a GLM to fitted to local data only (Chapter 5).

Eucalyptus viminalis is more common in the field than was predicted by NEWBRIND. The GLM for this species predicts it to be a regular component of forests at *c.* 1000 m altitude but its slower growth rate ($\mu_G = 56$) reduces its competitive effectiveness severely so its predicted abundance is very low.

To assess the effect of fire on model predictions, a further set of simulations in the 24 altitude–aspect locations was performed using the same parameters as for those shown in Table 8.6 but with the addition of fire, using an annual probability of 0.02 (as in Chapter 3). The results are shown in Table 8.7. The overall effect of fire on species distributions predicted by NEWBRIND is small because most species are fire-resistant and respond in a similar way to each fire event.

The exceptions are the two fire-sensitive species, *E. delegatensis* and *E. fastigata*, that were severely reduced in predicted abundance. This is a result of their sensitivity to individual fires, where repeated fires with a short interval in between will cause them to become locally extinct, and is a plausible prediction (Shugart and Noble 1981; Chapter 2).

Table 8.6: Mean values of proportion biomass predicted from fifty simulations without fire by NEWBRIND at each of 24 altitude–aspect sites on sediments in the Brindabella Range. Site parameters used in the simulations are given in Appendix B. Sufficient species are listed at each site so the cumulative mean proportion is at least 0.8. Species names are abbreviations of those given in Table 8.2. Predictions shown here are from simulations using modified values of the species parameter μ_G as described in the text (cf. Tables 8.4 and 8.5).

Altitude	Aspect							
	45°		135°		225°		315°	
1600 m	E.pau	0.99	E.pau	0.98	E.pau	0.99	E.pau	0.98
1400 m	E.pau	0.88	E.pau	0.89	E.pau	0.95	E.pau	0.96
1200 m	E.pau	0.46	E.del	0.56	E.pau	0.49	E.pau	0.58
	E.del	0.25	E.pau	0.29	E.dal	0.24	E.dal	0.30
	E.dal	0.22			E.del	0.19		
1000 m	E.dal	0.42	E.fas	0.42	E.dal	0.43	E.dal	0.67
	E.fas	0.26	E.rad	0.19	E.fas	0.28	E.rad	0.13
	E.rad	0.18	E.del	0.18	E.rad	0.15		
			E.dal	0.13				
800 m	E.div	0.40	E.rad	0.37	E.div	0.37	E.div	0.50
	E.rad	0.24	E.fas	0.30	E.rad	0.24	E.dal	0.20
	E.man	0.12	E.div	0.18	E.fas	0.14	E.man	0.09
	E.fas	0.09			E.man	0.11	E.rad	0.08
660 m	E.mac	0.28	E.rad	0.36	E.mac	0.35	E.div	0.31
	E.div	0.27	E.mac	0.24	E.div	0.24	E.mac	0.30
	E.man	0.21	E.man	0.18	E.man	0.19	E.man	0.16
	E.rad	0.11	E.div	0.12	E.rad	0.11	E.ros	0.12

Table 8.7: Mean values of proportion biomass predicted from fifty simulations with annual fire probability of 0.02 by NEWBRIND at each of 24 altitude–aspect sites on sediments in the Brindabella Range. Site parameters used in the simulations are given in Appendix B. Sufficient species are listed at each site so the cumulative mean proportion is at least 0.8. Species names are abbreviations of those given in Table 8.2. Predictions shown here are from simulations using modified values of the species parameter μ_G as described in the text (cf. Tables 8.4, 8.5 and 8.6).

Altitude	Aspect							
	45°		135°		225°		315°	
1600 m	E.pau	1.00	E.pau	1.00	E.pau	1.00	E.pau	1.00
1400 m	E.pau	0.99	E.pau	0.97	E.pau	0.99	E.pau	0.98
1200 m	E.pau	0.69	E.pau	0.66	E.pau	0.64	E.pau	0.63
	E.dal	0.28	E.dal	0.22	E.dal	0.27	E.dal	0.34
1000 m	E.dal	0.65	E.dal	0.40	E.dal	0.62	E.dal	0.83
	E.rad	0.19	E.rad	0.40	E.rad	0.23		
800 m	E.div	0.49	E.rad	0.55	E.div	0.46	E.div	0.61
	E.rad	0.22	E.div	0.35	E.rad	0.29	E.dal	0.16
	E.man	0.11			E.man	0.09	E.man	0.08
660 m	E.mac	0.33	E.rad	0.30	E.mac	0.35	E.mac	0.38
	E.div	0.26	E.mac	0.29	E.div	0.27	E.div	0.26
	E.man	0.20	E.man	0.21	E.man	0.19	E.man	0.11
	E.ros	0.08					E.ros	0.10

8.2.2 Biomass

Table 8.8 shows mean biomass values of field plots in 24 altitude–aspect cells in the Brindabella Range, and mean biomass values predicted by NEWBRIND at 24 equivalent locations without fire and with annual probability of fire of 0.02. Overall, NEWBRIND predicts plot biomass near to the field values at altitudes between 900 m and 1500 m but consistently predicts too much biomass outside this range. The field data show a clear response to altitude with higher values of plot biomass at intermediate altitudes and lower values at the extremes (Chapter 3) that is not replicated by the model.

A GLM was fitted to predicted plot biomass for the 2400 plots (50 replicates by 24 sites by fire/non-fire) with normal distribution of errors. Terms were factors of altitude class (6 levels), aspect class (4), fire (2) and their interactions. Significant terms ($P < 0.001$) were altitude, aspect, fire and the interaction of altitude and aspect.

In general, predicted plot biomass decreased with increasing altitude. Overall response to aspect was for highest predicted biomass on sheltered slopes, lowest on exposed slopes and intermediate on the other aspects. The pattern of response to aspect was not consistent at all altitudes, as described by the interaction term in the GLM. These responses reflect quite clearly the values of SOILQ for the 24 sites (Table 8.3) that were set according to the growth indices of temperature and soil moisture, T(GM) and MI (Section 8.1.2).

Predicted biomass was consistently less in simulations with fire, by an average of 22 t/ha. Reduction in plot biomass in simulations with fire was greater at the sites where the fast-growing, fire-sensitive species were reduced in abundance: those at 800, 1000 and 1200 m altitude on all but the most exposed aspects, where these species do not occur.

Table 8.8: Mean values of plot biomass in t/ha from 87 0.1 ha field plots in 24 altitude–aspect cells in the Brindabella Range on sedimentary soils, and mean predicted plot biomass at 24 equivalent altitude–aspect sites on sediments, without fire and with annual fire probability of 0.02. Site parameters used in the simulations are given in Appendix B.

Altitude	Source	Aspect			
		45°	135°	225°	315°
> 1500 m	field	171	144	165	151
1600 m	model, no fire	275	285	258	258
	model, fire	255	271	253	246
1301–1500 m	field	273	330	193	230
1400 m	model, no fire	319	324	302	284
	model, fire	288	304	276	265
1101–1300 m	field	444	324	428	286
1200 m	model, no fire	363	381	349	323
	model, fire	324	352	329	304
901–1100 m	field	351	420	377	351
1000 m	model, no fire	393	388	374	362
	model, fire	354	355	352	346
701–900 m	field	263	224	216	220
800 m	model, no fire	370	416	384	343
	model, fire	354	403	358	321
≤ 700 m	field	175	241	174	196
660 m	model, no fire	396	436	406	352
	model, fire	369	404	389	339

Plot biomass prediction in NEWBRIND is controlled by a complex interaction of many components of the model, like in BRIND. The maximum biomass parameter SOILQ clearly has a significant effect on biomass prediction but it is a parameter for which values are difficult to determine accurately.

8.2.3 Model dynamics

A feature of BRIND is its ability to predict the dynamics of forests subject to recurrent fire. Shugart and Noble (1981) showed that BRIND can predict plausible stand dynamics of forests in the *E. delegatensis* zone where this species dominates on suitable sites in the absence of fire.

NEWBRIND can make similar predictions, and Fig. 8.1 shows simulated dynamics from a 500-year example simulation of a forest plot on a sheltered slope (aspect 135°) at 1200 m altitude, during which time one fire occurred. Shugart and Noble (1981: Fig. 2) showed very similar dynamics predicted by BRIND.

The simulation was started from an initial plot with four small individuals of the three common species found in this zone: *E. dalrympleana*, *E. delegatensis* and *E. pauciflora*. During the first 100 years *E. pauciflora* maintained a significant presence in the stand but by *c.* 150 years *E. delegatensis* with its faster growth rate had gained dominance and *E. pauciflora* declined. By *c.* 220 years all trees of *E. pauciflora* had died from suppression leaving this species present only in the lignotuberous seedling pool.

The fire in year 264 was intense enough to kill all trees of *E. delegatensis.* This species re-established from seedlings and was able to successfully compete against the lignotuberous seedlings of the other two species which were released by the fire.

This pattern of domination by *E. delegatensis* will continue in the model

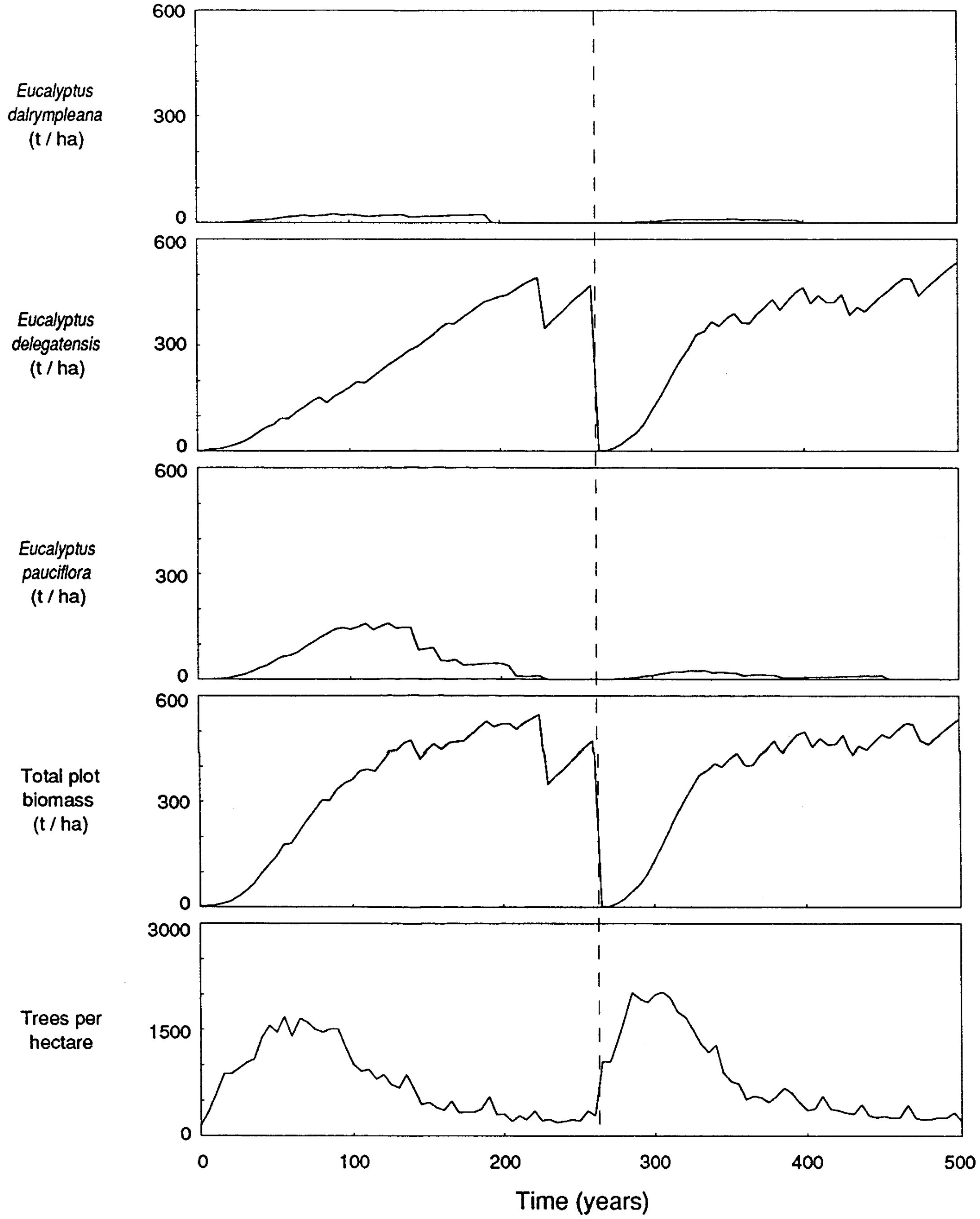

Figure 8.1: Example simulated stand dynamics of a forest plot on a sheltered slope (aspect 135°) at 1200 m altitude. The simulation was for a 500-year period during which one fire occurred, in year 264 (marked by the dashed line). Shown are plot biomass, total numbers of trees and biomass values for the three most important species.

unless two intense fires occur with an interfire period less than the maturation time of this species and cause it to become locally extinct. It is noteworthy that NEWBRIND does not explicitly model a pool of lignotuberous seedlings but the setting of particular model parameters (Section 8.1.3) creates a plausible simulation of one at the temporal and spatial scale at which the model operates.

Varying frequency of fire in NEWBRIND causes the model to predict mixtures of the three species like those described by Shugart and Noble (1981) for BRIND. For example, *E. delegatensis* regenerates strongly after a fire but with frequent fires it is lost from the simulated plot and a mixture of *E. dalrympleana* and *E. pauciflora* remains. These dynamics predicted by the model are not unidirectional successions (Chapter 1) but depend on the frequency and severity of fire.

The *E. delegatensis* forests have tall, closed canopies indicative of intense competition for light. NEWBRIND is able to predict plausible dynamics of these forests using a generalised one-sided competition index instead of using the light competition index in BRIND and many other gap models.

8.3 Conclusions

My aims in this study were to make improvements to BRIND based on a detailed study of its structure and predictions, and to expand the range of environments in which it can make predictions. Examination of that model directed my changes towards improving what I saw as unrealistic aspects of its structure. I believe that I have succeeded in these aims—the new model makes predictions comparable to those of BRIND but does so over two environmental gradients and with some more realistic structure—although some problems remain with NEWBRIND. In this concluding section I discuss the changes made to BRIND and make some observations about models and the modelling process.

8.3.1 NEWBRIND

The most important change to BRIND was to move the emphasis in prediction of response to environmental gradients from growth to establishment. I showed in Chapter 2 that, in BRIND, reduction of radial growth of all trees towards the ends of their field ranges combined with limited environmental control of establishment expresses a hidden assumption that growth is more important than establishment in controlling the distribution of tree species along environmental gradients. I argued that this assumption is unrealistic and I changed the model by reducing the strength of environmental effects on growth while explicitly modelling its effects on establishment ability.

The new model's prediction of species responses to the gradients of temperature and soil moisture are primarily controlled by establishment ability, as desired. There are problems with predictions by NEWBRIND that result from inaccuracies in the species GLMs used to calculate establishment probabilities. These problems are one of parameterisation rather than design and can be overcome with more field data of an appropriate nature.

Most gap models use T(DEGD) and some use other indices that also reduce growth to zero at the extremes of species field ranges (e.g., the model of Pastor and Post 1986). If these models are used to predict changes in forests along environmental gradients (like BRIND) they may predict plausible species distributions but my criticism of BRIND also applies to them: they are modelling an unrealistic mechanism. I believe that this issue has remained hidden because gap models have been used primarily as models of temporal change (succession).

A side-effect of removing strong environmental control from growth was that model predictions became very sensitive to the species growth parameters. This behaviour reveals that the outcome of competition as modelled by NEWBRIND is strongly deterministic: trees (and species) that grow faster nearly always suppress slower-growing ones, even when the difference between growth rates is small. Competitive outcomes in reality are less deterministic, with higher variance in growth rates found in co-existing trees than predicted by the model (Chapter 4).

The problem of strongly-deterministic competition is another hidden assumption that has emerged as a result of the change of model structure. It is a product of competition and suppression in the model: smaller trees have their growth reduced most; if the annual DBH increment of a tree is reduced to less than 0.1 cm it is marked as suppressed; suppressed trees suffer much higher mortality (Chapter 2). Slower-growing trees with smaller increments are always more likely to become suppressed and be removed from the simulated plot; thus faster-growing trees are always favoured.

The determinism of competition is present in BRIND but did not emerge as an important problem until the strong environmental control of growth in that model was removed. The principal effect of this determinism in both models is to reduce variance in model predictions. In NEWBRIND this emerges as increased sensitivity to growth rate parameters, so that it becomes important to estimate them as accurately as possible.

Another change to BRIND was to introduce variation in species' intrinsic growth rates by specifying distributions of the growth parameter G for each, rather than the fixed values in BRIND. This change is novel and is designed to simulate the natural variation in growth rates found in field trees resulting from genetic and microhabitat variability (Chapter 4). Examination of the effects of this change brought into sharp focus the extreme determinism in competition described above by showing that, in NEWBRIND, only the fastest-growing individuals of a species survived and intraspecific variance in G was severely reduced (Section 8.1.2).

A further change to BRIND was to reduce rates of random, size-independent (background) mortality in all species to plausible levels. The primary effect of higher mortality in BRIND is to reduce predicted biomass by causing a fast turnover of mature trees. The equation used to calculate background mortality rates in BRIND is also found in many other gap models, which suggests that they also predict unrealistic survivorship of trees. It may not have a significant effect on conclusions drawn from predictions by these models.

Extension of the environmental range of BRIND was achieved by the addition of a simple model of available soil moisture (Chapter 4). This model uses monthly rainfall, temperature and evaporation estimates to calculate soil moisture status that is used in a growth-modifying index and in calculation of species establishment probabilities. From the preliminary model tests presented in this chapter, its performance appears to be satisfactory.

A new submodel of fire response for fire-resistant eucalypt species was developed using field data collected after a fire. Modelling of response in BRIND, which consists of only immediate and longer-term mortality of trees from fire, was replaced with a three-stage submodel for fire in which a tree may be killed, have its above-ground parts killed or survive but with reduced vigour.

I removed the submodel of sprouting from BRIND that simply replaces most trees dying from any cause with a vigorous sapling. It is a poor model of sprouting in eucalypts and contributes significantly to prediction of excessive plot biomass by BRIND (Chapter 3). Lignotuberous and epicormic sprouting is modelled in NEWBRIND as part of trees' response to fire (Chapter 7).

In summary, NEWBRIND is an improvement on BRIND in a number of ways and makes comparable predictions over a broader range of environments. NEWBRIND still has some problems, principally in three areas:

1. Sensitivity to values of growth rate parameters and reduction in variance of intraspecific growth rates, which both result from the extreme determinism in the outcomes of competitive interactions. The combination of competition, suppression and growth rates in the model needs to be examined in more detail.

   An alternative model of suppression mortality was investigated, in which the criterion for marking a tree as suppressed was changed from an absolute DBH increment (suppression increment) of 0.1 cm to a relative change in increment—the combined reduction in growth from climate and competition. With this suppression model, two small trees of equal size but different growth rates are both suppressed, instead of only the slower-growing one. Experimental values of relative change varied from 0.1 to 0.5 but none had a significant effect on dominance by faster-growing species or on growth rate variance. This is because slower-growing trees spend more time in the lowest stratum of the stand and are subjected to strong competition for longer periods than faster-growing species.

   Perhaps susceptibility to suppression is related to tree age rather than size so that, for example, trees are only suppressed by competition during the first 10 years of their lives (regardless of the size of their growth increments).

2. The GLMs used to predict species establishment probabilities are inadequate for some species. This problem can be addressed in part by the collection of more field data to calculate more reliable models, but reliability of GLMs (as indicated by the maximum probabilities they predict) is also limited by the use of only two gradients. Predictions of GLMs could be improved by the addition of further environmental indices of (for example) topographic position (Austin *et al.* 1990).

In many cases the field data for fitting of GLMs may not be available. Establishment probabilities could possibly be weighted by T(DEGD) and similar indices derived from maps of species distributions (e.g., Aber *et al.* 1982; Pastor and Post 1986).

A possible solution to using GLMs whose predictions of occurrence are low because of incomplete environmental description is to scale predictions so that the maximum value for all species is the same. This change may have the side-effect of causing some species to be predicted as more common than they actually are.

3. NEWBRIND is not able to accurately predict the general changes in plot biomass with altitude in the Brindabella Range. The sensitivity of plot biomass to SOILQ combined with the difficulty of defining values for that parameter make biomass predictions by the model unreliable.

   It is possible that modelling changes with environment in other components of NEWBRIND may reduce the model's sensitivity to SOILQ. Parameters whose values could be made in some way dependent on temperature and soil moisture include the number of sites available for establishment each year and species parameters such as $D_{\max}$ and μ_G.

8.3.2 Observations about modelling

Some observations about modelling in general can be made from the work in this study.

BRIND and NEWBRIND make comparable predictions of species composition and dynamics in the Brindabella Range in spite of important differences in structure. Both models show inaccurate predictions that can be improved by suitable choice of parameter values. The question arises as to which is the better model.

There are doubts about the reliability of detailed predictions by both models because although the modelling of growth rates along environmental gradients and of mortality rates in NEWBRIND is based on more realistic principles, this model still predicts almost deterministic outcomes of competition (as death by suppression).

It is encouraging to see the efficacy of two idealisations that were included in the formulation of NEWBRIND. The simple treatment of lignotuberous seedling recruitment (Section 8.1.3) predicts plausible persistence of lignotuberous species in the absence of mature trees. However, this component of NEWBRIND is of no use if size and dynamics of the lignotuberous seedling pool need to be predicted.

The simple one-sided competition model incorporated into NEWBRIND appears to function adequately for general predictions of forest dynamics, as did the light competition index in BRIND. Its use poses the interesting question of which competition model is to be preferred: an idealisation expressing a general principle, or a representation of a mechanism that is known to be important at some sites (tall *E. delegatensis* stands) and unimportant at others (higher-altitude open *E. pauciflora* stands).

The emergence of hidden assumptions in the two models is an interesting phenomenon. Existence of implicit assumptions like these can be difficult to predict in advance. The environmental response assumption was found by evaluating the supporting rationale (Chapter 1) of BRIND as a whole but the excessive determinism assumption did not show clearly until predictions were made using the changed model. Both assumptions emerge from the structure of the models and appear to define another level of complexity within them, at the level of the species.

Finally, there are some observations about model generality that are relevant to forest gap models. Some components of the first forest gap model JABOWA (equations and methods of parameter estimation) have been adopted unmodified into many other gap models (FORET and some of its descendents including BRIND). These components together may be seen to constitute a 'philosophy' of forest modelling that is quite general because of its successful application to a variety of forest types.

In this study I have questioned the validity of some of these model components: that growth is the primary determinant of species distributions along environmental gradients; that background mortality rates are as high as predicted by the model; and that light is always the principal resource for which trees compete. I also questioned the validity of a more recently-discovered implicit assumption, that the outcome of competition is very strongly biased in favour of faster-growing trees.

Another part of the JABOWA 'philosophy' is that model equations use species parameters determined from the entire geographic range of each species (Chapter 1). In developing a new submodel of establishment for NEWBRIND, I attempted to follow this principle by using the widest species ranges for which data were available. These data do not include the complete geographical range of the species concerned but cover a large proportion of their environmental range as defined by temperature, rainfall and radiation (Boland *et al.* 1984). The GLMs fitted to the large data set were unable to accurately predict species distributions in the Brindabella Range from temperature and soil moisture because that environmental description was inadequate. This situation requires the modeller to choose whether to make the establishment submodel more complex by including more environmental dimensions or to limit its domain and use a simpler formulation.

There is generality in the gap model formulation but it is not found at the level of the 'philosophy' described above. The general core of gap models is the nature of relationships between species included in the model. I believe that this core is essentially qualitative in its predictions, relying on critical differences in species attributes and forms of equations. A typical gap model will predict a long-lived, slow-growing, shade tolerant tree species to maintain a presence under the canopy of a shorter-lived, fast-growing intolerant species and later dominate the simulated plot. Such a sequence is quite plausible but model predictions will be much less sensitive to the values of species growth rate and age parameters than their ranking. Similarly, the exact form of the competition equation is not important provided that it predicts an advantage in growth and survival of bigger trees.

Thus the core of gap models is general and realistic but its predictions are not precise (Levins 1966). Each implementation needs to be carefully constructed and its predictions validated so that it can make realistic predictions at a more precise level. Nevertheless, gap models have been successfully validated for a wide variety for forest modelling applications and have proved to be practical and useful tools in the study of forest dynamics. However, if such precision is not required in a particular modelling situation it may be more appropriate to choose a different model structure that is always less precise but which requires less information to implement, such as the FATE model (Moore and Noble 1990).

Bibliography

Aber J. D., Melillo J. M. and Federer C. A. (1982) Predicting the effects of rotation length, harvest intensity, and fertilization on fiber yield from northern hardwood forests in New England. *Forest Science* **28**, 31–45.

Allen T. F. H. and Starr T. B. (1982) *Hierarchy: Perspectives for Ecological Complexity.* University of Chicago Press, Chicago.

Amos G. L., Bisset I. J. W. and Dadswell H. E. (1950) Wood structure in relation to growth in *Eucalyptus gigantea* Hook f. *Australian Journal of Scientific Research* **3**, 393-413.

ANU Forestry (1973) *A Resource and Management Survey of the Cotter River Catchment.* Department of Forestry, Australian National University. x + 287 pp.

Adomeit E. M., Austin M. P., Hutchinson M. F. and Christenson K. (1984) *A Rainfall Surface and Contour Map of South-eastern Australia.* Technical Memorandum 84/1: CSIRO Division of Water and Land Resources, Canberra. 39 pp.

Ashton D. H. (1976) Phosphorus in forest ecosystems at Beenak, Victoria. *Journal of Ecology* **64**, 171–186.

Attiwill P. W. (1966) A method for estimating crown weight in eucalypts, and some implications of relationships between crown weight and stem diameter. *Ecology* **47**, 795–804.

Austin M. P. (1971) Role of regression analysis in plant ecology. *Proceedings of the Ecological Society of Australia* **6**, 63–75.

Austin M. P. (1982) Use of a relative physiological performance value in the prediction of performance in multispecies mixtures from monoculture performance. *Journal of Ecology* **70**, 559–570.

Austin M. P. (1985) Continuum concept, ordination methods, and niche theory. *Annual Review of Ecology and Systematics* **16**, 39–61.

Austin M. P. (1986) The theoretical basis of vegetation science. *Trends in Ecology and Evolution* **1**, 161–164.

Austin M. P. (1987) Models for the analysis of species' response to environmental gradients. *Vegetatio* **69**, 35–45.

Austin M. P. and Cunningham R. B. (1981) Observational analysis of environmental gradients. *Proceedings of the Ecological Society of Australia* **11**, 109–119.

Austin M. P., Cunningham R. B. and Good R. B. (1983) Altitudinal distribution of several eucalypt species in relation to other environmental factors in southern New South Wales. *Australian Journal of Ecology* **8**, 169–180.

Austin M. P., Cunningham R. B. and Fleming P. M. (1984) New approaches to direct gradient analysis using environmental scalars and statistical curve–fitting procedures. *Vegetatio* **55**, 11-27.

Austin M. P., Groves R. H., Fresco L. F. M. and Kaye P. E. (1985) Relative growth of six thistles along a nutrient gradient with multispecies competition. *Journal of Ecology* **73**, 667–684.

Austin M. P., Nicholls A. O. and Margules C. R. (1990) Measurement of the realized qualitative niche: environmental niches of five *Eucalyptus* species. *Ecological Monographs* **60**, 161–177.

Banks J. C. G. (1982) *The use of dendrochronology in the interpretation of the dynamics of the snow gum forest.* PhD Thesis, Australian National University.

Beck D. E. (1974) Predicting growth of individual trees in thinned stands of yellow poplar. In: *Growth Models for Tree and Stand Simulation* (ed. J. Fries), pp. 47–55. Research Notes 30, Department of Forest Yield Research, Royal College of Forestry, Stockholm.

Bella I. E. (1971) A new competition model for individual trees. *Forest Science* **17**, 364–372.

Boland D. J., Brooker M. I. H., Chippendale G. M., Hall N., Hyland B. P. M., Johnston R. D., Kleinig D. A. and Turner J. D. (1984) *Forest Trees of Australia.* Nelson and CSIRO, Melbourne. xvi + 687 pp. 4th edition.

Booth T. H. and Ryan P. (1985) Climatic effects on the diameter growth of *Araucaria cunninghamii* Ait. ex D. Don. *Forest Ecology and Management* **10**, 297–311.

Bormann F. H. and Likens G. E. (1979) *Pattern and Process in a Forested Ecosystem.* Springer–Verlag, New York.

Borough C. J., Incoll W. D., May J. R. and Bird T. (1978) Yield statistics. In: *Eucalyptus for Wood Production* (eds W. E. Hillis and A. G. Brown), pp. 201–205. CSIRO, Canberra.

Botkin D. B. (1975) Functional groups of organisms in model ecosystems. In: *Ecosystem Analysis and Prediction* (ed S. A. Levin), pp. 98–102. Society for Industrial and Applied Mathematics, Philadelphia.

Botkin D. B. (1981) Causality and succession. In: *Forest Succession: Concepts and Application* (eds D. C. West, H. H. Shugart and D. B. Botkin), pp. 36–55. Springer–Verlag, New York.

Botkin D. B., Janak J. F. and Wallis J. R. (1972) Some ecological consequenses of a computer model of forest growth. *Journal of Ecology* **60**, 849–873.

Bowman D. M. J. S. and Kirkpatrick J. B. (1984) Geographic variation in the demographic structure of stands of *Eucalyptus delegatensis* R. T. Baker on dolerite in Tasmania. *Journal of Biogeography* **11**, 427–437.

Bowman D. M. J. S. and Kirkpatrick J. B. (1986) Establishment, suppression and growth of *Eucalyptus delegatensis* R. T. Baker in multiaged forests. I. The effects of fire on mortality and seedling establishment. *Australian Journal of Botany* **34**, 63–72.

Busing R. T. and Clebsch E. E. C. (1987) Application of a spruce–fir forest canopy gap model. *Forest Ecololgy and Management* **20**, 151–169.

Cale W. G., O'Neill R. V. and Shugart H. H. (1983) Development and application of desirable ecological models. *Ecological Modelling* **18**, 171–186.

Caswell H. (1976) The validation problem. In: *Systems Analysis and Simulation in Ecology Volume IV* (ed. B. C. Patten), pp. 313–325. Academic Press, New York.

Chilvers G. A. and Brittain E. G. (1972) Plant competition mediated by host-specific parasites—a simple model. *Australian Journal of Biological Science* **25**, 749–56.

Clements F. E. (1916) *Plant succession: An analysis of the development of vegetation.* Carnegie Institute Publication 242, Washington D. C. 512 pp.

Cochrane G. R. (1966) Bushfires and vegetation regeneration. *Victorian Naturalist* **83**, 4–10.

Connell J. H. (1975) Some mechanisms producing structure in natural communities: a model and evidence from field experiments. In: *Ecology and evolution of communities* (eds M. L. Cody and J. M. Diamond), pp. 460–490. Harvard University Press, Cambridge Massachusetts.

Connell J. H. and Slatyer R. O. (1977) Mechanisms of succession in natural communities and their role in community stability and organization. *American Naturalist* **111**, 1119–1144.

Dale V. H., Doyle T. W. and Shugart H. H. (1985) A comparison of tree growth models. *Ecological Modelling* **29**, 145–169.

Dale V. H., Hemstrom M. and Franklin J. (1986) Modeling the long-term effects of disturbances on forest succession, Olympic Peninsula, Washington. *Canadian Journal of Forest Research* **16**, 56–67.

Dale V. H., Jager H. I., Gardner R. H. and Rosen A. E. (1988) Using sensitivity and uncertainty analyses to improve predictions of broad–scale forest development. *Ecological Modelling* **42**, 165-178.

Daniels R. F. (1976) Simple competition indices and their correlation with annual loblolly pine tree growth. *Forest Science* **22**, 454–456.

Drury W. H. and Nisbet I. C. T. (1973) Succession. *Journal of the Arnold Arboretum* **54**, 331-368.

Dunwiddie P. W. and LaMarche V. C. (1980) Dendrochronological characteristics of some native Australian trees. *Australian Forestry* **43**, 124–135.

Egler R. E. (1954) Vegetation science concepts. I. Initial floristic composition—A factor in old-field vegetation development. *Vegetatio* **4**, 412–468.

Ek A. R. and Monserud R. A. (1974) Trials with program FOREST: growth and reproduction simulation for mixed species even- or uneven-aged forest stands. In: *Growth Models for Tree and Stand Simulation* (ed. J. Fries), pp. 56–73. Research Notes 30, Department of Forest Yield Research, Royal College of Forestry, Stockholm.

Ellenberg H. (1953) Physiologisches und ökologisches Verhalten derselben Pflanzenarten (Physiological and ecological behaviour of the same plant species). *Berichte der Deutschen Botanischen Gesellschaft* **65**, 351–362.

Ellenberg H. (1954) Über einige Fortschritte der kausalen Vegetationskunde (Some advances in causal vegetation studies). *Vegetatio* **5/6**, 199–211.

Elton C. (1927) *Animal Ecology.* Sidgwick & Jackson, London. 209 pp.

Ernst W. (1978) Discrepancy between ecological and physiological optima of plant species. A re-interpretation. *Œcologia Plantarum* **13**, 175–188.

Fitzpatrick E. A. and Nix H. A. (1970) The climatic factor in Australian grassland ecology. In: *Australian Grasslands* (ed. R. M. Moore), pp. 3–26. Australian National University Press, Canberra.

Florence R. G. (1981) The biology of the eucalypt forest. In: *The Biology of Australian Plants* (eds J. S. Pate and A. J. McComb), pp. 147–180. University of Western Australia Press, Nedlands.

Florence R. G. (1982) The Australian Forests. In: *Handbook of Vegetation Science, Part 12* (ed. G. Jahn). Junk, The Hague.

Franklin J. F., Shugart H. H. and Harmon M. E. (1987) Tree death as an ecological process. *BioScience* **37**, 550–556.

Fritts H. C. (1976) *Tree Rings and Climate.* Academic Press, London xii + 567 pp.

Gardner R. H., Cale W. G. and O'Neill R. V. (1982) Robust analysis of aggregation error. *Ecology* **63**, 1771–1779.

Gauch H. G. and Whittaker R. H. (1972) Coenocline simulation. *Ecology* **53**, 446–451.

Gill A. M. (1975) Fire and the Australian flora: a review. *Australian Forestry* **38**, 4–25.

Gill A. M. (1978) Crown recovery of *Eucalyptus dives* following wildfire. *Australian Forestry* **41**, 207–214.

Gill A. M. (1981a) Coping with fire. In: *The Biology of Australian Plants* (eds J. S. Pate and A. J. McComb), pp. 65–87. University of Western Australia Press, Nedlands.

Gill A. M. (1981b) Adaptive responses of Australian vascular plant species to fires. In: *Fire and the Australian Biota* (eds A. M. Gill, R. H. Groves and I. R. Noble), pp. 243–271. Australian Academy of Science, Canberra.

Gleason H. A. (1917) The structure and development of the plant association. *Bulletin of the Torrey Botanical Club* **44**, 463–481.

Gleason H. A. (1926) The individualistic concept of the plant association. *Bulletin of the Torrey Botanical Club* **53**, 1–20.

Gleason H. A. (1927) Further views on the succession concept. *Ecology* **8**, 299–326.

Goodall D. W. (1963) The contiuum and the individualistic association. *Vegetatio* **11**, 297–316.

Green J. W. (1967) *Altitudinal variation in* Eucalyptus pauciflora. PhD thesis, Australian National University. 177 pp.

Green J. W. (1969a) Continuous measurements of radial variation in *Eucalyptus pauciflora* Sieb. ex Spreng. *Australian Journal of Botany* **17**, 191–198.

Green J. W. (1969b) Temperature responses in altitudinal populations of *Eucalyptus pauciflora* Sieb. ex Spreng. *New Phytologist* **68**, 399–410.

Griffith L. R. (1977) *Pattern and structure in snow gum* (Eucalyptus pauciflora) *understorey on the Brindabella Range.* MSc thesis, Deptartment of Forestry, Australian National University.

Grime J. P. (1977) Evidence for the existence of three primary strategies in plants and its relevance to ecological and evolutionary theory. *American Naturalist* **111**, 1169–1194.

Grime J. P. (1979) *Plant Strategies and Vegetation Processes.* John Wiley and Sons, Chichester.

Grubb P. J. (1977) The maintenance of species-richness in plant communities: the importance of the regeneration niche. *Biological Review* **52**, 107-145.

Grubb P. J. (1985) Plant populations and vegetation in relation to habitat, disturbance and competition: problems of generalization. In: *The Population Structure of Vegetation* (ed. J. White), pp. 595–621. Junk, Dordrecht.

Gunn R. H., Story R., Galloway R. W., Duffy P. J. B., Yapp G. A. and McAlpine J. F. (1969) *Lands of the Queanbeyan–Shoalhaven Area, A. C. T. and N. S. W.* Land Research Series No. 24, CSIRO, Melbourne.

Hall N., Johnston R. D. and Chippendale G. M. (1970) *Forest Trees of Australia.* Australian Government Publishing Service, Canberra. 334 pp. 2nd Edition.

Harcombe P. A. (1987) Tree life tables. *BioScience* **37**, 557–568.

Harwood C. E. (1980) Frost resistance of subalpine *Eucalyptus* species. I. Experiments using a radiation frost room. *Australian Journal of Botany* **28**, 587–589.

Hegyi F. (1974) A simulation model for managing jack-pine stands. In: *Growth Models for Tree and Stand Simulation* (ed. J. Fries), pp. 74–87. Research Notes 30, Department of Forest Yield Research, Royal College of Forestry, Stockholm.

Horn H. S. (1974) The ecology of secondary succession. *Annual Review of Ecology and Systematics* **5**, 25–37.

Horn H. S. (1975) Markovian properties of forest succession. In: *Ecology and evolution of communities* (eds M. L. Cody and J. M. Diamond), pp. 196–211. Harvard University Press, Cambridge Massachusetts.

Horn H. S. (1981) Succession. In: *Theoretical ecology. Principles and applications.* (ed. R. M. May), pp. 253–271. Blackwell, Oxford. 2nd edition.

Horn H. S., Shugart H. H. and Urban D. L. (1989) Simulators as models of forest dynamics. In: *Perspectives in Ecological Theory* (eds J. Roughgarden, R. M. May and S. A. Levin), pp. 256–267. Princeton University Press, Princeton.

Hughes T. P. (1984) Population dynamics based on individual size rather than age: a general model with a reef coral example. *American Naturalist* **123**, 778–795.

Hulst R. van (1979) On the dynamics of vegetation: Markov chains as models of succession. *Vegetatio* **40**, 3–14.

Hulst R. van (1980) Vegetation dynamics or ecosystem dynamics. Dynamic sufficiency in succession theory. *Vegetatio* **43**, 147–151.

Husch B., Miller C. I. and Beers T. W. (1982) *Forest Mensuration.* Wiley, New York. 402 pp. 3rd edition.

Huston M. A. and Smith T. M. (1987) Plant succession: life history and competition. *American Naturalist* **130**, 168–198.

Huston M. H., DeAngelis D. L. and Post W. M. (1988) New computer models unify ecological theory. *BioScience* **38**, 682–691.

Hutchinson G. E. (1958) Concluding remarks. *Cold Spring Harbor Symposium on Quantitative Biology* **22**, 415–427.

Jackson W. D. (1968) Fire, air, water and earth—an elemental ecology of Tasmania. *Proceedings of the Ecological Society of Australia* **3**, 9–16.

Jacobs M. R. (1955) *Growth Habits of the Eucalypts.* Forestry and Timber Bureau, Canberra. 262 pp.

Kac M. (1969) Some mathematical models in science. *Science* **166**, 695–699.

Keig G. and McAlpine J. R. (1974) *WATBAL: a computer system for the estimation and analysis of soil moisture regimes from simple climatic data.* Second edition. CSIRO Div. Land Use Research Technical Memorandum 74/4. 45 pp.

Keith H. (1982) *Investigation of the growth response of* Eucalyptus pauciflora *to climate and fire, using dendrochronology, in the Brindabella Range, A. C. T.* BSc Honours thesis, School of Geography, University of New South Wales.

Kercher J. R. and Axelrod M. C. (1984) A process model of fire ecology and succession in a mixed-conifer forest. *Ecology* **65**, 1725–1742.

Kessell S. R. (1979) *Gradient Modeling: Resource and Fire Management.* Springer–Verlag, New York. 432 pp.

Kessell S. R. (1988) *FIREPLAN Training Manual Version 2.2.* Gradient Modeling Pty Ltd, Perth.

Kessell S. R. (1990) An Australian geographical information and modelling system for natural area management. *International Journal of Geographical Information Systems* **4**, 333–362.

Kessell S. R. and Good R. B. (1985) Technological advances in bushfire management and planning. *Symposium on Natural Disasters in Australia: Technology in the Mitigation of Fire, Flood, Storm and Drought held in Sydney Australia, on 16–17 October 1985.* Australian Academy of Technological Sciences, Canberra. pp 323–344.

Kessell S. R. and Potter M. W. (1980) A quantitative succession model for nine Montana forest communities. *Environmental Management* **4**, 227–240.

Kessell R S., Potter M. W., Bevins C. D., Bradshaw L. and Jeske B. W. (1978) Analysis and application of forest fuels data. *Environmental Management* **2**, 347.

Lamb D. and Florence R. G. (1973) Soil moisture and the distribution of two eucalypts. *Australian Forestry* **36**, 138–145.

Landsberg J. J. (1986) *Physiological Ecology of Forest Production.* Academic Press, London. xvii + 198 pp.

Lee R. (1963) Evaluation of solar beam irradiation as a climatic parameter of mountain watersheds. *Colorado State University Hydrological Paper 2.* 50 pp.

Leemans R. and Prentice I. C. (1987) Description and simulation of tree-layer composition and size distributions in a primaeval *Picea–Pinus* forest. *Vegetatio* **69**, 147–156.

Levins R. (1966) The strategy of model building in population biology. *American Scientist* **54**, 421–431.

Liebig J. (1840) *Chemistry and its Application to Agriculture and Physiology.* Taylor and Walton, London.

Lindeman R. L. (1942) The trophic-dynamic aspect of ecology. *Ecology* **23**, 399–418.

Lindenmayer D. B., Norton T. W. and Tanton M. T. (in press) Differences between the effects of wildfire and clearfelling on the structure of montane ash forests of the Central Highlands of Victoria and its implications for fauna dependent on tree hollows. *Australian Forestry.*

Loehle C. (1983) Evaluation of theories and calculation tools in ecology. *Ecological Modelling* **19**, 239–247.

Loehle C. (1987) Errors of construction, evaluation, and inference: a classification of sources of error in ecological models. *Ecological Modelling* **36**, 297–314.

Lorimer C. G. (1983) Tests of age-independent competition indices for individual trees in natural hardwood stands. *Forest Ecology and Management* **6**, 343–360.

Luke R. H. and McArthur A. G. (1978) *Bushfires in Australia.* Australian Government Publishing Service, Canberra. 359 pp.

Mankin J. B., O'Neill R. V. and Shugart H. H. (1977) The importance of validation in ecosystem analysis. In: *New Directions in the Analysis of Ecological Systems Part 1* (ed. G. S. Innis), pp. 63–71. Simulation Council of America, La Jolla, California.

McAlpine J. R. (1971) Estimating pasture growth periods and droughts from simple water balance models. *Proceedings of the 11th Internation Grassland Congress, Surfers Paradise, Queensland.* pp 484–487.

McArthur A. G. (1967) *Fire behaviour in eucalypt forests.* Commonwealth of Australia Forestry and Timber Bureau Leaflet 107.

McCullagh P. and Nelder J. A. (1983) *Generalized Linear Models.* Chapman and Hall, London xiii + 261 pp.

McIntosh R. P. (1980) The relationship between succession and the recovery process in ecosystems. In: *The Recovery Process in Damaged Ecosystems* (ed. J. Cairns), pp. 11–62. Ann Arbor Science Publishers, Ann Arbor, Michigan.

McIntosh R. P. (1981) Sucession and ecological theory. In: *Forest Succession: Concepts and Application* (eds D. C. West, H. H. Shugart and D. B. Botkin), pp. 10–23. Springer–Verlag, New York.

Monserud R. A. (1986) Time-series analyses of tree-ring chronologies. *Forest Science* **32**, 349–372.

Moore A. D. (1989a) *Discrete Predictive Models of Vegetation Dynamics.* PhD Thesis, Australian National University.

Moore A. D. (1989b) On the maximum growth equation used in forest gap simulation models. *Ecological Modelling* **45**, 63–67.

Moore A. D. (1990) The semi-Markov process: a useful tool in the analysis of vegetation dynamics for management. *Journal of Environmental Management* **30**, 111–130.

Moore A. D. and Noble I. R. (1990) An individualistic model of vegetation stand dynamics. *Journal of Environmental Management* **31**, 61–81.

Moore J. A., Budelsky C. A. and Schlesinger R. C. (1973) A new index representing individual tree competitive status. *Canadian Journal of Forest Research* **3**, 495–500.

Morrow P. A. (1977) Host specificity of insects in a community of three co-dominant *Eucalyptus* species. *Australian Journal of Ecology* **2**, 89–106.

Morrow P. A. and LaMarche V. C. (1978) Tree ring evidence for chronic insect suppression of productivity in subalpine *Eucalyptus. Science* **201**, 1244–1246.

Mueller-Dombois D. and Ellenberg H. (1974) *Aims and Methods of Vegetation Ecology.* Wiley, New York. 547 pp.

Naveh Z. and Liebermann A. S. (1985) *Landscape Ecology.* Springer–Verlag, New York.

Nelder J. A. and Wedderburn R. W. M. (1972) Generalized Linear Models. *Journal of the Royal Statistical Society Series A* **135**, 370–384.

Noble I. R. (1984) Mortality of lignotuberous seedlings of *Eucalyptus* species after an intense fire in montane forest. *Australian Journal of Ecology* **9**, 47–50.

Noble I. R. and Slatyer R. O. (1977) Post fire succession of plants in Mediterranean ecosystems. In: *Proceedings of the Symposium on the Environmental Consequences of Fire and Fuel Management in Mediterranean Ecosystems* (eds H. A. Mooney and C. E. Conrad), pp. 27–36. USDA Forest Service General Technical Report WO-3.

Noble I. R. and Slatyer R. O. (1980) The use of vital attributes to predict successional changes in plant communities subject to recurring disturbances. *Vegetatio* **43**, 5–21.

Noble I. R. and Slatyer R. O. (1981) Concepts and models of succession in vascular plant communities subject to recurrent fire. In: *Fire and the Australian Biota* (eds A. M. Gill, R. H. Groves and I. R. Noble), pp. 311–335. Australian Academy of Science, Canberra.

Noble I. R., Shugart H. H. and Schauer J. S. (1980a) *A description of BRIND, a computer model of succession and fire response of the high-altitude* Eucalyptus *forests of the Brindabella Range, Australian Capital Territory.* Oak Ridge National Laboratory, Oak Ridge, Tennessee. ORNL/TM-7041. 96 pp.

Noble I. R., Bary G. A. and Gill A. M. (1980b) McArthur's fire danger meters expressed as equations. *Australian Journal of Ecology* **5**, 201–204.

Noble I. R., Moore A. D. and Strasser M. J. (1988) Stand structure and the prediction of vegetation dynamics. In: *Diversity and pattern in plant communities* (eds M. J. A. Werger and J. H. Willems), pp. 263–275. SPB Academic Publishing, The Hague.

O'Dowd D. J. and Gill A. M. (1984) Predator satiation and site alteration following fire: mass reproduction of alpine ash (*Eucalyptus delegatensis*) in southeastern Australia. *Ecology* **65**, 1052–1066.

Odum E. P. (1969) The strategy of ecosystem development. *Science* **164**, 262–270.

Odum E. P. (1971) *Fundamentals of Ecology.* Saunders College Publishing, Philadelphia.

Ogden J. (1978) On the dendrochronological potential of Australian trees. *Australian Journal of Ecology* **3**, 339–356.

O'Loughlin E. M., Cheney N. P. and Burns J. (1982) The Bushrangers experiment: hydrological response of a eucalypt catchment to fire. In: *The First National Symposium on Forest Hydrology, Melbourne, May 1982.*

O'Neill R. V., DeAngelis D. L., Waide J. B. and Allen T. F. H. (1986) *A hierarchical concept of ecosystems.* Princeton University Press, Princeton.

Pastor J. and Post W. M. (1984) Calculating Thornthwaite and Mather's actual evapotranspiration using an approximating function. *Canadian Journal of Forest Research* **14**, 466–467.

Pastor J. and Post W. M. (1986) Influence of climate, soil moisture, and succession on forest carbon and nitrogen cycles. *Biogechemistry* **2**, 3–27.

Peet R. K. and Christensen N. L. (1987) Competition and tree death. *BioScience* **37**, 586–595.

Penridge L. K. and Walker J. (1986) Effect of neighbouring trees on eucalypt growth in a semi-arid woodland in Australia. *Journal of Ecology* **74**, 925–936.

Penridge L. K., Walker J., Sharpe P. J. H., Spence D., Wu H. and Zou G. (1987) *RESCOMP: a resource competition model to simulate the dynamics of vegetation cover.* CSIRO Division Water and Land Resources Technical Memorandum 87/5.

Pickett S. T. A. (1976) Succession: an evolutionary interpretation. *American Naturalist* **110**, 107–119.

Pickett S. T. A., Collins S. L. and Armesto J. J. (1987a) A hierarchical consideration of causes and mechanisms of succession. *Vegetatio* **69**, 109–114.

Pickett S. T. A., Collins S. L. and Armesto J. J. (1987b) Models, mechanisms and pathways of succession. *The Botanical Review* **53**, 335–371.

Pomeroy L. R., Hargrove E. C. and Alberts J. J. (1988) The ecosystem perspective. In: *Concepts of Ecosystem Ecology* (eds L. R. Pomeroy and J. J. Alberts), pp. 1–17. Springer–Verlag, New York.

Pryor L. D. (1960) The 'ashbed' effect in eucalypt ecology. *Institute of Foresters of Australia Newsletter* **2**, 23–26.

Purdie R. W. and Slatyer R. O. (1976) Vegetation succession after fire in sclerophyll woodland communities in south-eastern Australia. *Australian Journal of Ecology* **1**, 223–236.

Shipley B. and Peters R. H. (1990) A test of the Tilman model of plant strategies: relative growth rate and biomass partitioning. *American Naturalist* **136**, 139–153.

Schoener T. W. (1989) The ecological niche. In: *Ecological Concepts. The Contribution of Ecology to an Understanding of the Natural World* (ed. J. M. Cherrett), pp. 79–113. Blackwell, Oxford.

Shugart H. H. (1984) *A Theory of Forest Dynamics. The Ecological Implications of Forest Succession Models.* Springer–Verlag, New York. xiv + 277 pp.

Shugart H. H. (1987) Dynamic ecosystem consequences of tree birth and death patterns. *BioScience* **37**, 596–602.

Shugart H. H. and Noble I. R. (1981) A computer model of succession and fire response of the high-altitude *Eucalyptus* forest of the Brindabella Range, Australian Capital Territory. *Australian Journal of Ecology* **6**, 149–164.

Shugart H. H. and O'Neill R. V. (1979) *Systems Ecology.* Dowden, Hutchinson and Ross, Stroudsberg. 368 pp.

Shugart H. H. and Urban D. L. (1988) Scale, synthesis and ecosystem dynamics. In: *Concepts of Ecosystem Ecology* (eds L. R. Pomeroy and J. J. Alberts), pp. 279–289. Springer–Verlag, New York.

Shugart H. H. and West D. C. (1977) Development of an Appalachian deciduous forest succession model and its application to assessment of the impact of the chestnut blight. *Journal of Environmental Management* **5**, 161–179.

Shugart H. H. and West D. C. (1979) Size and pattern of simulated forest stands. *Forest Science* **25**, 120–122.

Shugart H. H. and West D. C. (1980) Forest succession models. *BioScience* **30**, 308–313.

Shugart H. H., Crow T. R. and Hett J. M. (1973) Forest succession models: a rationale and methodology for modeling forest succession over large regions. *Forest Science* **19**, 203–212.

Silvertown J. and Law R. (1987) Do plants need niches? Some recent developments in plant community ecology. *Trends in Ecololgy and Evolution* **2**, 24–26.

Smith T. M. and Urban D. L. (1988) Scale and resolution of forest structural pattern. *Vegetatio* **74**, 143–150.

Sokal R. R. and Rohlf F. J. (1981) *Biometry.* W. H. Freeman and Co., San Francisco. 2nd edition.

Solomon D. L. (1979) On a paradigm for mathematical modeling. In: *Contemporary Quantitative Ecology and Related Ecometrics* (eds G. P. Patil and M. L. Rosenzweig), pp. 231–250. International Co-operative Publishing House, Fairland.

Specht R. L. (1981a) Foliage projective cover and standing biomass. In: *Vegetation Classification in Australia.* (eds A. N. Gillison and D. J. Anderson). CSIRO, Canberra.

Specht R. L. (1981b) Growth indices—their rôle in understanding the growth, structure and distribution of Australian vegetation. *Oecologia* **50**, 347-356.

Stage A. R. (1976) An expression for the effect of aspect, slope and habitat type on tree growth. *Forest Science* **22**, 457–460.

Stephens G. R. and Waggoner P. E. (1970) *The forests anticipated from 40 years of natural transitions in mixed hardwoods.* Bulletin of the Connecticut Agricultural Experiment Station, New Haven, no. 707.

Swartzman G. L. (1979) Evaluation of ecological simulation models. In: *Contemporary Quantitative Ecology and Related Ecometrics* (eds G. P. Patil and M. L. Rosenzweig), pp. 295–318. International Co-operative Publishing House, Fairland.

Swift L. W. (1976) Algorithm for solar radiation on mountain slopes. *Water Resources Research* **12**, 108–112.

Tajchman S. J. and Lacey C. J. (1986) Bioclimatic factors in forest site potential. *Forest Ecology and Management* 14, 211–218.

Talsma T. (1983) Soils of the Cotter catchment area, A. C. T.: distribution, chemical and physical properties. *Australian Journal Soil Research* **21**, 241–255.

Tansley A. G. (1935) The use and abuse of vegetational concepts and terms. *Ecology* **16**, 284–307.

Thornthwaite C. W. and Mather J. R. (1957) Instructions and tables for computing potential evaoptranspiration and the water balance. *Publications in Climatology* **10**, 185–311.

Tibbits W. N. and Reid J. B. (1987) Frost resistance in *Eucalyptus nitens* (Dean & Maiden) Maiden: Physiological aspects of hardiness. *Australian Journal of Botany* **35**, 235–250.

Tilman D. (1982) *Resource Competition and Community Structure.* Princeton University Press, Princeton.

Tilman D. (1985) The resource–ratio hypothesis of plant succession. *American Naturalist* **125**, 827-852.

Tilman D. (1988) *Plant Strategies and the Dynamics and Structure of Plant Communities.* Princeton University Press, Princeton.

Urban L D., O'Neill R. V. and Shugart H. H. (1987) Landscape ecology. *BioScience* **37**, 117–127.

Usher M. B. (1979) Markovian approaches to ecological succession. *Journal of Animal Ecological* **48**, 413–426.

Usher M. B. (1981) Modelling ecological succession with particular reference to Markovian models. *Vegetatio* **46**, 11–18.

Van Wagner C. E. (1973) Height of crown scorch in forest fires. *Canadian Journal of Forest Research* **3**, 373–378.

Waldrop T. A., Buckner E. R., Shugart H. H. and McGee C. E. (1986) FORCAT: A single tree model of stand development following clearcutting on the Cumberland Plateau. *Forest Science* **32**, 297–317.

Walker L. R. and Chapin F. S. (1987) Interactions among processes controlling successional change. *Oikos* **50**, 131–135.

Waring R. H. (1989) Ecosystems: fluxes of matter and energy. In: *Ecological Concepts. The Contribution of Ecology to an Understanding of the Natural World* (ed. J. M. Cherrett), pp. 17–41. Blackwell, Oxford.

Watt A. S. (1947) Pattern and process in the plant community. *Journal of Ecology* **35**, 1–22.

Webster J. R. (1979) Hierarchical organization of ecosystems. In: *Theoretical Systems Ecology: Advances and Case Studies* (ed. E. Halfon), pp. 119–129. Academic Press, New York.

Weiner J. and Thomas S. C. (1986) Size variability and competition in plant monocultures. *Oikos* **47**, 211–222.

Weinstein D. A., Shugart H. H. and West D. C. (1982) The long-term nutrient retention properties of forest ecosystems: a simulation investigation. ORNL/TM-8472. Oak Ridge National Laboratory, Oak Ridge, Tennessee. 145pp.

White J. H. (1980) Demographic factors in populations of plants. In: *Demography and Evolution in Plant Populations* (ed. O. T. Solbrig), pp. 21–48. Blackwell, Oxford xiv + 222 pp.

Whittaker R. H. (1953) A consideration of climax theory: the climax as a population and pattern. *Ecological Monographs* **23**, 41–73.

Whittaker R. H. (1967) Gradient analysis of vegetation. *Biological Review* **42**, 207–264.

Whittaker R. H. (1978) Direct gradient analysis. In: *Ordination of Plant Communities* (ed. R. H. Whittaker), pp. 7–50. W. Junk, The Hague.

Whittaker R. H., Levin S. A. and Root R. B. (1973) Niche, habitat and ecotope. *American Naturalist* **107**, 321–338.

Williams J. E. (1989) *Ecology of sub-alpine eucalypt forests in relation to environmental gradients.* PhD Thesis, Australian National University.

Wu H., Sharpe P. J., Walker J. and Penridge L. K. (1985) Ecological field theory: a spatial analysis of resource interference among plants. *Ecological Modelling* **29**, 215–243.

Appendix A

Field data information

Information about field data sites (0.1 ha) used for comparison with BRIND predictions in Chapters 3 and 8. Sites are sorted in order of ascending altitude. Grid references are given to the nearest 50 m on the Australian Metric Grid. Geology type describes parent material: S = sediments, G = granites. The biomass values given were calculated using the equation in BRIND (Chapter 2), adjusted for plot size.

Grid E (m)	Grid N (m)	Altitude (m)	Aspect (°)	Slope (°)	Geol type	Number of trees	Biomass (t/ha)
670500	6084250	660	210	30	S	106	114
668300	6082200	660	60	15	S	116	169
670750	6084650	670	340	27	S	103	116
670600	6084600	670	260	32	S	67	166
669450	6082850	670	135	29	S	70	116
671350	6085750	675	60	27	S	70	156
670150	6083500	680	45	43	S	81	51
671100	6085400	680	155	28	S	53	286
670250	6083850	680	255	26	S	75	154
668450	6083100	690	305	35	S	82	120
669650	6083250	700	30	26	S	175	209
668900	6089750	700	290	11	S	95	255
669150	6080550	710	215	32	S	57	156
671300	6085150	710	160	26	S	64	192
669550	6083000	730	130	30	S	109	180
668500	6090000	740	115	22	S	151	189

Grid E (m)	Grid N (m)	Altitude (m)	Aspect (°)	Slope (°)	Geol type	Number of trees	Biomass (t/ha)
671150	6085650	740	125	29	S	48	219
668800	6080600	740	295	15	S	48	213
667850	6089950	750	10	19	S	113	271
671400	6085450	750	55	25	S	178	141
669850	6081650	760	300	23	S	63	175
671750	6085300	760	210	30	S	46	201
668250	6090050	770	40	15	S	75	299
668400	6082300	770	295	26	G	100	180
668500	6082150	780	45	19	G	63	187
667600	6086200	800	290	18	S	77	142
668150	6082150	810	105	22	G	79	149
667200	6085300	820	95	20	S	74	192
667950	6086500	830	205	26	S	75	170
666900	6085450	840	200	27	S	65	193
666800	6084600	860	295	12	S	60	206
666500	6086500	890	80	21	S	90	197
668200	6081750	905	260	18	G	71	340
666400	6084200	920	41	25	S	195	287
666250	6086750	930	230	21	S	53	129
664400	6080100	940	120	13	G	51	313
666450	6085100	950	315	22	S	121	271
666250	6088250	950	200	19	S	94	402
664500	6079900	950	330	19	G	115	226
666500	6085050	960	110	25	S	37	430
667200	6088050	960	55	28	S	110	217
666400	6087850	980	30	22	S	54	305
664900	6079500	990	70	25	G	25	442
666100	6084800	1040	190	18	S	14	478
665700	6087200	1050	105	18	S	52	243
665000	6085050	1050	280	18	S	69	261
664900	6085250	1060	30	28	S	51	209
665450	6087300	1070	340	18	S	63	377
664900	6080800	1070	230	15	S	50	223
665900	6084800	1080	120	22	S	43	365
667600	6084100	1090	320	8	S	115	326
665150	6086250	1100	120	13	S	95	397
664900	6084300	1110	10	16	S	61	689
663500	6077450	1130	205	17	G	45	452

Grid E (m)	Grid N (m)	Altitude (m)	Aspect (°)	Slope (°)	Geol type	Number of trees	Biomass (t/ha)
664000	6077550	1150	320	13	G	74	292
664600	6084150	1150	10	22	S	71	322
664400	6084200	1175	200	17	S	146	239
664550	6085650	1180	130	24	S	44	166
664400	6085650	1190	205	20	S	98	236
662350	6079700	1210	120	20	S	75	264
662550	6080200	1210	355	23	S	50	276
663100	6081050	1220	110	28	S	49	313
662250	6079450	1220	20	21	S	111	249
662900	6079950	1220	180	25	S	55	625
662900	6080700	1230	330	20	S	92	216
663150	6080650	1230	30	15	S	130	223
662100	6079300	1230	220	20	S	73	231
663000	6081600	1255	95	21	S	68	388
662100	6078800	1260	25	22	G	31	368
664750	6081850	1270	150	22	G	105	217
662300	6079100	1270	325	18	S	76	185
663850	6082850	1290	270	18	S	31	222
664100	6082650	1330	200	9	S	112	274
664000	6082850	1330	280	14	S	217	196
664250	6083150	1340	45	24	S	80	225
660650	6073150	1340	50	16	S	170	209
664250	6082850	1340	100	19	S	176	342
661050	6075000	1390	30	22	S	80	230
660800	6075100	1390	250	14	S	164	270
661150	6076600	1395	320	13	S	117	192
660350	6072650	1395	265	16	S	81	111
660950	6074850	1400	280	16	S	38	271
660850	6073800	1400	150	22	S	191	254
660350	6072250	1410	250	22	S	165	105
660250	6071500	1420	300	27	S	78	118
660850	6074150	1440	100	23	S	175	140
661200	6075850	1440	125	27	S	188	276
660350	6073300	1440	45	23	S	173	120
661100	6057800	1450	145	21	G	86	365
661750	6065350	1450	235	11	G	181	43
661700	6064800	1460	300	12	G	156	183
661100	6057750	1460	10	22	G	80	353

Grid E (m)	Grid N (m)	Altitude (m)	Aspect (°)	Slope (°)	Geol type	Number of trees	Biomass (t/ha)
659950	6069750	1520	20	17	S	98	253
660100	6070900	1530	150	11	S	167	200
660050	6071000	1530	315	21	S	59	83
659800	6069500	1550	290	15	S	71	107
660700	6064500	1560	350	6	G	110	188
659900	6067850	1605	245	16	S	116	151
659850	6068250	1620	260	17	S	213	196
660350	6067250	1640	20	9	S	49	110
661900	6060850	1640	45	18	G	99	120
661300	6063400	1650	110	15	G	81	118
660700	6067150	1650	15	14	S	192	88
660850	6066350	1660	115	20	S	108	42
661800	6062300	1660	220	10	G	72	58
660200	6065800	1670	215	12	S	81	145

Appendix B

Site parameters for simulations

Parameters describing the sites at 24 altitude–aspect sites in the Brindabella Range used in NEWBRIND simulations (Chapter 8).

The 20° slope angle represents an average slope in the Brindabella Range (calculated from field observations). SOILQ is the maximum biomass achievable at the site (it is known as MaxBiom within the code of NEWBRIND, see Appendix C). FieldCap is the field capacity of the soil in millimetres; it is greater for sheltered slopes with deeper soils and less for exposed slopes with shallower soils (ANU Forestry 1973).

Site	Altitude (m)	Aspect (°)	Slope (°)	SOILQ (t/ha)	FieldCap (mm)
1	660	45	20	755	65
2	660	135	20	825	75
3	660	225	20	776	65
4	660	315	20	687	55
5	800	45	20	744	65
6	800	135	20	791	75
7	800	225	20	726	65
8	800	315	20	690	55
9	1000	45	20	715	65
10	1000	135	20	751	75
11	1000	225	20	696	65
12	1000	315	20	672	55

Site	Altitude (m)	Aspect (°)	Slope (°)	SOILQ (t/ha)	FieldCap (mm)
13	1200	45	20	687	65
14	1200	135	20	707	75
15	1200	225	20	662	65
16	1200	315	20	644	55
17	1400	45	20	633	65
18	1400	135	20	640	75
19	1400	225	20	600	65
20	1400	315	20	590	55
21	1600	45	20	580	65
22	1600	135	20	580	75
23	1600	225	20	564	65
24	1600	315	20	563	55

Appendix C

Listing of NEWBRIND

Full listing of the C source code of NEWBRIND. The program has been run under a number of operating systems: MS-DOS, VAX/VMS, OS/2 and SunOS (UNIX).

Header files are listed first. They contain many defined symbols which are used by the program.

User control of the program is handled by `SetVars()`. The user can change the value of many program variables from the command line as well as setting simulation parameters. Some variables are boolean flags which control program execution, e.g., to prevent establishment and mortality so that predicted rates of growth can be examined in detail (see listing of `vstructs.h`).

C.1 Header files

```
/*********************************************************************************

  globals.h

  Header file for NEWBRIND with global symbols and constants

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601
    Australia

  Description:

    This header file contains some of the global variables and symbols
    used in NEWBRIND.  They are described as they are declared below.

*********************************************************************************/

/**
  Symbols to determine machine type and graphics option:  tested as true
  or false
```

```
**/
#ifdef BC

# define  VMS        0  /* VAX/VMS C */
# define  TC         1  /* Turbo C */
# define  ZTC        0  /* Zortech C(++) */
# define  UNIX       0  /* Sun (non-ANSI) C compiler */

# define  GRAPHICS   1  /* graphics option */

#else

# define  VMS        0  /* VAX/VMS C */
# define  TC         0  /* Turbo C */
# define  ZTC        1  /* Zortech C(++) */
# define  UNIX       0  /* Sun (non-ANSI) C compiler */

# define  GRAPHICS   0  /* graphics option */

#endif /* BC */

/* Symbol for the Borland (Turbo C) graphics driver path */
#if TC && GRAPHICS
#define  GrDriverPath  "C:\\LOCAL\\TC\\"
#endif

/* Symbols for boolean values */
#define NO  0
#define YES 1

/* Symbols for program execution. Returned by SetVars() and Grow() */
#define Stop 0
#define Run  1

/* Symbols for the errz() function, which exits on FATAL error */
#define FATAL   1
#define WARNING 2

/**
  Global variables.  MAIN is only defined in newbrind.c and serves
  to allow these to "reside" in that unit and be external to others.
**/
#ifdef MAIN
# define Global
#else
# define Global extern
#endif

Global int Burn,       /* Boolean: is this a fire year? */
           Year,       /* Current year of simulation */
           RunYears;   /* How many years to run in this simulation */

Global double RunningTotalBiomass; /* Used for calculation of mean plot
                                      biomass over a number of years */

Global char ParamFileName[64]; /* Pathname of species parameter file */

/**
  The antilogit function, used in Birth() and Fire()
**/
#define AntiLogit( lp ) ( exp( lp ) / (1.0 + exp( lp )) )

/**
  Things for life tables
**/
#define MAXSPEC 19    /* Maximum number of species with life tables */
#define MAXDCLASS 41  /* Maximum number of DBH classes */
#define MAXFATE 5     /* Maximum number of fates (shown below) */
```

```
/**
  The five fates recorded in the life table for trees of each
  species are:
**/
#define stayedput    0
#define tonextclass  1
#define firekill     2
#define sizeindmort  3
#define suppresskill 4

/**
  Declarations of functions in this program.  Most of the names are
  self-explanatory.
**/
#if ! UNIX  /* The Sun C compiler won't accept ANSI declarations */

void    Birth( double, double, int );
double  BrRand( int );  /* Pseudo-random number generator */
void    Climate( double *, double *, double *, int * );
void    Death( void );
void    DiscardCurLine( void );  /* Discard input line */
void    errz( unsigned, char * );  /* Report errors, perhaps stop program */
void    Fire( void );
int     Grow( double, double );
double  HeatRatio( int, double, double, double );  /* Calculate adjustment of
                                                      evap. for aspect */
void    Help( void );
void    InitLifeTable( void );
void    LifeTable( void );
double  NormDev( double, double, int, int );  /* Provide random normal nos. */
void    Output( unsigned );  /* General output handler */
void    PrintVar( void );  /* Print program variables */
void    RandSSet( void );  /* Set random no. seeds */
int     ReadDouble( FILE *, double *, char * );  /* Read a double (float) */
int     ReadInt( FILE *, int *, char * );  /* Read an integer */
int     ReadLong( FILE *, long *, char * );  /* Read a long integer */
int     ReadString( FILE *, char *, char * );  /* Read a string */
void    ReportFires( void );
void    ReportTime( void );
void    SaveFire( double, double, int, int );
int     SetVars( void );  /* Set program variables */
void    SiteEnvt( void );  /* Set site environment */
void    SiteList( void );
int     SiteSummary( void );  /* Summarise site information */
void    SpParams( void );  /* Set species parameters */
# if ! (TC || ZTC)
int     strnicmp( char *, char *, int ); /* Case-insensitive limited-length
                                            string comparison: already in
                                            Turbo and Zortech C */
# endif
void    StrtSite( void );  /* Set starting site for simulation */
void    TogglePrompts( int );  /* Turn prompting on and off */

#else

void    Birth( );
double  BrRand( );
void    Climate( );
void    Death( );
void    DiscardCurLine( );
void    errz( );
void    Fire( );
int     Grow( );
double  HeatRatio( );
void    Help( );
void    InitLifeTable( );
void    LifeTable( );
```

```
double  NormDev( );
void    Output( );
void    PrintVar( );
void    RandSSet( );
int     ReadDouble( );
int     ReadInt( );
int     ReadLong( );
int     ReadString( );
void    ReportFires( );
void    ReportTime( );
void    SaveFire( );
int     SetVars( );
void    SiteEnvt( );
void    SiteList( );
int     SiteSummary( );
void    SpParams( );
void    StrtSite( );
void    TogglePrompts( );

#endif


/*******************************************************************************

  trees.h

  Definitions of the species and tree data stuctures and declarations
  of the functions which operate on them.

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601

*******************************************************************************/

typedef struct SpStruct {         /* Structure for each species */
  double B2;                      /* B2 in DBH-height eqn       */
  double B3;                      /* B3  "  "    "     "        */
  double SizeIndMort;             /* Size independent mortality */
  double GMean;                   /* Growth param. mean         */
  double GSD;                     /* Growth param. SD           */
  double MoistMin;                /* Min. moiture index toler'd */
  double BirthConst;              /* Birth prob'y const. term   */
  double BirthTemp[3];            /* Birth prob'y temp terms    */
  double BirthMoist[2];           /* Birth prob'y moist terms   */
  double BirthTempMoist;          /* Birth prob'y temp-mois term*/
  double BirthSediments;          /* Birth prob'y sediments term*/
  double BirthGranites;           /* Birth prob'y granites term */
  double SeedSurv;                /* Annual seed survival prob. */
  double SeedDisp;                /* Seed dispersal factor      */
  double SeedDBH;                 /* Min. DBH for seed          */
  double PlotLowerBound;          /* Lower bound in plots       */
  double PlotUpperBound;          /* Upper bound in plots       */
  double Seeds;                   /* Contribution to seed pool  */
  double CumSeeds;                /* Cumul've seed pool contrib */
  double Fire2Const;              /* Fire stage 2 const. term   */
  double Fire2DBH;                /* Fire stage 2 DBH term      */
  double GrowthForm;              /* Ratio of bole to tree ht   */
  double Biomass;                 /* Total biomass of species   */
  double DBHMax;                  /* Max. DBH                   */
  double HeightMax;               /* Max. Height                */
  double HtB;                     /* Coeff. used to calc. height*/

  /**
```

```
    Temporary storage for G values of trees that die
    with DBH > 5 cm
  **/
  double SumG;
  int nG;

  int MaxFrosts;                 /* Max. frosts for estab't    */
  int NTrees;                    /* Number of trees            */
  int NSuprs;                    /* Number of suppressed trees */
#if GRAPHICS
  int GraphColour;               /* Colour of species' symbols */
  int GraphShape;                /* Diamond or Square          */
#endif
  int LifeTable[MAXDCLASS][MAXFATE];
                                 /* This species' life table   */
  struct SpStruct *Next;         /* Pointer to next species    */
  struct TrStruct *Tree1;        /* Pointer to 1st tree        */
  char Name[26];                 /* Name of species            */
  char ShortName[7];             /* Short name for plotting    */
  char PlotSym;                  /* Plot symbol                */
  unsigned HasLigSdlgs: 1;       /* A lignotuberous spcies?    */
  unsigned FireResistant: 1;     /* Fire resistant species?    */
  unsigned GermNeedFire: 1;      /* Fire needed for germ'n?    */
  unsigned GermFireEnhanced: 1;  /* Enhanced germ after fire?  */
  unsigned WellDispSeeds: 1;     /* Well dispersed seeds?      */
  unsigned GotMatureTrees: 1;    /* Guess what this means      */
} Species;

typedef struct TrStruct {        /* Structure for each tree    */
  double G;                      /* Growth param. this tree    */
  double DBH;                    /* DBH of this tree           */
  double OldDBH;                 /* DBH last year of tree      */
  double Basalarea;              /* Current basal area         */
  double Height;                 /* Height of this tree        */
  double LeafArea;               /* Leaf area of this tree     */
  double Biomass;                /* Biomass of this tree       */
  double Vigour;                 /* Vigour in fire response    */
#if GRAPHICS
  double GraphHeight;            /* Height of tree at last plot*/
  int XPos;                      /* X-position of this tree    */
#endif
  int Age;                       /* Age of this tree           */
  int SuppressYrs;               /* No. of consecutive supp yr */
  struct TrStruct *Next;         /* Pointer to next tree       */
} Tree;

#if ! UNIX  /* The Sun UNIX compiler doesn't like ANSI declarations */

double Biomass( void );
double LeafArea( void );
double TotalPlotBiomass( void );
int NSpecies( void );
int TotNTrees( void );
Species *CreateSpecies( void );
Species *FirstSpecies( void );
Species *KillSpecies( void );
Species *NextSpecies( void );
Tree *CreateTree( void );
Tree *FirstTree( void );
Tree *NextTree( void );
Tree *KillTree( void );
void AddDBH( double );
void KillAll( void );

#else

double Biomass( );
double LeafArea( );
```

```
double TotalPlotBiomass( );
int NSpecies( );
int TotNTrees( );
Species *CreateSpecies( );
Species *FirstSpecies( );
Species *KillSpecies( );
Species *NextSpecies( );
Tree *CreateTree( );
Tree *FirstTree( );
Tree *NextTree( );
Tree *KillTree( );
void AddDBH( );
void KillAll( );

#endif


/*****************************************************************************
  vstructs.h

  Definitions of variables-structures, and shortcut names for reference

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601

  Description:

    Each structure is used to represent a variable in the program.
    It contains an eight-character name for it and a value of it.
    SetVars() initialises these and uses them to allow the user to
    change the value by specifying the name followed by a new value.

    For example, the array RYears[] of IntVars has for RYears[0]:

      { "StrtYear", 0 }

    so the user can change its value by typing StrtYear 100 (for
    example).  This file contains the line:

      #define StrtYear RYears[0].IVal

    so that value (set to 100 or whatever) is easily referred to
    within the program code by the symbol StrtYear.

    The IOUnit structures are more complex (see below).

*****************************************************************************/

/* Integers */
typedef struct {
  char Name[9];
  int IVal;
} IntVar;

/* Floats (doubles) */
typedef struct {
  char Name[9];
  double FVal;
} FloatVar;

/* Functions */
typedef struct {
  char Name[9];
```

```
  void (*FuncPtr)();
} FuncVar;


/* Flags */
typedef struct {
  char Name[9];
  int FlVal;
} FlagVar;

/* I/O units */
typedef struct {
  char Name[9];
  enum { Reading, Writing } Direction;  /* Reading or writing? */
  FILE *CurPtr;                         /* Current pointer */
  FILE *DfltPtr;                        /* Default pointer: the unit is re-
                                           turned to this when the user
                                           closes a file */
  char Filename[32];                    /* Current filename for this unit */
} IOUnitVar;

/**
  The symbol SETVARS is only defined in setvars.c where these
  arrays "reside" (and where they are initialised).  It includes
  this file to get the definitions above.  Other files including
  this one get the extern declarations of the arrays given below.
**/
#ifndef SETVARS

extern IntVar    RYears[];
extern FlagVar   RunFlags[];
extern FloatVar  Envt[];
extern char      Geology[];
extern IOUnitVar IOUnits[];

#endif

#define StrtYear RYears[ 0].IVal  /* Starting year of this simulation run,
                                     reset after each run */
#define PLinePlt RYears[ 1].IVal  /* Period between line plots (sim'd years) */
#define PSiteLst RYears[ 2].IVal  /* Period between site listings */
#define PRepFire RYears[ 3].IVal  /* Period between fire reports */
#define LastFrYr RYears[ 4].IVal  /* Last year in which a fire occurred,
                                     reset when a fire occurs */
#define PageWdth RYears[ 5].IVal  /* Page width for line printer plots */
#define LeadTime RYears[ 6].IVal  /* Lead time of sim'n before site summaries */
#define PRepTime RYears[ 7].IVal  /* Period between real time reports */
#define YSiteSum RYears[ 8].IVal  /* Year of site summary after lead time,
                                     or after last fire if PrFire > 0 */
#define Replicat RYears[ 9].IVal  /* No. of replicate for each simulation,
                                     incremented in main() */
#define FireYr1  RYears[10].IVal  /* Burn plot this year (0 for none) */
#define FireYr2  RYears[11].IVal  /* ditto */

#define NRYears 12                /* Size of RYears array */

/* Flags (defaults shown in brackets) */
#define BatchFlag    RunFlags[ 0].FlVal  /* Batch run with no prompts [0] */
#define BirthFlag    RunFlags[ 1].FlVal  /* With or w/o establishment [1] */
#define SIMortFlag   RunFlags[ 2].FlVal  /* With or w/o SI mortality [1] */
#define SuMortFlag   RunFlags[ 3].FlVal  /* With or w/o suppr. mort'y [1] */
#define LifeTablFlag RunFlags[ 4].FlVal  /* Generate life tables [0] */
#define SeedDispFlag RunFlags[ 5].FlVal  /* Add seed dispersal to species
                                            seed pools [1] */
#define NewSupprFlag RunFlags[ 6].FlVal  /* Use new suppr mortality [0] */
#define ClimGrowFlag RunFlags[ 7].FlVal  /* Climate affects growth [1] */
#define Graphics     RunFlags[ 8].FlVal  /* Show graphics */
```

```
#define NRunFlags 9                    /* Size of RunFlags array */

/* Floating point variables */
#define PlotArea  Envt[ 0].FVal  /* Plot area in sq metres */
#define Altitude  Envt[ 1].FVal  /* Altitude of plot (m) */
#define Aspect    Envt[ 2].FVal  /* Aspect of plot (degrees) */
#define Slope     Envt[ 3].FVal  /* Slope of plot (degrees) */
#define CompnIntensity  Envt[ 4].FVal  /* Intensity of competition (known
                                          as k to the user) */
#define CClWdth   Envt[ 5].FVal  /* Width of DBH classes for aggregating
                                    trees to calc 1-sided competiton */
#define SuprMort  Envt[ 6].FVal  /* Annual mortality of suppressed trees */
#define SurvProp  Envt[ 7].FVal  /* Proportion of trees surviving suppression
                                    and fire which also survive SI mortality
                                    at max. age */
#define FieldCap  Envt[ 8].FVal  /* Soil field capacity (mm) */
#define MaxFuel   Envt[ 9].FVal  /* Maximum fine fuel level on plot (t/ha) */
#define FuelDeco  Envt[10].FVal  /* Fuel decomposition constant (1/yr) */
#define FDIMean   Envt[11].FVal  /* Mean fire danger index of site (wildfire) */
#define FDIStDv   Envt[12].FVal  /* SD of FDI (wildfire) */
#define FireProb  Envt[13].FVal  /* Annual probability of wildfire */
#define SeedProd  Envt[14].FVal  /* Power to which biomass is raised in
                                    calculating seed pool contributions */
#define MaxBiom   Envt[15].FVal  /* Max. biomass on plot (t/ha)
                                    ===>> This is SOILQ in BRIND <<=== */
#define MinMMI    Envt[16].FVal  /* Value of MMI moisture index when MI is
                                    at Sp->MoistMin */
#define ROSadj    Envt[17].FVal  /* Adjustment for ROS to for more realistic
                                    flame characteristics */
#define NSites1   Envt[18].FVal  /* Number of sites/ha for establishment in
                                    most years */
#define NSites2   Envt[19].FVal  /* No. of sites/ha for estab. in 10% of
                                    years */
#define FSites    Envt[20].FVal  /* No. of sites/ha for estab. in fire years */
#define SupprYrs  Envt[21].FVal  /* No. of years of suppression before a
                                    tree suffers increased mortality */
#define SupprInc  Envt[22].FVal  /* DBH incr. needed to avoid suppression */
#define FireStim  Envt[23].FVal  /* Estab. seed pool stimulation by fire */

#define NEnvVars 24              /* Size of Envt array */

/* I/O units (defaults in brackets) */
#define In        IOUnits[0].CurPtr  /* Stream for instructions (stdin) */
#define Out       IOUnits[1].CurPtr  /* Most output goes here (stdout) */
#define Debug     IOUnits[2].CurPtr  /* Used for debugging (NULL) */
#define GrowOut   IOUnits[3].CurPtr  /* Can be used for writing specific
                                        growth rate information (NULL) */
#define SummOut   IOUnits[4].CurPtr  /* Site summaries written here (stdout) */
#define TimeOut   IOUnits[5].CurPtr  /* Time reports go here (stdout) */
#define FireOut   IOUnits[6].CurPtr  /* Fire reports go here (stdout) */

#define NIOUnits 7                   /* Size of array */


/*******************************************************************************

  brgraph.h

  Header file for NEWBRIND graphics using Turbo C

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601
    Australia
```

```
  Description:

    Declares the TreeShapes enumerated type and the graphics functions

*****************************************************************************/
/**
  Shapes of trees.  Each tree is drawn as a stick with a shape
  on top.  The shapes are diamonds or squares.
**/
enum TreeShapes
{
  Diamond,
  Square
};

/* Functions */
void    DrawPlot( void );
void    ClearGraphics( void );
int     gcprintf( int x, int y, int colour, char * fmt, ... );
void    GraphAllTrees( void );
void    GraphTree( int XPos, double Height, int Colour, int Shape );
int     MinXPos( void );
int     RangeXPos( void );
void    Picture( void );


/*****************************************************************************

  lipldef.h

  Definition of data structure used by InitLiPl()

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601

*****************************************************************************/
typedef struct {
  double LBound;
  double UBound;
  short int LinePos;
  short int ListPtr;
  char Name[7];
  char Symbol;
} LiPlVar;

#if ! UNIX

void InitLiPl( int, LiPlVar *, int );
void LinePlot( int, double *, char * );

#else

void InitLiPl( );
void LinePlot( );

#endif


/*****************************************************************************

  outoptns.h
```

```
  Definitons of option bits used by Output()

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601
    Australia

  Description:

    These flags are set in main() (in file newbrind.c) each simulated
    year and passed to Output() to control how much is output.

*****************************************************************************/
#define OutNoOp 0x0000  /* No output */
#define OutInit 0x0001  /* Initialise line-printer plot */
#define OutLiPl 0x0002  /* Output a line on the line-printer plot */
#define OutSiLi 0x0004  /* List the site this year */
#define OutRFir 0x0008  /* Report fires this year */
#define OutRTim 0x0010  /* Report the real time this simulated year */


/*****************************************************************************

  ranseeds.h

  Header file with indices into the random number seeds array.

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601
    Australia

  Description:

    These indices are passed to BrRand() to specify which sequence of
    pseudo-random numbers to use.  By keeping separate indexed sequences,
    it is possible to run parallel sets of simulations with some
    psuedo-random sequences identical and others different.

*****************************************************************************/
enum { RSClim1,
       RSClim2,
       RSWFire1,
       RSWFire2,
       RSPFire1,
       RSPFire2,
       RSPrWFire,
       RSPrPFire,
       RSPrFMort,
       RSPrNMort,
       RSPrSMort,
       RSBirth1,
       RSBirth2,
       RSBirth3,
       RSBirth4,
       RSStSite,
       RSPrFStDth,
       RSLigSprt,
```

```
       RSClim3,
       RSClim4
#if GRAPHICS
       , RSXPos
#endif
       };
```

C.2 Main program

```
/*******************************************************************************

  N E W B R I N D

  A forest gap simulator for eucalypt forests of the Brindabella Range,
  Australian Capital Territory

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601

  Description:

    This program simulates dynamics of eucalypt forests on a small
    forest plot (c. 1/50 ha) in the Brindabella Range.  It is a gap
    model of the type described by Botkin et al. (1972) and Shugart
    (1984).  It is derived from the BRIND model of Shugart and
    Noble (1981).

    NEWBRIND differs from BRIND in simulating dynamics on a smaller
    forest plot, and specifying site environment as altitude above
    sea level in metres, angle of slope in degrees and aspect (azimuth)
    of slope in degrees.

    This version of the program has been compiled on a number of
    computers, with differences in the code delimited by preprocessor
    symbols, set in the header file "globals.h".  The operating systems
    and compilers are:

      MS-DOS      Turbo C version 2.0, with or without screen graphics
      Sun 3       Sun C
      VAX/VMS     VAX C

  Program logic:

    Startup: Check for command-line arguments.  If present, they become
             standard input and output streams.  Otherwise use stdin and
             stdout.

    Main loop: This is executed once for each simulation run (of length
               specified by the user).  It starts with a call to SetVars(),
               which accepts user commands to set variables which control
               program execution and model behaviour.  When the user
               enters the "run" command, SetVars() returns with the value
               Run, and the program proceeds.  When "stop" is entered,
               files are closed and final reports made.  For each
               simulation, line plot initialisation is performed if
               required, and for graphics output, the plot outline (no
               trees) is drawn on the screen.

    Annual loop: Executed once for each simulated year.  In the MS-DOS
                 version the keyboard is polled for a press of the
                 Escape key, which provides a mechanism for the user
```

```
                to stop the current simulation.

                In the annual loop, the functions are called in the
                order:  Climate(), Grow(), Fire() (if this is a fire
                year, determined randomly), Death(), Birth(), LifeTable()
                and Output();

                SiteSummary() is called if the simulation has passed
                the lead time, and may cause the simulation to break
                out of the annual loop.

  Global variables:

    int    Burn                     Flag: non-zero in fire years
    int    Year                     Current simulation year
    int    RunYears                 The number of years to run in the
                                    current simulation

    Other "global" variables are preprocessor symbols defined in
    vstructs.h.  Their values can be changed by the user (see setvars.c).

  Required files:

    A species parameter file with parameter values in free-field (space-
    separated) format.  See files trees.h and spparams.c for details of
    parameters.  Read by SpParams() in spparams.c (called from SetVars()).

    Site listing file (optional).  Format is species short name (see
    trees.h) followed by number of trees and DBH values (cm) for each tree.
    Read by StrtSite() in strtsite.h (called from SetVars()).

    Sets of program instructions can also be placed in files and read.  See
    setvars.c for syntax of program instructions.

  Global variables, constants and required functions are described in
  the header files globals.h, vstructs.h, trees.h, ranseeds.h and
  outoptns.h

  Local variables are documented where they are declared

****************************************************************************/
/**
  The preprocessor symbol MAIN is set only in newbrind.c so the global
  variables declared in globals.h are not declared extern (i.e., they
  are declared as belonging to this program unit.
**/
#define MAIN

#include <stdio.h>
#include "globals.h"
#if TC
# include <conio.h>
#endif
#include <math.h>
#if ! UNIX
# include <stdlib.h>
#endif
#include "outoptns.h"
#include "ranseeds.h"
#include "trees.h"
#include "vstructs.h"

#if GRAPHICS
# include <graphics.h>
# include "brgraph.h"
#endif

#if VMS
```

```
# define EXIT_OK 1
#else
# define EXIT_OK 0
#endif

/**
  Here's the main program...
**/

main( argc, argv )
char **argv;
int argc;
{

  /**
    Local variables
  **/
  double MeanTemp;        /* Mean annual temperature */
  double GrowMonths;      /* Equiv. no. of months with tree growth */
  double MoistIndex;      /* Mean moisture index for the year */
  int    Frosts;          /* Number of frosts this year */
  enum {
    Plotting,
    NotPlotting
  } LiPlStatus = NotPlotting;  /* Status of line-printer output */
  int OldStrtYear = -1;        /* Previous starting year, used for line-
                                  printer plot initialisation */
  unsigned OutOpt = OutNoOp;   /* Options for output */

#if GRAPHICS
  register int MaxY;           /* Maximum Y value on the screen */
#endif

#if VMS
  /**
    Set I/O units:  VAX C won't allow their initialization in SetVars()
    See setvars.c for details of the IOUnits array
  **/
  In = IOUnits[0].DfltPtr = stdin;
  Out = IOUnits[1].DfltPtr = stdout;
  /* Debug & GrowOut stay NULL */
  SummOut = IOUnits[4].DfltPtr = stdout;
  TimeOut = IOUnits[5].DfltPtr = stdout;
  FireOut = IOUnits[6].DfltPtr = stdout;
#endif

  /* Initialise the global parameter file name */
  strcpy( ParamFileName, "*** No species parameters yet ***" );

  /* Nondescript introductory message */
  puts( "\nNew BRIND...  Here goes nothing!!\n" );

  /**
    Startup:
    If there are command line arguments, they are input and output files.
    On file open failure, use default values: if it is because file names
    are mis-spelt, then the program doesn't stop running.
  **/
  if( argc > 1 )
  {
    /* Filename for input stream */
    if( (In = fopen( *++argv, "r" )) == NULL )
    {
      errz( WARNING, "Open failure on input file:  Using default stdin" );
      In = stdin;
    }
    else
      printf( "Input stream being read from %s\n\n", *argv );
```

```
    if( argc > 2 )
    {
      /* Filename for output stream */
      if( (Out = fopen( *++argv, "w" )) == NULL )
      {
        errz( WARNING, "Open failure on output file:  Using default stdout" );
        Out = stdout;
      }
      else
        printf( "Output stream being sent to %s\n\n", *argv );
    }
  }  /* if( argc > 1 ) */

  /**
    Outer program loop, executed when SetVars() returns after
    the user gives a "run" command
  **/
  while( SetVars() != Stop )
  {
    /* Only run if we have some species */
    if( FirstSpecies() != NULL )
    /**
      Program running block...
    **/
    {
      /* Reset Last fire year if it's nonsensical */
      if( LastFrYr > StrtYear )
        LastFrYr = StrtYear;

      /**
        This fools the program into line plot initialising when a set of
        years has been run with no plotting
      **/
      if( PLinePlt && (LiPlStatus == NotPlotting) )
        OldStrtYear--;
      if( PLinePlt > 0 )
        LiPlStatus = Plotting;
      else
        LiPlStatus = NotPlotting;

      /* Report the time here (StrtYear) if PRepTime > 0 */
      if( PRepTime > 0 )
        Output( OutRTim );

      /* Initialize line plotting if StrtYear != OldStrtYear */
      if( PLinePlt > 0 && StrtYear != OldStrtYear )
      {
        Output( OutInit );
        Year = StrtYear;
        Output( OutLiPl );
      }

#if GRAPHICS
      if( Graphics )
      {
        DrawPlot();
        GraphAllTrees();
        MaxY = getmaxy();
        gcprintf( 0, MaxY, WHITE, "Year" );
        gcprintf( 0, MaxY - textheight( "H" ) - 2, YELLOW, "Biom" );
        gcprintf( 0, MaxY - 2 * (textheight( "H" ) + 2), YELLOW, "No." );
      }
#endif

      /************************

        A N N U A L   L O O P
```

```
    ====================
    Here is the beginning of the annual loop of the model.

   ************************/
   for( Year = StrtYear + 1; Year <= StrtYear + RunYears; Year++ )
   {
#if TC
    /* Bail-out check for pressing of Esc key */
    if( kbhit() && getch() == '\033' )
    {
#if GRAPHICS
     if( Graphics )
      gcprintf( 100, getmaxy() - textheight( "H" ), WHITE,
           "Bailing out!" );
     else
#endif
      fprintf( Out, "\nBailing out!\n\n" );
     break;
    }
#endif

#if GRAPHICS
    if( Graphics )
    {
     setfillstyle( EMPTY_FILL, BLACK );
     bar( 40, MaxY - 30, 90, MaxY );
     gcprintf( 40, MaxY, WHITE, "%-5d", Year );
     gcprintf( 40, MaxY - textheight( "H" ) - 2, YELLOW, "%.1lf",
          TotalPlotBiomass() );
     gcprintf( 40, MaxY - 2 * (textheight( "H" ) + 2), YELLOW, "%d",
          TotNTrees() );
    }
#endif

    Climate( &MeanTemp, &GrowMonths, &MoistIndex, &Frosts );

    Burn = NO;
    if( BrRand( RSPrWFire ) < FireProb
      || Year == FireYr1 || Year == FireYr2 )
     Fire();

    Death();

    if( BirthFlag )
     Birth( MeanTemp, MoistIndex, Frosts );

    /* Growth returns Stop if something is wrong */
    if( Grow( GrowMonths, MoistIndex ) == Stop )
     break;

    if( LifeTablFlag )
     LifeTable();

    OutOpt = OutNoOp;
    if( PLinePlt > 0 && ! ((Year - StrtYear) % PLinePlt) )
     OutOpt |= OutLiPl;
    if( PRepFire > 0 && ! ((Year - StrtYear) % PRepFire) )
     OutOpt |= OutRFir;
    if( PSiteLst > 0 && ! ((Year - StrtYear) % PSiteLst) )
     OutOpt |= OutSiLi;
    if( PRepTime > 0 && ! ((Year - StrtYear) % PRepTime) )
     OutOpt |= OutRTim;
    if( OutOpt )
     Output( OutOpt );

    /**
     Call SiteSummary after lead time.  It will return Stop if the
     annual loop is to be broken
```

```
        **/
        if( YSiteSum >= 0 && Year > LeadTime )
          if( SiteSummary() == Stop )
            break;

      }  /* End of annual loop */

      /* Reset Year and StrtYear, set OldStrtYear */
      StrtYear = --Year;
      OldStrtYear = StrtYear;

      /* Increment the replicates counter */
      Replicat++;

#if GRAPHICS
      if( Graphics )
      {
        gcprintf( 100, getmaxy(), WHITE, "Press any key to continue..." );
        getch();
        ClearGraphics();
      }
#endif

    }  /* End of program running block */
    else
      errz( WARNING, "No species!!" );

  }  /* End of outer program loop */

  /* Tidy up: delete trees */
  KillAll();

  /* Then close any open files (this should flush them) */
  if( In != stdin )
    fclose( In );
  if( Out != stdout )
    fclose( Out );
  if( Debug != NULL && Debug != stdout )
    fclose( Debug );
  if( GrowOut != NULL && GrowOut != stdout )
    fclose( GrowOut );
  if( SummOut != stdout )
    fclose( SummOut );
  if( TimeOut != stdout )
    fclose( TimeOut );
  if( FireOut != stdout )
    fclose( FireOut );

  /* Report any unreported fires */
  ReportFires();

  return( EXIT_OK );
}
```

C.3 Biological simulation functions

```
/*****************************************************************************

  grow.c

  Function Grow() which calculates annual increments for all trees on
  the simulated plot

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
```

```
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601
    Australia

  Description:

    Tree growth is calculated from DBH and height.  An optimal increment
    is first calculated, which is then reduced by indices of soil
    moisture stress, low temperature and competition from other trees.
    Trees showing slow growth are marked as suppressed.

  Global variables, constants and required functions are documented
  in the  header files globals.h, vstructs.h and trees.h

  Local variables are explained where they are declared

*****************************************************************************/
#include <stdio.h>
#include "globals.h"
#include <math.h>
#if ! UNIX
# include <stdlib.h>
#endif
#include "trees.h"
#include "vstructs.h"

#define MAXPROF    700  /* Size of competitor profile array */

int Grow( GrowMonths, MoistIndex )
double GrowMonths, MoistIndex;
{
  double   *CompnProfile,     /* Array for basal areas in competitor size
                                 profile */
           TempResponse,      /* Index of growing temperature */
           MoistResponse,     /* Index of soil moisture stress */
           DBMaxByHtMax,      /* Dmax * Hmax for a species */
           Competition,       /* Competition index */
           DBHInc,            /* DBH increment (cm) */
           B2by3,             /* 3 * b2 for a species */
           B3by4,             /* 4 * b3 for a species */
           CompnCoeff;        /* Competiton coefficent (plot size dependent) */
  double   PlotBiomass = 0.0;  /* Total plot biomass accumulator */
  double   BiomassIndex;       /* Biomass index if wanted */
  double   GrowthReduction;    /* Combined reduction from competition & climate */
  register int i;              /* Index into CompnProfile */
  register Species *Sp;        /* Species pointer */
  register Tree *Tr;           /* Tree pointer */
  Tree *BiggestTree;           /* Pointer to biggest tree */

  /**
    Temporary array for storage of competitor profile
  **/
  if( (CompnProfile = (double *) calloc( MAXPROF, sizeof( double ) )) == NULL )
    errz( FATAL, "No room for allocation of competitor profile array" );

  /**
    Fill up competitor profile array -- accumulate tree basal area
    indexed by DBH in CClWdth increments.
    Also accumulate plot biomass, if we're using the biomass index.
    And find the biggest tree if its growth is to be reported (see below).
  **/
  BiggestTree = NULL;
  for( Sp = FirstSpecies(); Sp != NULL; Sp = NextSpecies() )
    for( Tr = FirstTree(); Tr != NULL; Tr = NextTree() )
      if( (i = Tr->DBH / CClWdth) >= MAXPROF )
      {
```

```
      fprintf( stderr, "Year %d, %s:  %.1lf\n", Year, Sp->ShortName,
          Tr->DBH );
      errz( WARNING, "Tree too big for competition array: reduce CClWdth" );
      free( (char *) CompnProfile );
      return( Stop );
    }
    else
    {
      CompnProfile[i] += Tr->Basalarea;
      if( MaxBiom > 0.0 )
        PlotBiomass += Biomass();

      /* Keep track of which is the biggest tree */
      if( GrowOut )
        if( BiggestTree == NULL || Tr->DBH > BiggestTree->DBH )
          BiggestTree = Tr;
    }

/**
  Accumulate profile information so each cell holds the cumulative
  basal area of all trees in it and bigger cells.
**/
for( i = MAXPROF - 2; i >= 0; --i )
  CompnProfile[i] += CompnProfile[i+1];

/**
  Index of growing months this year, and coefficient used to calculate
  competition index.  The user can stop climatic effects on growth with
  the ClimGrow flag.
**/
if( ClimGrowFlag )
  TempResponse = GrowMonths / 12.0;
else
  TempResponse = 1.0;

CompnCoeff = CompnIntensity / PlotArea;

/**
  Biomass index is only used if it is wanted (if the user
  has set MaxBiom to a positive value).
**/
if( MaxBiom > 0.0 )
  BiomassIndex = 1.0 - PlotBiomass / MaxBiom;
else
  BiomassIndex = 1.0;

/**
  Species loop
**/
for( Sp = FirstSpecies(); Sp != NULL; Sp = NextSpecies() )
{
  /**
    Reset suppressed tree counter & calculate environmental response
    index, max. DBH by max. height, B2 * 3.0, and B3 * 4.0.  All these
    things should help speed things up.
  **/
  Sp->NSuprs = 0;
  /**
    The following complicated equation sets the species moisture
    response according to MI this year (MoistIndex), the species
    MImin value (Sp->MoistMin) and the value of MoistResponse at
    MI=MImin for the species (MinMMI).
    The user can stop climatic effects on growth with the ClimGrow
    flag.
  **/
  if( ClimGrowFlag )
    MoistResponse = ((1.0 - MinMMI) * MoistIndex + MinMMI - Sp->MoistMin) /
                    (1.0 - Sp->MoistMin);
```

```
else
  MoistResponse = 1.0;

DBMaxByHtMax = Sp->DBHMax * Sp->HeightMax;
B2by3 = 3.0 * Sp->B2;
B3by4 = 4.0 * Sp->B3;

/**
  Tree loop
**/
for( Tr = FirstTree(); Tr != NULL; Tr = NextTree() )
{
  /**
    Calculate competition index.  It is given by
      1 - (CompIntensity / PlotArea) * (summed BA of competitors)
    for plot area in sq m.  Note that this trees's basal area
    is not counted as competing with it.
  **/
  i = Tr->DBH / CClWdth;
  Competition = 1.0 - CompnCoeff * (CompnProfile[i] - Tr->Basalarea);

  /**
    Calculate optimal DBH increment from DBH and height, allowing
    for costs of maintenance respiration.  Division by vigour term
    is so that DH/DmHm still makes sense when D is large but H is
    small because of low vigour.
  **/
  DBHInc = Tr->G * Tr->DBH *
           (1.0 - Tr->DBH * Tr->Height / Tr->Vigour / DBMaxByHtMax) /
           ( (B2by3 - B3by4 * Tr->DBH) * Tr->DBH + 274.0 );

  /**
    The user can report details of increments (also below).
    This has now been modified to report on only the biggest
    tree.
  **/
  if( GrowOut && Tr == BiggestTree )
    fprintf( GrowOut,
             "%s\t%.2lf\t%.0lf\t%.3lf\t%.0lf\t%.2lf\t%.3lf\t%.3lf\t%.3lf\t",
             Sp->ShortName, Tr->DBH, Tr->Height, Tr->Vigour, Tr->G,
             DBHInc, Competition, TempResponse, MoistResponse );

  /**
    The optimal increment is reduced by indices of competition from
    bigger trees, cold temperatures, soil moisture stress and
    stand biomass.
  **/
  GrowthReduction = Competition * TempResponse * MoistResponse *
      BiomassIndex;
  DBHInc *= GrowthReduction;

  if( GrowOut && Tr == BiggestTree )
    fprintf( GrowOut, "%.4lf\n", GrowthReduction );

  /**
    If the tree is suppressed, add one to the number of years
    it has been suppressed (see death.c for use of this), and
    the counter of suppressed trees for the species is
    incremented.  If the increment is calculated as negative
    from REALLY strong competition, set it to zero.

    NewSupprFlag determines whether or not to use the new
    suppression method (default not): with this method a tree
    is suppressed if the indices reducing its growth multiply to
    less than SupprInc (note the different meaning of SupprInc in
    this case).
  **/
  if( (NewSupprFlag && GrowthReduction < SupprInc) ||
```

```
          (! NewSupprFlag && DBHInc < SupprInc) )
      {
        if( DBHInc < 0.0 )
          DBHInc = 0.0;
        Tr->SuppressYrs++;
        Sp->NSuprs++;
      }
      else
        Tr->SuppressYrs = 0;

      /* Now add the increment to the tree */
      AddDBH( DBHInc );
    }
  }

  free( (char *) CompnProfile );

  return( Run );

}


/****************************************************************************

  fire.c

  Function Fire() which calculates response of individual trees to fire

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601
    Australia

  Description:

    Fire characteristics are calculated from fire danger index 'on the
    day of the fire'.  If the fire is too slow to spread then it does no
    harm to any tree on the plot.  Tree responses are calculated
    according to whether the species is fire-resistant or
    fires-sensitive.  Trees killed by fire are marked as dead by setting
    their DBH to a negative value.

  Global variables, constants and required functions are documented
  in the header files globals.h, vstructs.h and trees.h

  Local variables are explained where they are declared

****************************************************************************/
#include <stdio.h>
#include "globals.h"
#include <math.h>
#include "ranseeds.h"
#include "trees.h"
#include "vstructs.h"

#if GRAPHICS
# include <graphics.h>
# include "brgraph.h"
#endif

void Fire()
{
  double  FireDangerIndex,  /* FDI 'on the day of the fire' */
          Fuel,             /* Fine fuel load in t/ha */
```

```
        RateOfSpread,     /* Flame front ROS in m/hr */
        FlameHeight,      /* in metres */
        ScorchHeight,     /* in metres */
        LP,               /* Linear predictor for GLMs */
        PrMort,           /* Probability of death */
        PrStemDeath,      /* Probability of death of above-ground parts */
        BestHeight,       /* How high a tree could be w/o fire damage */
        PropCanScorch;    /* Proportion of canopy scorch */
int     isCrownFire = NO, /* Is flame high enough to ignite the canopy? */
        YrsSinceFire;     /* Years since last fire */
int     Killed = 0;       /* No. killed by this fire */
Species *Sp;              /* Pointer to species in linked list */
Tree    *Tr;              /* Pointer to trees in linked lists */

/**
  Calculate FDI 'on the day of the fire' from user-supplied mean and SD
  of FDI.  It can't be negative.
**/
FireDangerIndex = NormDev( FDIMean, FDIStDv, RSWFire1, RSWFire2 );
if( FireDangerIndex < 0.0 )
  FireDangerIndex = 0.0;

/**
  Fine fuel load is calculated using years since last fire, max.
  fuel load and fuel decomposition constant.  The last two vars
  can be set by the user.
**/
YrsSinceFire = Year - LastFrYr;
Fuel = MaxFuel * (1.0 - exp( - FuelDeco * YrsSinceFire ));

/**
  Flame characteristics are calculated using the Noble et al. (1980)
  equations from the McArthur meters.

  Calculated ROSs and flame heights have been shown to be very
  unreliable and usually far too high.  In the absence of better flame
  models (which are strictly outside the scope of this gap model) we
  adjust the calc'd ROS by a fudge factor (chocolate) derived from
  comparisons between this rubbish and decent models (and real fires
  too).  If the user doesn't like the fudge factor ROSadj he/she can
  change it.
**/
RateOfSpread = 1.2 * Fuel * FireDangerIndex * exp( Slope * 0.069 );
RateOfSpread = RateOfSpread / ROSadj;

/* If the fire is going nowhere then trees are not affected */
if( RateOfSpread < 1.0 )
  return;
else
  Burn = YES;

/**
  Calculate flame height in metres.  If it's more than 16 m it is
  considered a crown fire.
**/
FlameHeight = 0.013 * RateOfSpread + 0.24 * Fuel - 2.0;
if( FlameHeight > 16.0 )
  isCrownFire = YES;

/**
  Leaf scorch height is 6 times flame height (Luke and McArthur 1978)
**/
ScorchHeight = 6.0 * FlameHeight;

/* Species loop */
for( Sp = FirstSpecies(); Sp != NULL; Sp = NextSpecies() )
{
  for( Tr = FirstTree(); Tr != NULL; Tr = NextTree() )
```

```
/**
  Fire-resistant species: use the 3-stage model from the
  Gudgenby data.
**/
if( Sp->FireResistant )
{
  /**
    Stage one: tree death.  Calculate probability from vigour and
    DBH (binomial GLM).  Mark the dead ones.
  **/
  LP = -2.518 - 1.715 * Tr->Vigour + (0.1185 - 0.1603 * Tr->Vigour) *
       Tr->DBH ;
  PrMort = AntiLogit( LP );
  if( BrRand( RSPrFMort ) < PrMort )
  {
    Tr->DBH = - Tr->DBH;
    Killed++;
  }
  else
  {
    /**
      Stage two: stem death.  Another binomial GLM.  If the stem is
      dead, replace it with a vigorous sprout (DBH 2-3cm).
    **/
    LP = Sp->Fire2Const + Sp->Fire2DBH * Tr->DBH;
    PrStemDeath = AntiLogit( LP );
    if( BrRand( RSPrFStDth ) < PrStemDeath )
    {
      Tr->DBH = 0.0;
      AddDBH( 2.0 + BrRand( RSLigSprt ) * 1.0 );
    }
    else
    {
      /**
        Calculate recovery as proportion of height left after crown
        death and epicormic growth (another binomial GLM).
      **/
      LP = 0.513 + 0.0477 * Tr->DBH;
      Tr->Height = Tr->Height * AntiLogit( LP );
      /**
        BestHeight is the height of a tree with vigour of 1.0 (i.e.,
        undamaged by fire).  The actual height of a tree is therefore
        BestHeight * vigour.
      **/
      BestHeight = 137.0 + Tr->DBH * (Sp->B2 - Tr->DBH * Sp->B3);
      Tr->Vigour = Tr->Height / BestHeight;

    }  /* Stem not killed block */
  }  /* Tree not killed block */
}  /* Fire-resistant species block */
else
{
  /**
    Fire-sensitive species: calculate mortality from prop'n canopy
    scorch (scorch is converted to centimetres first).  Probability
    of mortality is calculated so the tree is killed by full scorch
    but survives 50% scorch.
  **/
  PropCanScorch = (ScorchHeight * 100.0 - Sp->GrowthForm * Tr->Height) /
                  Tr->Height / (1.0 - Sp->GrowthForm);
  PrMort = 2.0 - 4.0 * exp( -1.39 * PropCanScorch );
  if( BrRand( RSPrFMort ) < PrMort )
  {
    Tr->DBH = - Tr->DBH;
    Killed++;
  }
}  /* Non-sprouting species block */
```

```
  }  /* Species loop */

  /* Save fire details to memory with SaveFire() */
  SaveFire( isCrownFire ? 1000.0 : FlameHeight, RateOfSpread,
            Killed, TotNTrees() );

  /* Reset last fire year */
  LastFrYr = Year;

#if GRAPHICS
  /* Summary to screen if in graphics mode */
  if( Graphics )
  {
    int MaxY = getmaxy();
    int MaxX = getmaxx();
    int XStart = 0.6 * MaxX;
    int THeight = textheight( "H" );

    /* Blank that section of the screen first */
    setfillstyle( EMPTY_FILL, BLACK );
    bar( XStart, MaxY - 3 * THeight, MaxX, MaxY );
    gcprintf( XStart, MaxY - 2 * THeight, LIGHTRED, "FIRE in year %-5d", Year );
    if( isCrownFire )
      gcprintf( XStart, MaxY - THeight, RED, "Crown fire               " );
    else
      gcprintf( XStart, MaxY - THeight, RED, "Flame height:  %.1lf",
                FlameHeight );
    gcprintf( XStart, MaxY, RED, "Trees killed : %d / %d     ",
              Killed, TotNTrees() );
  }
#endif
}

/*****************************************************************************

  death.c

  Function Death() which removes trees killed by suppression, fire
  and size-independent mortality.

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601

  Description:

    Trees marked as dead by Fire() (-ve DBH) are removed. Then trees
    are killed randomly according to their annual probability of size-
    independent mortality.  Life table information is then tallied.
    The flags AIMortFlag and SuMortFlag can be reset by the user to
    prevent size-independent and suppression mortality, respectively.

*****************************************************************************/
#include <stdio.h>
#include "globals.h"
#include <math.h>
#include "ranseeds.h"
#include "trees.h"
#include "vstructs.h"

void Death()
{
  register int dc;        /* DBH class (5cm wide) for life table */
```

```
register int TreeFate;   /* Fate of each tree this year */
register Species *Sp;    /* Species pointer */
register Tree *Tr;       /* Tree pointer */

/**
  Loop through all the trees in all the species.  The functions
  return pointers to species and trees in the lists of them held
  in memory.  See trees.c for details.
**/
for( Sp = FirstSpecies(); Sp != NULL; Sp = NextSpecies() )
{
  Tr = FirstTree();
  while( Tr != NULL )
  {
   /**
     Fire() sets a tree's DBH negative if it has been killed by
     fire.  Reset the DBH to a positive number so it can
     be used to generate an index in the species' LifeTable array.
   **/
    if( Tr->DBH < 0.0 )
    {
      Tr->DBH = - Tr->DBH;
      TreeFate = firekill;  /* firekill is defined in globals.h */
    }
    /**
      If SIMortFlag is true (its default value, alterable by
      the user) then randomly check whether this tree will die
      according to its species' annual probability of mortality.
    **/
    else if( SIMortFlag &&
             (BrRand( RSPrNMort ) <= Sp->SizeIndMort) )
      TreeFate = sizeindmort;

    /**
      If SuMortFlag is true (default) and the tree is suppressed
      it is subject to greater mortality, changeable by the user.
    **/
    else if( SuMortFlag &&
        Tr->SuppressYrs >= SupprYrs &&
        BrRand( RSPrSMort ) <= SuprMort )
      TreeFate = suppresskill;

    /**
      Otherwise the tree is marked as staying put.
    **/
    else
      TreeFate = stayedput;

    /**
      For dead trees, update their species' LifeTable and then
      remove the tree from the program's list.  Note that
      KillTree() returns the pointer to the next tree in the
      list (like NextTree()).
    **/
    if( TreeFate == firekill || TreeFate == sizeindmort ||
        TreeFate == suppresskill )
    {
      if( Year > LeadTime )
      {
        dc = Tr->DBH / 5.0;  /* Tree is dead, so tally against DBH */
        if( dc > MAXDCLASS - 1 )  /* Only 41 DBH classes */
          dc = MAXDCLASS - 1;
        /**
          One more tree of this species has endured one of these
          (dying) fates.  Other (living) fates are recorded in
          LifeTable().
        **/
        Sp->LifeTable[dc][TreeFate]++;
```

```
          /**
            Accumulate G value data of dying trees with DBH > 5cm
          **/
          if( Tr->DBH > 5.0 )
          {
            Sp->SumG += Tr->G;
            Sp->nG++;
          }
        }

#if 0
        if( Debug && Tr->DBH > 5.0 )
        {
          fprintf( Debug, "%s\t%.2lf\t%.1lf\n", Sp->ShortName, Tr->DBH, Tr->G );
          fflush( Debug );
        }
#endif

        Tr = KillTree();
      }
      else
        Tr = NextTree();

    } /* Tree loop */
  } /* Species loop */

} /* Death() */


/****************************************************************************

  birth.c

  Function Birth() which establishes new trees on the NEWBRIND plot.

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601

  Description:

    This function adds new trees to the plot according to species
    biomass contribution and environmental conditions.

  Global variables, constants and required functions are documented
  in the header files globals.h, vstructs.h and trees.h

  Local variables are explained where they are declared

****************************************************************************/
#include <math.h>
#include <stdio.h>
#include "globals.h"
#include "ranseeds.h"
#include "trees.h"
#include "vstructs.h"

/**
  These symbols are only used in this function and are defined here
  so they can be easily changed.
**/
#define FireTrig    0.001  /* Seed pool diminution in the absence of fire */
#define BirthSize   1.2    /* DBH of new trees of non-lignotuberous species */
```

```
#define LSBirthSize 1.7    /* DBH of new trees of lignotuberous species */

void Birth( Temp, Moist, Frosts )
double Temp;      /* Mean temperature this year (degr C) */
double Moist;   /* Mean moisture index this year */
int    Frosts;  /* Number of frosts this year */
{
  /* Local variables */
  double            TotBiom = 0.0;  /* Biomass accumulator */
  double            LP;             /* Linear predictor used in antilogit */
  double            TotSeeds = 0.0; /* "Seed pool" accumulator */
  double            SeedP;          /* Species seed pool temp variable */
  register int      Sites;          /* Number of sites for establishment */
  register Species  *Sp;            /* Species pointer */
  register Tree     *Tr;            /* Tree pointer */

  /**
    First, accumulate biomasses of species and total on plot
    (function Biomass() returns biomass of current tree, scaled
    by plot size).  Also set the species GotMatureTrees flag
  **/
  for( Sp = FirstSpecies(); Sp != NULL; Sp = NextSpecies() )
  {
    Sp->Biomass = 0.0;
    Sp->GotMatureTrees = 0;
    for( Tr = FirstTree(); Tr != NULL; Tr = NextTree() )
    {
      Sp->Biomass += pow(Biomass(), SeedProd);  /* Try this modification:
                                                   SeedProd is defined in
                                                   SetVars() etc.

      Sp->Biomass += Biomass(); */
      if( Tr->DBH > Sp->SeedDBH )
        Sp->GotMatureTrees = 1;
    }
    TotBiom += Sp->Biomass;

#if 0
    /* Print out each species' initial contrib. to the seed pool */
    if (Debug)
    {
      fprintf(Debug, "%s\t%.4lf\n", Sp->ShortName, Sp->Seeds);
      fflush(Debug);
    }
#endif

  }

  /**
    Main species loop:
    The "seed pool" contributions of species are calcuated
  **/
  for( Sp = FirstSpecies(); Sp != NULL; Sp = NextSpecies() )
  {
    /**
      Calculate seed input: if a species has mature trees (set
      above) then its Seeds value is its proportional contribution
      to biomass; if not, its Seeds value is reduced by the decay
      factor SeedSurv.  Then the adjustment for seed dispersal into
      the plot is made.
    **/
    if( Sp->GotMatureTrees )
      Sp->Seeds = Sp->Biomass / TotBiom;
    else
      Sp->Seeds *= Sp->SeedSurv;

    /* Optionally add seed dispersal factor */
    if( SeedDispFlag )
      Sp->Seeds += Sp->SeedDisp;
```

```
    /**
      A species will only establish if there are not too many
      frosts for it this year
    **/
    if( Frosts < Sp->MaxFrosts )
    {

      /**
        Species seed pool contrib. is affected by fire for certain
        species.
      **/
      SeedP = Sp->Seeds;
      if( Sp->GermFireEnhanced && Burn )
        SeedP *= FireStim;
      if( Sp->GermNeedFire && ! Burn )
        SeedP *= FireTrig;

      /**
        Generalised Linear Model (regression) of environment for this
        year and this species...
      **/
      LP = Sp->BirthConst +
           ((Sp->BirthTemp[2] * Temp +
             Sp->BirthTemp[1]) * Temp +
             Sp->BirthTemp[0]) * Temp +
            (Sp->BirthMoist[1] * Moist +
             Sp->BirthMoist[0]) * Moist +
           Sp->BirthTempMoist * Temp * Moist +
           ((Geology[0] == 'S') ? Sp->BirthSediments : Sp->BirthGranites);
      SeedP *= AntiLogit( LP );

    }
    else
      /* Too many frosts this year for this species */
      SeedP = 0.0;

#if 0
    /* Print out each species final contrib. to the seed pool */
    if (Debug)
    {
      fprintf(Debug, "%s\t%.4lf\n", Sp->ShortName, SeedP);
      fflush(Debug);
    }
#endif

    /**
      Accumulate running total of seed pool, and each species' cumulative
      position in the pool.
    **/
    TotSeeds += SeedP;
    Sp->CumSeeds = TotSeeds;
  }

  /**
    Sites:
    First set an upper limit to the number of sites for establishment.
    In fire years, there are more sites available for establishment
    than non-fire years.  Otherwise, there are more available in 10%
    of years, randomly chosen.  Note that FSites, NSites1 and NSites2
    are expressed on a per-hectare basis, so must be scaled by plot
    area.
  **/
  if( Burn )
    Sites = FSites * PlotArea / 10000.0;
  else if( BrRand( RSBirth2 ) < 0.1 )
    Sites = NSites2 * PlotArea / 10000.0;
  else
```

```
    Sites = NSites1 * PlotArea / 10000.0;

  /**
    The actual number of sites is randomly chosen from 1 to the
    upper limit calculated above.
  **/
  Sites = (double) Sites * BrRand( RSBirth2 ) + 0.5;

  /**
    Plant trees:
    For each site, choose a species according to its contribution
    to the total seed pool.
  **/
  while( Sites-- )
  {
    /* Calcuate a random value between 0 and TotSeeds */
    SeedP = BrRand( RSBirth3 ) * TotSeeds;
    /**
      Find the first species whose cumulative seed pool value
      exceeds SeedP.
    **/
    for( Sp = FirstSpecies(); Sp != NULL && SeedP > Sp->CumSeeds;
         Sp = NextSpecies() )
      ;
    /**
      Add a tree of this species with "fuzzed" DBH: larger for
      lignotuberous species.
    **/
    if( Sp != NULL )
    {
      CreateTree();
      AddDBH( (Sp->HasLigSdlgs ? LSBirthSize : BirthSize) +
              0.5 * BrRand( RSBirth4 ) );
    }
  }

} /* Birth() */
```

C.4 Service functions

```
/*****************************************************************************

  trees.c

  Tree and species handling functions

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601
    Australia

  Description:

    These functions are for creating and removing species and trees and
    for fetching them in the list.  Also functions for incrementing DBH
    and fetching tree information.

    Trees and species are represented in memory by data structures defined
    in trees.h.  The structures are kept in dynamically-allocated
    singly-linked lists: one list of species and a list of trees for each
    species.  List handling functions are:
```

```
     Purpose           Species            Tree
     --------------------------------------------------
     Creation          CreateSpecies()    CreateTree()
     Deletion          KillSpecies()      KillTree()
     Beginning of list FirstSpecies()     FirstTree()
     Next in list      NextSpecies()      NextTree()
     --------------------------------------------------

   Each returns a pointer to its structure: Createxxx() to the new one
   created; Firstxxx() to the first in the list (or NULL if the list is
   empty); and both Nextxxx() and Killxxx() return the next in the list
   or NULL if there are no more in the list.

   A typical list traversal for all trees is:

     Species * Sp;
     Tree    * Tr;
       ...

     for( Sp = FirstSpecies(); Sp != NULL; Sp = NextSpecies() )
       for( Tr = FirstTree(); Tr != NULL; Tr = NextTree() )

   During list traversal the static variables CurrentSp and CurrentTr
   keep track of which species an tree are being dealt with.

   Other functions are:

   KillAll()    Remove all trees and species in one fell chop
   TotNTrees()  Return the total number of trees on the plot
   NSpecies()   Return the number of species currently defined
   TotalPlotBiomass() With a name like this, who needs comments?
   AddDBH()     Add the supplied increment to the current tree and update
                that tree's fields (and graphics position if relevant)
   LeafArea()   LAI contribution of this tree, scaled by plot area
   Biomass()    Biomass contrib. of this tree, scaled by plot area

  Global variables, constants and required functions are documented
  in the header files globals.h, vstructs.h and trees.h

  There are some static variables declared below which are private to all
  the functions in this file.

  Local variables are explained where they are declared

****************************************************************************/
#include <stdio.h>
#include "globals.h"
#include <math.h>
#if ! UNIX
# include <stdlib.h>
#endif
#include "ranseeds.h"
#include "trees.h"
#include "vstructs.h"

#if GRAPHICS
# include "brgraph.h"
  extern enum { Text, Graph } CurrentState;
#endif

static int      NSpec = 0;           /* Number of species           */
static int      TotNTr = 0;          /* Total number of trees       */
static Species *Species1 = NULL;     /* Pointer to first species    */
static Species *CurrentSp = NULL;    /* Current species is known    */
static Species *PrevSp = NULL;       /* Previous species referenced */
static Species *LastSp = NULL;       /* Last species of all         */
static Tree    *CurrentTr = NULL;    /* Current tree is known       */
static Tree    *PrevTr = NULL;       /* Last tree referenced        */
```

```
static double  TotalPlotBiom = 0.0;   /* Guess what this is!         */

/*****************************************************************************/
Species *CreateSpecies()       /* Make a new species            */
{                              /* New species becomes the last one  */
                               /* Return pointer to new species     */

  /* Running out of memory will halt the program */
  if( (CurrentSp = (Species *) malloc( sizeof( Species ) )) == NULL )
    errz( FATAL, "No storage left for new species" );

  /* Put it at the end of the list */
  if( LastSp != NULL )
    LastSp->Next = CurrentSp;
  LastSp = CurrentSp;
  CurrentSp->Next = NULL;
  CurrentSp->Tree1 = NULL;
  if( Species1 == NULL )
    Species1 = CurrentSp;
  NSpec++;

  /* Initialize fields that won't be filled by SpParams() */
  CurrentSp->NTrees = 0;
  CurrentSp->NSuprs = 0;
  CurrentSp->GotMatureTrees = 0;
  CurrentSp->Seeds = 0.0;

  /* G distribution stuff */
  CurrentSp->SumG = 0.0;
  CurrentSp->nG = 0;

  return( CurrentSp );
}

Species *KillSpecies()         /* "Kill" the current species     */
{                              /* Return pointer to next species */
  register Species *DoomedSp;

  if( PrevSp == NULL )         /* For first species              */
    Species1 = CurrentSp->Next;
  else                         /* For others                     */
    PrevSp->Next = CurrentSp->Next;

  DoomedSp = CurrentSp;
  CurrentSp = DoomedSp->Next;
  TotNTr -= DoomedSp->NTrees;
  free( (char *) DoomedSp );
  NSpec--;

  return( CurrentSp );
}

Species *FirstSpecies()        /* Get first species */
{
  PrevSp = NULL;
  CurrentSp = Species1;
  return( CurrentSp );
}

Species *NextSpecies()         /* Get next species */
{
  PrevSp = CurrentSp;
  CurrentSp = CurrentSp->Next;
  return( CurrentSp );
}
/*****************************************************************************/
Tree *CreateTree()             /* New tree of current species is "born"     */
{                              /* It becomes the first on the species' list */
```

```
                                /* Pointer to the new tree is returned       */

  /* Running out of memory will halt the program */
  if( (CurrentTr = (Tree *) malloc( sizeof( Tree ) )) == NULL )
    errz( FATAL, "No storage left for new tree" );

  /* Insert at the head of the list */
  CurrentTr->Next = CurrentSp->Tree1;
  CurrentSp->Tree1 = CurrentTr;
  CurrentSp->NTrees++;
  TotNTr++;

  /* Initialise fields */
  CurrentTr->DBH = 0.0;
  CurrentTr->OldDBH = 0.0;
  CurrentTr->Basalarea = 0.0;
  CurrentTr->Biomass = 0.0;
  CurrentTr->LeafArea = 0.0;
  CurrentTr->Height = 0.0;
  CurrentTr->Age = -1;
  CurrentTr->SuppressYrs = 0;
  CurrentTr->Vigour = 1.0;
  /**
    G for this tree taken from the species parameters of mean and SD of
    distribution, but can't be negative.
  **/
  do {
    CurrentTr->G = NormDev( CurrentSp->GMean, CurrentSp->GSD, RSBirth1,
                            RSBirth2 );
  } while( CurrentTr->G < 0.0 );

#if GRAPHICS
  CurrentTr->XPos = MinXPos() + BrRand( RSXPos ) * RangeXPos();
  CurrentTr->GraphHeight = 0.0;
#endif

  return( CurrentTr );
}

Tree *KillTree()                /* "Kill" current tree                  */
{                               /* Return pointer to next tree in list */
  register Tree *DoomedTr;

  TotalPlotBiom -= CurrentTr->Biomass;

  if( PrevTr == NULL )          /* For first tree of the species        */
    CurrentSp->Tree1 = CurrentTr->Next;
  else                          /* For other trees                      */
    PrevTr->Next = CurrentTr->Next;

#if GRAPHICS
  /**
    Graphics stuff:  XOR mode replotting = removal from screen, but
    check state first.
  **/
  if( CurrentState == Graph && CurrentTr->GraphHeight > 0.0 )
    GraphTree( CurrentTr->XPos, CurrentTr->GraphHeight,
               CurrentSp->GraphColour, CurrentSp->GraphShape );
#endif

  DoomedTr = CurrentTr;
  CurrentTr = DoomedTr->Next;
  free( (char *) DoomedTr );
  CurrentSp->NTrees--;
  TotNTr--;

  return( CurrentTr );
}
```

```
Tree *FirstTree()              /* First tree of current species */
{
  PrevTr = NULL;
  CurrentTr = CurrentSp->Tree1;
  return( CurrentTr );
}

Tree *NextTree()               /* Next tree of current species */
{
  PrevTr = CurrentTr;
  CurrentTr = CurrentTr->Next;
  return( CurrentTr );
}
/*******************************************************************************/
void KillAll()
{
  Species *Sp;
  Tree *Tr;

  for( Sp = FirstSpecies(); Sp != NULL; Sp = KillSpecies() )
    for( Tr = FirstTree(); Tr != NULL; Tr = KillTree() )
      ;

  TotalPlotBiom = 0.0;
}

int TotNTrees()
{
  return( TotNTr );
}

int NSpecies()
{
  return( NSpec );
}
double TotalPlotBiomass()
{
  return( TotalPlotBiom );
}
/*******************************************************************************/
void AddDBH( DBHInc )  /* Add (annual) DBH increment, update age, height, */
double DBHInc;         /* basalarea, biomass, leaf area                   */
{
  TotalPlotBiom -= CurrentTr->Biomass;
  CurrentTr->OldDBH = CurrentTr->DBH;
  CurrentTr->DBH += DBHInc;
  /* 7.854e-5 is pi / 40000 to give BA in sq. m */
  CurrentTr->Basalarea = CurrentTr->DBH * CurrentTr->DBH * 7.854e-5;

  /**
    Height is determined from a quadratic equation and adjusted by
    vigour
  **/
  CurrentTr->Height = CurrentTr->Vigour * (137.0 + CurrentTr->DBH *
      (CurrentSp->B2 - CurrentTr->DBH * CurrentSp->B3));

  /* Adjustment of coefficients for plot area in sq m */
  CurrentTr->Biomass = 0.7123 / PlotArea * pow( CurrentTr->DBH, 2.58 );
  TotalPlotBiom += CurrentTr->Biomass;
  CurrentTr->LeafArea = 5.392e-3 / PlotArea * pow( CurrentTr->DBH, 2.65 );
  CurrentTr->Age++;

#if GRAPHICS
  /* Graphics plotting stuff */
  if( Graphics && CurrentState == Graph )
  {
    int diff;
```

```
    if( (diff = CurrentTr->Height - CurrentTr->GraphHeight) >= 100.0 ||
        diff <= -100.0 )
    {
      if( CurrentTr->GraphHeight > 0.0 )
        GraphTree( CurrentTr->XPos, CurrentTr->GraphHeight,
                   CurrentSp->GraphColour, CurrentSp->GraphShape );
      CurrentTr->GraphHeight = CurrentTr->Height;
      GraphTree( CurrentTr->XPos, CurrentTr->GraphHeight,
                 CurrentSp->GraphColour, CurrentSp->GraphShape );
    }
  }
#endif
}

double LeafArea()
{
  return( CurrentTr->LeafArea );
}

double Biomass()
{
  return( CurrentTr->Biomass );
}


/*****************************************************************************

  climate.c

  Function Climate() which provides mean annual temperature, growing months,
  soil moisture index and number of frosts.

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601

  Description:

    Monthly estimates (from field data) of temperature, rainfall and
    evaporation are used to calculate mean annual temperature and
    soil moisture index (using a simple moisture model).  Also a
    continuous estimate of months warm enough for tree growth is
    calculated.  Frosts are calculated using a regression from monthly
    mean of daily minimum temperature.

  Global variables, constants and required functions are documented
  in the header files globals.h, vstructs.h and ranseeds.h

  Local variables are explained where they are declared

*****************************************************************************/
#include <math.h>
#include <stdio.h>
#include "globals.h"
#include "ranseeds.h"
#include "vstructs.h"

/* These symbols are used only in Climate() */
#define GrowTempBase   6.0  /* Temperature below which trees don't grow */
#define Latitude     -35.5  /* Latitude (south) of Brindabellas */

/**
  Mean & SD of monthly (Jul - Jun) total rainfall for the base station:
```

```
  Blundells Trig, at 1040 m
**/
static double RainMeanBase[12] = { 104.1, 133.1, 125.4, 127.4,  99.0,  79.6,
                                    85.9,  90.6,  87.4,  94.8,  95.1,  69.1 };
static double RainStDvBase[12] = {  84.4,  58.0,  62.8,  85.0,  65.9,  60.0,
                                    50.6,  77.9,  51.6,  88.8,  64.8,  30.5 };

/**
  Mean & SD of monthly (Jul - Jun) mean temperature for the base station:
  Bulls Head, at 1366 m
**/
static double TempMeanBase[12] = {  1.38,  2.56,  5.35,  8.41, 10.61, 13.87,
                                   15.53, 15.55, 12.85,  9.15,  5.68,  2.68 };
static double TempStDvBase[12] = {  1.03,  1.60,  1.33,  1.61,  1.25,  1.34,
                                    1.90,  1.38,  1.34,  1.10,  0.99,  1.14 };

/**
  Mean & SD of monthly (Jul - Jun) daily min temperature for the base
  station: Bulls Head, at 1366 m
**/
static double MinTMeanBase[12] = { -1.40, -0.58,  1.34,  3.97,  5.57,  8.24,
                                    9.89, 10.28,  8.10,  5.17,  2.55, -0.19 };
static double MinTStDvBase[12] = {  2.33,  2.92,  3.32,  3.99,  4.26,  3.91,
                                    3.89,  3.59,  3.46,  3.26,  3.28,  2.69 };

/**
  Mean & SD of monthly (Jul - Jun) total pan evaporation for the base station:
  Bulls Head, at 1366 m
**/
static double EvapMeanBase[12] = {  21.4,  36.1,  59.2,  97.3, 127.6, 180.5,
                                   164.2, 133.7, 107.7,  66.2,  35.5,  22.5 };
static double EvapStDvBase[12] = {   7.1,   7.6,  14.9,  24.2,  23.2,  28.8,
                                    37.1,  31.1,  14.5,  14.0,   5.5,   5.9 };

/**
  Climate estimate equations:  mean values from altitude difference
  between current site and base station.  These are not used anywhere
  else, but are here so they can be conveniently changed if necessary.
**/
#define RainEsti( B, A ) (B + 8.3333 - exp( 2.1203 - 2.0691e-3 * (A - 1040.0) ))
#define TempEsti( B, A ) (B + 5.7973e-3 * (1366.0 - A))
#define EvapEsti( B, A ) (B + 5.8818e-2 * (1366.0 - A))

/**
  If soil water is unknown, it is set to field capacity.  Initialize it
  to signify unknown
**/
static double SoilWater = -999.0;

double HeatRatio();  /* From solar.c */

void Climate( MeanTemp, GrowMonths, MoistureIndex, Frosts )
double *MeanTemp,        /* Mean annual temperature */
       *GrowMonths,      /* Number of months with T > 6 deg */
       *MoistureIndex;   /* Mean soil moisture index */
int    *Frosts;          /* Number of frosts this year */
{
  /* Local variables */
  double Temp,           /* Monthly mean temperature */
         MinT,           /* Monthly mean of daily min. temperature */
         z,              /* Z-score for linking min. and mean temperatures */
         Rain,           /* Monthly rainfall */
         PanEvap,        /* Monthly pan evaporation */
         PEvapo,         /* Monthly potential evapotranspiration */
         AEvapo,         /* Monthly actual evapotranspiration */
         SumMoistIndex,  /* Moisture index accumulator */
         MonthGrow,      /* Monthly growth proportion */
         PrevTemp;       /* Previous monthly temperature */
```

```
register int month;   /* Month iteration counter */

/* Initialization */
if( SoilWater == -999.0 )
  SoilWater = FieldCap;
*GrowMonths = 0.0;
*MeanTemp = 0.0;
SumMoistIndex = 0.0;
PrevTemp = GrowTempBase + 1.0;
*Frosts = 0;

/**
  Months loop:
  For each month (July - June)...
**/
for( month = 0; month < 12; ++month )
{
  /**
    Rainfall is estimated from a distribution calculated from a base value
    and altitude difference, and must be >= 0.0
  **/
  Rain = NormDev( RainEsti( RainMeanBase[month], Altitude ),
                  RainStDvBase[month], RSClim1, RSClim2 );
  if( Rain < 0.0 )
    Rain = 0.0;

  /**
    Temperatures are estimated from similar distributions: mean monthly
    and monthly mean of daily min are linked, so we use the same z-score
    from the normal distribution.  Both temperatures can be negative.
  **/
  z = NormDev( 0, 1, RSClim1, RSClim2 );
  Temp = TempEsti( TempMeanBase[month], Altitude ) +
         z * TempStDvBase[month];
  MinT = TempEsti( MinTMeanBase[month], Altitude ) +
         z * MinTStDvBase[month];

  /* Pan evaporation:  ditto (must not be negative) */
  PanEvap = NormDev( EvapEsti( EvapMeanBase[month], Altitude ),
                     EvapStDvBase[month], RSClim1, RSClim2 );
  if( PanEvap < 0.0 )
    PanEvap = 0.0;

  /* Adjust evaporation for aspect */
  PanEvap *= HeatRatio( month, Latitude, Slope, Aspect );

  /* Assume potential evapotransiration to be 0.8 of pan evap'n */
  PEvapo = 0.8 * PanEvap;

  /**
    If there's enough water, actual evapotranspiration is equal to
    potential, otherwise it is as much water as is there
  **/
  if( SoilWater + Rain > PEvapo )
    AEvapo = PEvapo;
  else
    AEvapo = SoilWater + Rain;

  /* Adjust soil water level */
  SoilWater += (Rain - AEvapo);
  if( SoilWater > FieldCap )
    SoilWater = FieldCap;

  /**
    If it is warm enough for growth, accumulate summed growth indices
    and counter of growing months
  **/
  /**
```

```
      An attempt at a continuous version of growing months.  It depends
      on the temperature of the previous month (initialized
      to GrowTempBase + 1).
    **/
    if( PrevTemp < GrowTempBase )
      if( Temp < GrowTempBase )
        MonthGrow = 0.0;
      else
        MonthGrow = (Temp - GrowTempBase) / (Temp - PrevTemp);
    else
      if( Temp < GrowTempBase )
        MonthGrow = (PrevTemp - GrowTempBase) / (PrevTemp - Temp);
      else
        MonthGrow = 1.0;
    PrevTemp = Temp;

    SumMoistIndex += MonthGrow * ((PEvapo > 0.0) ? (AEvapo / PEvapo) : 1.0);
    *GrowMonths += MonthGrow;
    *MeanTemp += Temp;

    /**
      Monthly frosts are calculated from monthly mean of daily min
      temperature (MinT) using a regression fit to data from 3 Met
      Bureau climatic stations: Canberra, Cooma and Khancoban.  Of
      course, monthly frosts can't be negative!  The equation:

        month frosts = 20.3 - 2.44 MinT

      is zero at MinT = 8.32 degrees.
    **/
    if (MinT < 8.32)
      *Frosts += 20.3 - 2.44 * MinT;

  }  /* Months loop */

  *MeanTemp /= 12.0;
  /**
    The annual moisture index is the average of that for the growing months
  **/
  if (*GrowMonths > 0.0)
    *MoistureIndex = SumMoistIndex / *GrowMonths;
  else
    *MoistureIndex = 0.0;

#if 0
  if (Debug)
  {
    fprintf(Debug, "%.1lf\t%6.4lf\t%.1lf\t%d\n", *MeanTemp, *MoistureIndex,
        *GrowMonths, *Frosts);
    fflush(Debug);
  }
#endif

} /* Climate() */


/*******************************************************************************

  solar.c

  Functions to calculate incoming solar radiation on a slope

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
```

```
   Canberra A.C.T.  2601
   Australia

 Description:

   This is an implementation of the method of

   Swift L.W. (1976) Algorithm for solar radiation on mountain
     slopes.  Water Resources Research 12, 108-112.

   The names of most variables are as in Swift (1976).  I don't fully
   understand how this works but have checked the correctness of it very
   carefully.

   Externally-known functions:

   SolarRad()    Radiation on the specified surface in cal/sq.cm/day
   SolarRatio()  Ratio of radiation on the specified surface to that on a
                 level one at the same location (radiation index)
   HeatRatio()   SolarRatio, adjusted by a time offset to allow for higher
                 heat load on west-facing slopes

   All three take arguments:

   Month      Southern Hemisphere growth year (Jul-Jun) month, numbered 0
              (Jul) to 11 (Jun).  Radiation is calculated for the middle
              day of the specified month.
   Latitude   Latitude of the site in degrees (negative for Southern
              Hemisphere).
   Slope      Slope of site in degrees.
   Aspect     Aspect or azimuth of site in degrees.

****************************************************************************/
#include <math.h>

#define RO         1.95
#define July16     197
#define DegToRad   57.29577951
#define Pi         3.141592654
#define TwoPi      6.283185308
#define PiOn2      1.570796327
#define TwlvOnPi   3.819718634  /* 12 / pi */

#define Func3( V, W, X, Y )  \
          R1 * ( sin( D ) * sin( W ) * (X - Y) * TwlvOnPi + \
          cos( D ) * cos( W ) * (sin( X + V ) - sin( Y + V )) * TwlvOnPi )

typedef enum { Absolute, Ratio } Option;

#ifdef LINT_ARGS
double Func2( double, double );
double Solar( int, double, double, double, double, Option );
#else
double Func2();
double Solar();
#endif

/* The hour offset between actual and equivalent slopes is known
   to all these routines */
static double L2;

/* Here are the externally-known routines */
double SolarRad( Month, Latitude, Slope, Aspect )
int Month;
double Latitude, Slope, Aspect;
{
  return( Solar( Month, Latitude, Slope, Aspect, Absolute ) );
}
```

```
double SolarRatio( Month, Latitude, Slope, Aspect )
int Month;
double Latitude, Slope, Aspect;
{
  return( Solar( Month, Latitude, Slope, Aspect, Ratio ) );
}

/* HeatRatio returns the solar ratio, adjusted by an allowance for the
   difference between east and west, which is calculated from L2, the
   hour offset (converted from radians to hours) between actual and
   equivalent slopes.  This allowance is calculated on the basis of 20%
   change in heat for an offset of 3 hours */
double HeatRatio( Month, Latitude, Slope, Aspect )
int Month;
double Latitude, Slope, Aspect;
{
  return( Solar( Month, Latitude, Slope, Aspect, Ratio ) *
          (1.0 - 0.0667 * L2 * TwlvOnPi) );
}

/* Here is the local routine which does the work */
static double Solar( Month, Latitude, Slope, Aspect, Choice )
int Month;
double Latitude, Slope, Aspect;
Option Choice;
{
  double D, D1, EE, J, L1, R1, R4, T, T0,
         T1, T2, T3, T6, T7, T8, T9;

  /* From growth season month numbers (0 = July, 6 = Jan, 11 = Jun),
     calculate Julian day as middle of the month */
  J = (Month * 30 + July16) % 365;

  /* Convert angles in degrees to radians */
  Latitude /= DegToRad;
  Slope /= DegToRad;
  Aspect /= DegToRad;

  /* L1 is the latitude of the equivalent slope */
  L1 = asin( cos( Slope ) * sin( Latitude ) +
             sin( Slope ) * cos( Latitude ) * cos( Aspect ) );
  D1 = cos( Slope ) * cos( Latitude ) -
       sin( Slope) * sin( Latitude ) * cos( Aspect );
  L2 = atan2( sin( Slope ) * sin( Aspect ), D1 );
  if( D1 < 0.0 )
    L2 += Pi;

  /* D is the solar declination, and EE the radius vector */
  D = asin( 0.39785 * sin( 4.868961 + 0.017203 * J +
                          0.033446 * sin( 6.224111 + 0.017202 * J ) ) );
  EE = 1.0 - 0.0167 * cos( (J - 3) * 0.0172 );

  /* R1 is the solar constant for 60 minutes */
  R1 = 60.0 * R0 / EE / EE;

  /* The Tx variables are hour angles:

          slope          rise    set
          ---------------------------
          horizontal      T0      T1
          actual slope    T2      T3
          equiv. slope    T6      T7
          actual slope,
            2nd rise/set  T8      T9
  */
  T = Func2( L1, D );
  T7 = T - L2;
```

```
  T6 = - T - L2;

  T = Func2( Latitude, D );
  T1 = T;
  T0 = - T;

  if( T7 < T1 )
    T3 = T7;
  else
    T3 = T1;
  if( T6 > T0 )
    T2 = T6;
  else
    T2 = T0;

  if( T3 < T2 )
    T2 = T3 = 0.0;
  T6 += TwoPi;
  if( T6 < T1 )
  {
    T8 = T6;
    T9 = T1;

    /* R4 is the potential solar radiation on the actual slope */
    R4 = Func3( L2, L1, T3, T2 );
    R4 += Func3( L2, L1, T9, T8 );
  }
  else
  {
    T7 -= TwoPi;
    if( T7 > T0 )
    {
      T8 = T0;
      T9 = T7;

      R4 = Func3( L2, L1, T3, T2 );
      R4 += Func3( L2, L1, T9, T8 );
    }
    else
      R4 = Func3( L2, L1, T3, T2 );
  }

  if( Choice == Absolute )
    return( R4 );
  else
  {
    double R3;

    R3 = Func3( 0.0, Latitude, T1, T0 );
    return( R4 / R3 );
  }
}

static double Func2( L, D )
double L, D;
{
  double T;

  /* We can't take a tan of pi/2 */
  if( L == PiOn2 )
    L += 0.03;
  T = - tan( L ) * tan( D );

  /* Nor can we get arc cos of a number outside [-1, 1] */
  if( T > 1.0 )
    T = 1.0;
  else if( T < -1.0 )
    T = -1.0;
```

```
  return( acos( T ) );
}

/******************************************************************************

  errz.c

  errz() function to report error messages and possibly stop the program

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601

  Description:

    Given a severity code (defined in globals.h) and a message, errz()
    prints the message and either returns or exits.

 ******************************************************************************/
#include <stdio.h>
#include "globals.h"
#if ! UNIX
# include <stdlib.h>
#endif

void errz( severity, message )
unsigned severity;
char *message;
{
  if( severity == FATAL )  /* This one's fatal */
  {
    fprintf( stderr, "\n\nFATAL ERROR>> %s -- stopping -- sorry\n\n", message );
    exit( FATAL );
  }
  else if( severity == WARNING )
  {
    fprintf( stderr, "\n\nWARNING>> %s\nRemainder of input line discarded\n",
             message );

    /* Discard the remainder of the current line of user input */
    DiscardCurLine();
  }

  return;
}


/******************************************************************************

  help.c

  Function Help() to print screens of help info

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601
    Australia
```

```
 Description:

   Nothing flashy, just formatted output.

 Global variables, constants and required functions are documented
 in the header files globals.h, vstructs.h and trees.h

 Local variables are explained where they are declared

*******************************************************************************/
#include <stdio.h>
#include "globals.h"
#include "vstructs.h"

/**
  Pause after a screenful of text.  Different compilers do this
  different ways.
**/
int ScreenPause()
{
  /* Pause if output to screen */
  if( Out == stdout )
  {
#if TC
    int c;

    printf( "Press any key to continue (ESC to cancel)..." );
    c = getch();
    printf( "\r                                            \r");
    if( c == 033 )
      return 0;
#else
    char dummy[11];

    printf( "Press <RETURN> to continue..." );
    gets( dummy );
    puts( "" );
#endif
  }

  return 1;
}

void Help()
{
  fputs( "\nCommands (case insensitive):\n", Out );
  fputs( "===========================\n", Out );

  fputs( "Run n\t\t\t\tRun program with current settings for n years\n", Out );
  fputs( "Stop\t\t\t\tStop the program\n", Out );

  fputs( "Input <filename>\t\tRedirect input stream (EOF returns to stdin)\n",
         Out );
  fputs( "Output <filename>\t\tRedirect output stream (\"*\" for stdout)\n",
         Out );
  fputs( "SpParams <filename>\t\tRead species parameters from file\n", Out );
  fputs( "StrtSite <filename>\t\tRead starting site from file\n", Out );
  fputs( "SiteEnvt <filename>\t\tRead site environment from file\n", Out );
  fputs(
    "RandSSet <seed no.> <seed>\tSet random no. seeds (-ve seed no. for all)\n",
    Out );
  fputs( "Print <variable> | ALL\t\tPrint values of variables\n", Out );
  fputs( "SiteList\t\t\tList trees on site\n", Out );
  fputs( "RepTime\t\t\t\tShow current time and simulation year\n", Out );
#if GRAPHICS
  fputs( "Picture\t\t\t\tShow current plot as a graphic image\n", Out );
#endif
```

```
  fputs( "Help\t\t\t\tPrint this help information\n\n", Out );

  if( ! ScreenPause() )
    return;

  fputs( "Variables (case insensitive), type:  <variable name> <new value>\n",
         Out );
  fputs( "==============================================================\n",
         Out );
  fputs( "PlotArea\t\tArea of plot (sq m)\n", Out );
  fputs( "Altitude\t\tAltitude of site (m)\n", Out );
  fputs( "Aspect\t\t\tAspect of site (degrees)\n", Out );
  fputs( "Slope\t\t\tSlope of site in (degrees)\n", Out );
  fputs( "k\t\t\tCompetition coefficient\n", Out );
  fputs( "CClWdth\t\t\tWidth of DBH classes for trees in comp'n (cm)\n", Out );
  fputs( "SuprMort\t\tAnnual mortality prob of suppressed trees\n", Out );
  fputs( "FieldCap\t\tField capacity of soil (mm)\n", Out );
  fputs( "MaxFuel\t\t\tFine fuel asymptotic max (t/ha)\n", Out );
  fputs( "FuelDeco\t\tFine fuel decomposition coefficient\n", Out );
  fputs( "FDIMean\t\t\tMean value of fire danger index on wildfire day\n", Out );
  fputs( "FDIStDv\t\t\tStd deviation of FDI on wildfire day\n", Out );
  fputs( "FireProb\t\tAnnual probability of wildfire\n", Out );

  if( ! ScreenPause() )
    return;

  fputs( "StrtYear\t\tStarting year of simulation\n", Out );
  fputs( "LeadTime\t\tLead time before some outputs (years)\n", Out );
  fputs( "PLinePlt\t\tPeriod between line plot lines (years)\n", Out );
  fputs( "PSiteLst\t\tPeriod between site listings (years)\n", Out );
  fputs( "PRepFire\t\tPeriod between fire reports (years)\n", Out );
  fputs( "LastFrYr\t\tYear of last fire\n", Out );
  fputs( "PageWdth\t\tWidth of output page (columns)\n", Out );
  fputs( "PRepTime\t\tPeriod between real time listings (years)\n", Out );
  fputs( "NSites1\t\t\tNo. sites/ha for tree estab't in most non-fire years\n", Out );
  fputs( "NSites2\t\t\tNo. sites/ha for estab't in 10% of non-fire years\n", Out );
  fputs( "FSites\t\t\tNo. sites/ha for estab't in fire years\n", Out );
  fputs( "YSiteSum\t\tYear to write site summary & stop program\n", Out );
  fputs( "SupprYrs\t\tYears slow growth before a tree marked suppressed\n", Out );

  fputs( "\nFlags (case insensitive), type:  <name> to set; No<name> to reset\n",
         Out );
  fputs( "==============================================================\n",
         Out );
  fputs( "Batch\t\t\tSet to remove prompting\n", Out );
  fputs( "Birth\t\t\tReset to stop tree establishment\n", Out );
  fputs( "SIMort\t\t\tReset to stop size-indep. mortality\n", Out );
  fputs( "SuMort\t\t\tReset to stop suppression mortality\n", Out );
  fputs( "LifeTabl\t\tSet to generate life table information\n", Out );
  fputs( "SeedDisp\t\tSet to add seed dispersal a la BRIND\n", Out );
#if GRAPHICS
  fputs( "Graphics\t\tSet for graphic output\n", Out );
#endif
  putc( '\n', Out);

  fflush( Out );
}


/***************************************************************************

  random.c

  BrRand() uniform random number generator and RandSSet() to set seeds

  Written by:

    Mike Strasser
```

```
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601
    Australia

  Description:

    Common-or-garden congruence method generator giving numbers in [0,1),
    adapted from:

      Press W.H. et al. (1986) Numerical Recipes, p. 197

    An array of seeds is kept and calls to BrRand pass an index into this
    array which is used to fetch the seed.  Keeping multiple seeds allows
    independence of program components which use random numbers.

    RandSSet() collects two numbers from the user: an index and a seed.
    With a valid index, the seed is set to the second argument.  If the
    index argument is negative, all seeds in the array are set, starting
    with the supplied one.

  Global variables, constants and required functions are documented
  in the header files globals.h, vstructs.h and trees.h

  Local variables are explained where they are declared

*******************************************************************************/
#include <stdio.h>
#include "globals.h"
#include <math.h>
#include "vstructs.h"

#define M 714025L
#define A 1366L
#define C 150889L

static long RandSeeds[] = {
  520189L,   /* RSClim1    0 */
  156992L,   /* RSClim2    1 */
  178692L,   /* RSWFire1   2 */
   16367L,   /* RSWFire2   3 */
    2500L,   /* RSPFire1   4 */
  541403L,   /* RSPFire2   5 */
  337106L,   /* RSPrWFire  6 */
  535602L,   /* RSPrPFire  7 */
  573530L,   /* RSPrFMort  8 */
  106037L,   /* RSPrNMort  9 */
  263545L,   /* RSPrSMort 10 */
  199454L,   /* RSBirth1  11 */
  120388L,   /* RSBirth2  12 */
  570801L,   /* RSBirth3  13 */
  544611L,   /* RSBirth4  14 */
  360230L,   /* RSStSite  15 */
   82932L,   /* RSFStDth  16 */
   47914L,   /* RSLigSprt 17 */
  416505L,   /* RSClim3   18 */
   26513L    /* RSClim4   19 */

#if GRAPHICS
  , 655746L    /* RSXPos     20 */
};

#define NRanSeeds 21
#else
};
```

```
#define NRanSeeds 20
#endif

double BrRand( SeedIndex )
int SeedIndex;
{
  double result;  /* Storage for the outcome */

  RandSeeds[SeedIndex] = (A * RandSeeds[SeedIndex] + C) % M;
  result = (double) RandSeeds[SeedIndex] / (double) M;

  return result;
}

void RandSSet()
{
  int   ISeed;  /* Index into array */
  long  Seed;   /* Seed or starting seed */
  int   i;      /* Loop variable */

  if( ReadInt( In, &ISeed, "Rand. no. index" ) != 1 )
    errz( WARNING, "Error reading random no. seed index" );
  else
    if( ReadLong( In, &Seed, "Rand. no. seed" ) != 1 )
      errz( WARNING, "Error reading random no. seed" );
    else
      if( ISeed < 0 || ISeed >= NRanSeeds )
      {
        for( RandSeeds[0] = Seed, i = 1; i < NRanSeeds; i++ )
          RandSeeds[i] = BrRand( i - 1 ) * (double) M;
        BrRand( NRanSeeds - 1 );
      }
       else
         RandSeeds[ISeed] = Seed;

}
/******************************************************************************

  normdev.c

  Function NormDev() to return a random normal deviate.

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601
    Australia

  Description:

    This method has been handed down to me via a FORTRAN function.  I
    don't know how it works, but if you really want to know, I bet it is
    in "Numerical recipes" (see file random.c for full reference).

  Global variables, constants and required functions are documented
  in the header file globals.h.

  Local variables are explained where they are declared

******************************************************************************/
/**
  Try this some day:

      Unif = BrRand( SomeSeed );
```

```
      Gauss = (pow( Unif, 0.135 ) - pow( (1 - Unif), 0.135 )) / 0.1975;

  which will vary approx. as N(0,1), but with tails restricted to < 5(?)
  SD.  (Then,

      return( Mean + Gauss * StdDev );

  plus zero check if desired.)  This has 2 nasty floating point calls
  instead of 3, but I think there must be a better way, with none.

                                                 mjs 7th Nov 1988
**/
#include <stdio.h>
#include "globals.h"
#include <math.h>

#define TWOPI 6.2831853  /* 2 * pi */

double NormDev( Mean, StdDev, ISeed1, ISeed2 )
double Mean, StdDev;  /* These describe the distribution */
int ISeed1, ISeed2;   /* These are indices to the random seed array used by
                         BrRand(), not the seeds themselves */
{
  double Gauss;

  Gauss = Mean + StdDev * sqrt( -2.0 * log( BrRand( ISeed1 ) ) )
                        * sin( TWOPI * BrRand( ISeed2 ) );
  return( Gauss );
}
```

C.5 Input/output functions

```
/****************************************************************************

  setvars.c

  Function to accept user input for controlling program and setting
  variables

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601
    Australia

  Description:

    Function SetVars() accepts user instructions for setting values of
    variables and for program control.

    SetVars() repeatedly polls the user for instructions.  On receipt of a
    keyword it looks it up.  Special keywords are:

      "run": read an integer number of years and return that to main() to
        do the simulation run

      "stop": return to main() to stop the program

      "geology": read a string, if first letter is 'g' then set geology
        to "Granites", otherwise to "Sediments"

    If the keyword is found as a name in one of the arrays, the following
    happens:
```

```
     RYears[]: Read an integer and set the variable part of that structure
       in the array to it.

     Envt[]: Read a double and set the variable part of that structure to
       it.

     Funcs[]: Call the function pointed to, no arguments are passed.

     RunFlags[]: The keyword is recognised as "flag" or "noflag" and the
       flag is set accordingly.

     IOUnits[]: Another keyword is read:

       "*" : use stdin or stdout as appropriate
       "off": return to default (which is NULL for many units)
       filename: open filename and redirect for reading or writing
       +filename: append to filename

   If an EOF is encountered (SetVars reads from the unit In), close the
   input file if not stdin, else stop the program.

   NOTE that input redirections do not nest, at EOF from a file, In is
   always reset to stdin, not to any previous file (which is closed and
   forgotten).

   THERE IS A BUG in that if you type

     input stuff.dat run 100

   the program will run for 100 years before reading stuff.dat because
   ReadString() etc. only read from In when a line ends: they still have
   the line containing "run 100" after In is redirected to stuff.dat
   before they actually read from that file.

   NOTE that keyword searches are case-insensitive and limited to a
   maximum of 8 characters.

  Required functions:

   strnicmp() to compare limited-length strings case-insensitive.

   ReadString(), ReadDouble(), ReadInt() from simplepa(rse).c to accept
   user input of various types.

  Global variables, constants and required functions are documented
  in the header files globals.h, vstructs.h and trees.h

  Local variables are explained where they are declared

*****************************************************************************/
#define SETVARS /* Defined so some info in vstructs.h is ignored.  See
                   that file for details */

#include <stdio.h>
#include "globals.h"
#include <string.h>
#include "vstructs.h"

#if GRAPHICS
# include "brgraph.h"
#endif

/**
  Here are the structures defined in vstructs.h which allow the user to
  set values of internal program variables from the command line.  The
  arrays of them are also initialised here.  These are described in
  detail in vstructs.h.
**/
```

```
/**
  Starting year, years to run, output periods and graphics flag
**/
IntVar RYears[] = {
  { "StrtYear", 0    },
  { "PLinePlt", 0    },
  { "PSiteLst", 0    },
  { "PRepFire", 0    },
  { "LastFrYr", 0    },
  { "PageWdth", 80   },
  { "LeadTime", 0    },
  { "PRepTime", 0    },
  { "YSiteSum", -1   },
  { "Replicat", 1    },
  { "FireYr1",  0    },
  { "FireYr2",  0    }
};

/**
    Run control flags
**/
FlagVar RunFlags[] = {
  { "Batch",    0 },
  { "Birth",    1 },
  { "SIMort",   1 },
  { "SuMort",   1 },
  { "LifeTabl", 0 },
  { "SeedDisp", 1 },
  { "NewSuppr", 0 },
  { "ClimGrow", 1 },
  { "Graphics", 0 }
};


/**
  Floating point variables...
**/
FloatVar Envt[] = {
  { "PlotArea", 833.0  },
  { "Altitude", 1100.0 },
  { "Aspect",   135.0  },
  { "Slope",    0.0    },
  { "k",        75.2   },
  { "CClWdth",  0.5    },
  { "SuprMort", 0.368  },
  { "SurvProp", 0.2    },
  { "FieldCap", 65.0   },
  { "MaxFuel",  23.5   },
  { "FuelDeco", 0.23   },
  { "FDIMean",  18.0   },
  { "FDIStDv",  12.0   },
  { "FireProb", 0.02   },
  { "SeedProd", 1.0    },
  { "MaxBiom",  1000.0 },
  { "MinMMI",   0.66   },
  { "ROSadj",   7.0    },
  { "NSites1",  50.0   },
  { "NSites2",  350.0  },
  { "FSites",   2500.0 },
  { "SupprYrs", 1.0    },
  { "SupprInc", 0.1    },
  { "FireStim", 3.0    }
};

/**
    Geology type
**/
char Geology[10] = "Sediments";
```

```
/**
  Functions to execute...
**/
FuncVar Funcs[] = {
  { "SpParams", SpParams },
  { "StrtSite", StrtSite },
  { "SiteEnvt", SiteEnvt },
  { "RandSSet", RandSSet },
  { "Print",    PrintVar },
  { "SiteList", SiteList },
  { "RepFires", ReportFires },
  { "Help",     Help     },
  { "RepTime",  ReportTime },
  { "InitLT",   InitLifeTable }
#if GRAPHICS
, { "Picture",  Picture }
};
# define NFuncs 11
#else
};
# define NFuncs 10
#endif

/**
  I/O units...
**/
IOUnitVar IOUnits[] = {
#if VMS  /* VAX/VMS compiler doesn't like stdin/stdout in initialization */
  { "Input",   Reading, NULL,   NULL,   "stdin"                },
  { "Output",  Writing, NULL,   NULL,   "stdout"               },
  { "Debug",   Writing, NULL,   NULL,   "The wild blue yonder" },
  { "GrowOut", Writing, NULL,   NULL,   "Past the black stump" },
  { "SummOut", Writing, NULL,   NULL,   "stdout"               },
  { "TimeOut", Writing, NULL,   NULL,   "stdout"               },
  { "FireOut", Writing, NULL,   NULL,   "stdout"               }
#else
  { "Input",   Reading, stdin,  stdin,  "stdin"                },
  { "Output",  Writing, stdout, stdout, "stdout"               },
  { "Debug",   Writing, NULL,   NULL,   "The wild blue yonder" },
  { "GrowOut", Writing, NULL,   NULL,   "Past the black stump" },
  { "SummOut", Writing, stdout, stdout, "stdout"               },
  { "TimeOut", Writing, stdout, stdout, "stdout"               },
  { "FireOut", Writing, stdout, stdout, "stdout"               }
#endif
};

int SetVars()
{
  /* Local variables */
  char Keyword[11];  /* Input keyword */
  char Filename[32]; /* Filename for some actions */
  char InString[13]; /* Input string for others */

  /* Status variable used when scanning for recognisable input */
  enum { NotFound, Found, Error, Skip, OpenFail } Status = NotFound;

  FILE * TempFile; /* Place holder for swapping I/O units */
  register int i;  /* Iteration counter (what else would it be?) */

  /**
    Variable/function reading loop...
  **/
  while( 1 )
  {
    /**
      Read the variable.  On EOF:
         If input is from a file, reset In to be stdin
```

```
     If input is from stdin, stop the program
**/
while( ReadString( In, Keyword, "BRIND" ) == EOF )
{
  if( In != stdin )
  {
    fclose( In );
    puts( "\nInput redirected to stdin\n" );
    In = stdin;
  }
  else
    return( Stop );
}

/* This will stop the program */
if( ! strnicmp( Keyword, "Stop", 8 ) )
  return( Stop );

/* This will run the program for the specified number of years */
if( ! strnicmp( Keyword, "Run", 8 ) )
{
  ReadInt( In, &RunYears, "Years to run" );
  return( Run );
}

/**
   Check for I/O Unit redirection.  Options are:
     <I/O name> <filename>  -- read from/write to a file (on input, EOF
                               restores stream to stdin)
     <I/O name> +<filename> -- append to file (output only)
     <I/O name> *           -- write to stdout (output only)
     <I/O name> Off         -- reset unit to default (output only)
**/
for( i = 0; i < NIOUnits; i++ )
  if( ! strnicmp( Keyword, IOUnits[i].Name, 8 ) )
  {
    Status = Found;
    break;
  }
if( Status == Found )
{
  ReadString( In, Filename, "File/stream" );
  if( ! strcmp( Filename, "*" ) )
    if( IOUnits[i].Direction == Reading )
      TempFile = stdin;
    else
      TempFile = stdout;
  else if( ! strnicmp( Filename, "Off", 8 ) )
      TempFile = IOUnits[i].DfltPtr;
  else if( IOUnits[i].Direction == Reading )
  {
    if( (TempFile = fopen( Filename, "r" )) == NULL )
      Status = OpenFail;
  }
  else if( Filename[0] == '+')
  {
    if( (TempFile = fopen( &Filename[1], "a" )) == NULL )
      Status = OpenFail;
  }
  else if( (TempFile = fopen( Filename, "w" )) == NULL )
    Status = OpenFail;

  if( Status != OpenFail )
  {
    if( IOUnits[i].CurPtr != stdin && IOUnits[i].CurPtr != stdout )
      fclose( IOUnits[i].CurPtr );
    IOUnits[i].CurPtr = TempFile;
    strcpy( IOUnits[i].Filename, Filename );
```

```
    }
    goto EndLoop;
  }

  /* Check for geology specification */
  if( ! strnicmp( Keyword, "Geology", 8 ) )
  {
    Status = Found;
    fscanf( In, "%s", InString );
    if( InString[0] == 'g' || InString[0] == 'G' )
      strcpy( Geology, "Granites" );
    else
      strcpy( Geology, "Sediments" );
    goto EndLoop;
  }

  /* Check for function specifiers.  When one is found, call that function. */
  for( i = 0; i < NFuncs; i++ )
    if( ! strnicmp( Keyword, Funcs[i].Name, 8 ) )
    {
      Status = Found;
      break;
    }
  if( Status == Found )
  {
    (*Funcs[i].FuncPtr)();
    goto EndLoop;
  }

  /* Check for environmental variables.  When one is found, read its new
     value. */
  for( i = 0; i < NEnvVars; i++ )
    if( ! strnicmp( Keyword, Envt[i].Name, 8 ) )
    {
      Status = Found;
      break;
    }
  if( Status == Found )
  {
    if( ReadDouble( In, &Envt[i].FVal, "New value" ) != 1 )
      Status = Error;
    goto EndLoop;
  }

  /* Check for running/output years specification.  Ditto. */
  for( i = 0; i < NRYears; i++ )
    if( ! strnicmp( Keyword, RYears[i].Name, 8 ) )
    {
      Status = Found;
      break;
    }
  if( Status == Found )
  {
    if( ReadInt( In, &RYears[i].IVal, "New value" ) != 1 )
      Status = Error;
    /**
      Special case:  if StrtYear is specified, reset the global variable
      Year.
    **/
    if( i == 0 )
      Year = RYears[0].IVal;
    goto EndLoop;
  }

  /* Check for running flags, as Name or "no"+Name */
  /* Setting */
  for( i = 0; i < NRunFlags; i++ )
    if( ! strnicmp( Keyword, RunFlags[i].Name, 8 ) )
```

```
      {
        Status = Found;
        break;
      }
    if( Status == Found )
    {
      RunFlags[i].FlVal = 1;
      /**
        Special case:  if Batch was specified, prompts off.  Also send
        a newline to Out so the next line written starts at the left
        margin (the file pointer was at the end of a prompt).
      **/
      if( i == 0 )
      {
        TogglePrompts( 0 );
        putc( '\n', Out );
      }
      goto EndLoop;
    }
    /* Resetting */
    for( i = 0; i < NRunFlags; i++ )
      if( (! strnicmp( Keyword, "no", 2 )) &&
          (! strnicmp( Keyword + 2, RunFlags[i].Name, 8 )) )
      {
        Status = Found;
        break;
      }
    if( Status == Found )
    {
      RunFlags[i].FlVal = 0;
      /**
        Special case:  if NoBatch was specified, turn prompts on.
      **/
      if( i == 0 )
        TogglePrompts( 1 );
      goto EndLoop;
    }

    EndLoop:

    /* Check Status and report if necessary */
    if( Status == Found )
      Status = NotFound;
    else if( Status != Skip )
    {
      if( Status == NotFound )
        fprintf( stderr, "'%s' not a recognized keyword\n", Keyword );
      else if( Status == Error )
        fprintf( stderr, "Error reading number:  '%s' not set\n", Keyword );
      else if( Status == OpenFail )
        fprintf( stderr, "Open failure on '%s'\n", Filename );

      Status = Skip;
    }

  }  /* End of Variable/function reading loop */
}


/******************************************************************************

  simplepa(rse).c

  Simple command-line parsing functions which issue prompts

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
```

```
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601
    Australia

  Description:

    Command-line parsing functions.  They read information from the
    string CurLine which is replenished from the specified file stream
    when necessary.  A prompt is issued to stdout if the flag
    IssuePrompts is true when fetching a new line.

    THERE IS A BUG:  if the calling program changes the stream from which
    commands are to be read, the change does not come into effect until
    CurLine (read from the old stream) is fully parsed.  I can't be
    bothered fixing this.

    ReadString()  gets the next whitespace-delineated token
    ReadInt()     calls ReadString() and converts it to an int
    ReadDouble()  calls ReadString() and converts it to a double
    ReadLong()    calls ReadString() and converts it to a long int

    If any of these functions has problems it returns EOF (otherwise 1)

    DiscardCurLine()  discards the remainder of CurLine (called when an
                      error occurs)
    TogglePrompts()   controls whether prompts are issued when ReadString()
                      is called
    DumpData()        dumps CurLine info to stderr, called on error

  Global variables, constants and required functions are documented
  in the header file globals.h.

  Local variables are explained where they are declared

******************************************************************************/
#include <stdio.h>
#include "globals.h"
#include <ctype.h>
#include <math.h>
#if ! UNIX
# include <stdlib.h>
#endif

#define LLINE 133  /* Length of the command line read in + 1 */

/**
  These objects are private to these
  functions.
**/
static char CurLine[LLINE] = ""; /* Current command line */
static char *CurPtr = CurLine;   /* Pointer to current position in CurLine */
static int IssuePrompt = 1;      /* Flag: issue prompts or not */

/**
  ReadString

  Parse the whitespace-delineated string from the current line.  At the
  end of the current line, prompt for and get a new one.
**/
int ReadString( fp, string, prompt )
FILE *fp;
char *string, *prompt;
{
  register int i;

  /**
```

```
    Skip blanks and tabs
  **/
  while( *CurPtr == ' ' || *CurPtr == '\t' )
    CurPtr++;

  /**
    Get a new line if necessary.  Prompt if input stream is stdin.
  **/
  while( ! *CurPtr || *CurPtr == '\n' )
  {
    CurPtr = CurLine;
    if( IssuePrompt && fp == stdin )
      printf( "%s> ", prompt );
    if( fgets( CurLine, LLINE, fp ) == NULL )
    {
      CurLine[0] = '\0';
      return( EOF );
    }
  }

  /**
    Find first non-whitespace character.
  **/
  while( isspace( *CurPtr ) )
    CurPtr++;

  /**
    Add non-whitespace characters to string.  Stop at whitespace or EOL.
    Null-terminate string.
  **/
  for( i = 0; *CurPtr && ! isspace( *CurPtr ); i++ )
    string[i] = *CurPtr++;
  string[i] = '\0';

  return( 1 );
}

/**
  ReadInt

  Call ReadString, then put the returned string into an integer or return EOF
**/
int ReadInt( fp, number, prompt )
FILE *fp;
int *number;
char *prompt;
{
  char str[41];

  if( ReadString( fp, str, prompt ) == EOF )
    return( EOF );

  *number = atoi( str );
  return( 1 );
}

/**
  ReadDouble

  Call ReadString, then put the returned string into a double or return EOF
**/
int ReadDouble( fp, number, prompt )
FILE *fp;
double *number;
char *prompt;
{
  char str[41];
  double atof();
```

```
  if( ReadString( fp, str, prompt ) == EOF )
    return( EOF );

  *number = atof( str );
  return( 1 );
}

/**
  ReadLong

  Call ReadString, then put the returned string into a long or return EOF
**/
int ReadLong( fp, number, prompt )
FILE *fp;
long *number;
char *prompt;
{
  char str[41];
  long atol();

  if( ReadString( fp, str, prompt ) == EOF )
    return( EOF );

  *number = atol( str );
  return( 1 );
}

/**
  DiscardCurLine

  The name says it all!
**/
void DiscardCurLine()
{
  CurLine[0] = '\0';
  CurPtr = CurLine;
}
/**
  TogglePrompts

  Ditto.
**/
void TogglePrompts( OnOff )
int OnOff;
{
  IssuePrompt = OnOff;
}

/**
  DumpData

  Temporary function to dump current data to stderr.
**/
void DumpData()
{
  fprintf(stderr, "Current line >>%s<<\nCurrent pointer position:  %d\n",
      CurLine, CurPtr - CurLine);
}


/*****************************************************************************

  lifetable.c

  Functions LifeTable() and InitLifeTable() for handling of life tables.

  Written by:
```

```
    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601
    Australia

  Description:

  LifeTable() records fates of living trees each year after the lead time
    has elapsed (lead time is set by the user).  Living trees either
    shift into the next size class (5 cm class width) or stay put.
    LifeTable() also accumulates biomasses to the global variable
    RunningTotalBiomass for reporting by SiteSummary().

  InitLifeTable() initialises each species' life table, and
    RunningTotalBiomass to zero.

  Global variables, constants and required functions are documented
  in the header files globals.h, vstructs.h and trees.h

  Local variables are explained where they are declared

*****************************************************************************/
#include <stdio.h>
#include "globals.h"
#include <math.h>
#if ! UNIX
# include <stdlib.h>
#endif
#include "trees.h"
#include "vstructs.h"

void LifeTable()
{
  int olddc;  /* DBH class of tree's OldDBH */
  int dc;     /* DBH class of tree's (current) DBH */
  register Species *Sp;  /* Iterative pointer */
  register Tree *Tr;     /* Ditto */

  /* Only accumulate data if the lead time has passed */
  if( Year > LeadTime )

    /* For each tree in each species... */
    for( Sp = FirstSpecies(); Sp != NULL; Sp = NextSpecies() )
      for( Tr = FirstTree(); Tr != NULL; Tr = NextTree() )
      {
        /**
          Calculate DBH classes of previous and current DBH, limiting
          class number to a maximum of 40 (DBH > 200 cm)
        **/
        olddc = (Tr->OldDBH > 200.0) ? 40 : Tr->OldDBH / 5.0;
        dc = (Tr->DBH > 200.0) ? 40 : Tr->DBH / 5.0;

        if( dc == olddc )  /* No change in DBH class this year */
          Sp->LifeTable[olddc][stayedput]++;
        else
          Sp->LifeTable[olddc][tonextclass]++;

        /* Add each tree to the running total of biomass */
        RunningTotalBiomass += Biomass();
      }

} /* LifeTable() */

void InitLifeTable()
```

```
{
  /* These are all iterative counters */
  register int d, f;
  register Species *Sp;

  for( Sp = FirstSpecies(); Sp != NULL; Sp = NextSpecies() )
    for( d = 0; d < MAXDCLASS; d++ )
      for( f = 0; f < MAXFATE; f++ )
        Sp->LifeTable[d][f] = 0;

  RunningTotalBiomass = 0.0;
}

/******************************************************************************

  lineplot.c

  Line printer and text screen continuous graphical output functions.

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601
    Australia

  Acknowledgements:

    This code is a port of a FORTRAN subroutine devised by Ian Noble

  Description:

    LinePlot() writes one line of information for the given integer X
      value, comprising an array of Y values and a character string
      label.  The plot runs with the X axis down the page and Y across.
      Each Y value has its own scale, set in InitLiPl(), which must be
      called before any calls to LinePlot().  LinePlot() plots a symbol
      for each Y value (set in InitLiPl()).  If two symbols fall on the
      same place, they are flagged on the RHS, as are undershoots and
      overshoots.

    InitLiPl() initialises things for calls to LinePlot().  They are the
      plot width, the number of Y values and an array of LiPlVars
      (defined in lipldef.h), structs containing names, symbols, min and
      max values for each Y variable.

    Example output from LinePlot():

 R   3500+A#-------+---------+-*^-----[+---------+---------+R>
 R       | A       |         |   *     |[        |         |R>,A$#=,*^=
 R       | A       |         |   ^ *   |   [     |         |R>,A#=
 R       |  A#     |  $      |    ^  * |     [   |         |R>
 R       |$# A     |         |     ^  *|       [ |         |R>
 R   3550+#-A------+---------+-----^---+*--------[---------+R>,A$=

    from the years (X values 3500 to 3550) every 10 years.  LinePlot()
    writes a horizontal line and X value every 5 lines.  Each symbol
    represents a non-zero Y value: zero values are not plotted.  In all
    lines, the value for species R is greater than the range set for it in
    InitLiPl() so it is marked with "R>".  The second line shows two
    cases of overprinting:  firstly "$" and "#" coincident with "A"
    (shown as "A$#=") and "^" with "*" (shown as "*^=").  The label "R"
    appearing on the left is an argument to LinePlot() sent by the
    calling function.
```

```
  Global variables, constants and required functions are documented
  in the header files globals.h, vstructs.h and lipldef.h

  Local variables are explained where they are declared

*****************************************************************************/
#include <stdio.h>
#include "globals.h"
#include "lipldef.h"
#include "vstructs.h"

#define MAXWIDTH 210
#define LAST  -1
#define UNDER -2
#define OVER  -3

static int LinesPlotted;         /* How many lines plotted since init */
static int NPlotVars = 0;        /* Number of Y variables */
static int PlotWidth;            /* Width in chars on the page */
static LiPlVar *PlotVars = NULL; /* Array of structs describing Y vars,
                                    defined in lipldef.h */

/************/
/* LinePlot */
/************/
void LinePlot( XVal, YVals, Label )
char *Label;    /* Label for the beginning of the line */
double YVals[]; /* Array of Y values this call */
int XVal;       /* Current X value */
{
  char PlotLine[MAXWIDTH-29]; /* Line which is output to screen/printer */
  enum {
    None,
    Some,
    Printed
  } Status; /* Status variable used when writing out under- and
               overshoots and coincident symbols */
  register int i, j, k; /* Misc iteration counters */

  /* Blank the line */
  for( i = 0; i < PlotWidth; i++ )
    PlotLine[i] = ' ';

  /**
    Main variable loop.  Here PlotLine is filled up with symbols and Y
    vars are marked as undershoots etc.  For coincident points the line
    is filled with the symbol of the first Y var.
  **/

  for( i = 0; i < NPlotVars; i++ )
  {
    /* Calculate position of this var in line from Y val and bounds */
    PlotVars[i].LinePos = PlotWidth * (YVals[i] - PlotVars[i].LBound) /
                          (PlotVars[i].UBound - PlotVars[i].LBound);

    /* Set the list pointer to default value */
    PlotVars[i].ListPtr = LAST;

    /* Undershoot */
    if( PlotVars[i].LinePos < 0 )
      PlotVars[i].ListPtr = UNDER;

    /* Overshoot */
    else if( PlotVars[i].LinePos >= PlotWidth )
      PlotVars[i].ListPtr = OVER;

    /* Overlap:  for the previous variable j plotted on this point,
       PlotVars[j].ListPtr stores i, thus linking them in a list */
```

```
    else if( PlotLine[PlotVars[i].LinePos] != ' ' )
    {
      for( j = i - 1; j >= 0; j-- )
        if( PlotVars[i].LinePos == PlotVars[j].LinePos )
          break;
      PlotVars[j].ListPtr = i;
    }

    /* If the symbol falls on an origin which is zero, don't plot it */
    else if( PlotVars[i].LinePos != 0 || PlotVars[i].LBound != 0.0 )
      PlotLine[PlotVars[i].LinePos] = PlotVars[i].Symbol;

  } /* End of main variable scan loop */

  /**********/
  /* Output */
  /**********/

  /* Label */
  fprintf( Out, "%-3.3s", Label );

  /* XVal and horizontal line every fifth call */
  if( ! (LinesPlotted % 5) )
  {
    /* Print out XVal */
    if( XVal )
      fprintf( Out, "%5d", XVal );
    else
      fputs( "    0", Out );

    /* Fill empty part of line with +---------+---------+ etc. */
    for( i = 0; i < PlotWidth; i++ )
      if( PlotLine[i] == ' ' )
        if( i % 10 )
          PlotLine[i] = '-';
        else
          PlotLine[i] = '+';
  }
  else
  {
    /* Put out spaces instead of number */
    fputs( "     ", Out );

    /* Otherwise, just |        |        | */
    for( i = 0; i < PlotWidth; i++ )
      if( PlotLine[i] == ' ' )
        if( !(i % 10) )
          PlotLine[i] = '|';
  }

  /* Write out plot line */
  PlotLine[PlotWidth] = '\0';
  fputs( PlotLine, Out );

  /**
    Write out undershoots.
    The variable Status is used in each case to signal whether any
    undershoots (or overshoots etc. below) are found.  If they're found
    then put them out.
  **/
  Status = None;
  for( i = 0; i < NPlotVars; i++ )
    if( PlotVars[i].ListPtr == UNDER )
    {
      putc( PlotVars[i].Symbol, Out );
      PlotVars[i].ListPtr = LAST;
      Status = Some;
    }
```

```
  if( Status == Some )
  {
    putc( '<', Out );
    Status = Printed;
  }

  /* Write out overshoots */
  for( i = 0; i < NPlotVars; i++ )
    if( PlotVars[i].ListPtr == OVER )
    {
      if( Status == Printed )
        putc( ',', Out );
      putc( PlotVars[i].Symbol, Out );
      PlotVars[i].ListPtr = LAST;
      Status = Some;
    }
  if( Status == Some )
  {
    putc( '>', Out );
    Status = Printed;
  }

  /**
    Write out overlaps.  This is a bit trickier because each one in the
    list must be shown.
  **/
  for( i = 0; i < NPlotVars; i++ )
    if( PlotVars[i].ListPtr != LAST )
    {
      if( Status == Printed )
        putc( ',', Out );
      putc( PlotVars[i].Symbol, Out );
      Status = Printed;
      j = i;
      do {
        k = PlotVars[j].ListPtr;
        putc( PlotVars[k].Symbol, Out );
        PlotVars[j].ListPtr = LAST;
        j = k;
      } while( PlotVars[j].ListPtr != LAST );
      putc( '=', Out );
    }

  /* Tidy up */
  LinesPlotted++;
  putc( '\n', Out );

  fflush( Out );

}  /* LinePlot() */

/**
  InitLiPl:  Initialize plotting and write out heading information with
  names, symbols and ranges.
  This must be called before LinePlot
**/
void InitLiPl( PageWidth, YVars, NYVars )
int PageWidth;    /* Width of page in characters */
int NYVars;       /* No. of Y vars */
LiPlVar YVars[];  /* Array of Y var description structures */
{
  int NHeadCols;        /* Number of columns of headings */
  register int i = 0;   /* Iteration counter */

  if( PageWidth > MAXWIDTH )
    PageWidth = MAXWIDTH;

  /* Calculate plot width to allow margins on left and right */
```

```
  PlotWidth = ( (int) (PageWidth / 10) - 3 ) * 10 + 1;
  NPlotVars = NYVars;
  PlotVars = YVars;

  /* Calc. no. of heading columns */
  NHeadCols = PageWidth / 30;
  if( NHeadCols < 1 )
    NHeadCols = 1;

  /* Initialise line counter */
  LinesPlotted = 0;

  /* Put out heading information */
  fputs( "\n\tPlotting:  Name, Symbol, Origin, Max.\n\n", Out );
  while( i < NYVars )
  {
    fprintf( Out, "%-6s %c %8.2f %8.2f", YVars[i].Name, YVars[i].Symbol,
                                          YVars[i].LBound, YVars[i].UBound );
    if( ++i % NHeadCols && i < NYVars )
      fputs( " :  ", Out );
    else
      putc( '\n', Out );
  }
  putc( '\n', Out );
}

/*****************************************************************************

  output.c

  General output function, calling specific ones as needed

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601
    Australia

  Description:

    Called by main() in newbrind.c with an argument specifying which
    options are to be output this call.

    It will call InitLiPl(), LinePlot(), ReportFires(), SiteList() or
    ReportTime() as necessary.

  Global variables, constants and required functions are documented
  in the header files globals.h, vstructs.h and trees.h

  Local variables are explained where they are declared

*****************************************************************************/
#include <stdio.h>
#include "globals.h"
#if ! UNIX
# include <stdlib.h>
#endif
#include <string.h>
#include "lipldef.h"
#include "outoptns.h"
#include "trees.h"
#include "vstructs.h"

/* Define these if necessary */
#ifndef min
```

```
# define min( a, b ) ((a < b) ? (a) : (b))
#endif
#ifndef max
# define max( a, b ) ((a > b) ? (a) : (b))
#endif

/**
  Stuff for initialisation of line plotting.  We need one variable for
  each species plus NExtrPlVr extra.
**/
static LiPlVar *PlotParams = NULL;
#define NExtrPlVr 5
enum {
  TotBio, /* Total plot biomass */
  NSprsd, /* Number of suppressed trees */
  TotalN, /* Total # of trees */
  LfArea, /* Leaf area */
  BigsTr  /* DBH of biggest tree */
};
static LiPlVar ExtraPlotVars[NExtrPlVr] = {
  { 0.0, 1000.0, 0, 0, "TotBio", '*' },
  { 0.0,   80.0, 0, 0, "NSuprs", '$' },
  { 0.0,  400.0, 0, 0, "Total#", '#' },
  { 0.0,    8.0, 0, 0, "LfArea", '[' },
  { 0.0,  200.0, 0, 0, "BigsTr", '^' }
};

void Output( Option )
unsigned Option;  /* Option flag:  see outoptns.h for definitions */
{
  register int i; /* Iteration counter */
  register Species *Sp;  /* Species iteration counter */
  register int NSpec = NSpecies();  /* Number of species */

#if GRAPHICS
  /**
    None of this stuff if we're in graphics mode and Out is stdout, 'cos
    it will mess up the screen.
  **/
  if( Graphics && Out == stdout )
    return;
#endif

  /* Initialise line plotting */
  if( Option & OutInit )
  {
    if( PlotParams != NULL )
      free( (char *) PlotParams );

    /* Allocate memory for line plot params array */
    if( (PlotParams = (LiPlVar *) calloc( NSpec + NExtrPlVr, sizeof( LiPlVar ) ))
        == NULL )
      errz( FATAL, "No room left for line plot parameter array" );

    /* Set species params in array */
    for( i = 0, Sp = FirstSpecies(); Sp != NULL; i++, Sp = NextSpecies() )
    {
      strcpy( PlotParams[i].Name, Sp->ShortName );
      PlotParams[i].Symbol = Sp->PlotSym;
      PlotParams[i].LBound = Sp->PlotLowerBound;
      PlotParams[i].UBound = Sp->PlotUpperBound;
    }
    /* Set extra params */
    for( i = 0; i < NExtrPlVr; i++ )
    {
      strcpy( PlotParams[NSpec+i].Name, ExtraPlotVars[i].Name );
      PlotParams[NSpec+i].Symbol = ExtraPlotVars[i].Symbol;
      PlotParams[NSpec+i].LBound = ExtraPlotVars[i].LBound;
```

```
    PlotParams[NSpec+i].UBound = ExtraPlotVars[i].UBound;
  }

  InitLiPl( PageWdth, PlotParams, NSpec + NExtrPlVr );
}
else
{
  char   Dominants[4];     /* String to describe dominant species */
  double *PlotValues;      /* Array of values to pass to LinePlot() */
  double RunningTot = 0.0; /* Biomass accumulator for calc'ing dominants */
  register int j;          /* Iterator */
  register Tree *Tr;       /* Tree iterator */

  /* Send stuff to LinePlot() this year */
  if( Option & OutLiPl )
  {
    /* Get space for array of values */
    if( (PlotValues = (double *) calloc( NSpec + NExtrPlVr, sizeof( double ) ))
        == NULL )
      errz( FATAL, "no room for allocation of line plot values" );

    /* For each species accumulate info for extra vars */
    for( i = 0, Sp = FirstSpecies(); Sp != NULL; i++, Sp = NextSpecies() )
    {
      PlotValues[NSpec+(int) NSprsd] += Sp->NSuprs;
      for( Tr = FirstTree(); Tr != NULL; Tr = NextTree() )
      {
        PlotValues[i] += Biomass();
        PlotValues[NSpec+(int) LfArea] += LeafArea();
        PlotValues[NSpec+(int) BigsTr] = max( PlotValues[NSpec+(int) BigsTr], Tr->DBH );
      }
      PlotValues[NSpec+(int) TotBio] += PlotValues[i];
    }

    /**
      Load up the Dominants string according to species biomass
      dominance.  Dominance is defined as 1, 2 or 3 species accounting
      for 90% or more of total plot biomass.

      First the string (array) Dominants is filled with indices of the
      3 most important species (biomasses stored in PlotValues[i] for
      species i) in decreasing order.
    **/
    Dominants[0] = Dominants[1] = Dominants[2] = -1;
    for( i = 0; i < NSpec; i++ )
      if( Dominants[2] == -1 || PlotValues[i] > PlotValues[Dominants[2]] )
      {
        Dominants[2] = i;
        if( Dominants[1] == -1 || PlotValues[i] > PlotValues[Dominants[1]] )
        {
          j = Dominants[1];
          Dominants[1] = Dominants[2];
          Dominants[2] = j;
          if( Dominants[0] == -1 || PlotValues[i] > PlotValues[Dominants[0]] )
          {
            j = Dominants[0];
            Dominants[0] = Dominants[1];
            Dominants[1] = j;
          }
        }
      }
    /**
      Then j is set to the number of species required to achieve 90% of
      total biomass: if more than 3, then j is left at zero.
    **/
    for( i = j = 0; i < min( 3, NSpec ); i++ )
    {
      RunningTot += PlotValues[Dominants[i]];
```

```
        if( RunningTot >= 0.9 * PlotValues[NSpec+(int) TotBio] )
        {
          j = i + 1;
          break;
        }
      }
      /* Fill the Dominants string with symbols of the j dominants */
      for( i = 0; i < j; i++ )
        Dominants[i] = PlotParams[Dominants[i]].Symbol;
      Dominants[j] = '\0';

      PlotValues[NSpec+(int) TotalN] = TotNTrees();
      LinePlot( Year, PlotValues, Dominants );
      free( (char *) PlotValues );

#if 1
/* Define positions in PlotValues array (spparams-file dependent) */
#define DALR 1
#define DELE 2
#define PAUC 7

    if (Debug)
    {
      /* Print out some selected species and plot info to Debug */
      fprintf(Debug, "%d\t%.01f\t%.01f\t%.01f\t%.01f\t%.21f\t%d\n",
          Year, PlotValues[DALR], PlotValues[DELE], PlotValues[PAUC],
          PlotValues[NSpec+(int) TotBio], PlotValues[NSpec+(int) LfArea],
          TotNTrees());
      fflush(Debug);
    }
#endif

    }
    /* Otherwise simply call the appropriate output funtions */
    if( Option & OutRFir )
      ReportFires();
    if( Option & OutSiLi )
      SiteList();
    if( Option & OutRTim )
      ReportTime();
  }
}

/*****************************************************************************

  printvar.c

  Print one or all of the user-changeable variables and other info

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601
    Australia

  Description:

    PrintVar() accepts a keyword from the user.  If the keyword (case
    insignificant) is "all", all the variables set in SetVars() are
    printed with their current values.  If one of their names is listed,
    only it is printed out.  The output format is suitable for re-input
    to NEWBRIND.

  Global variables, constants and required functions are documented
  in the header files globals.h, vstructs.h and trees.h
```

```
  Local variables are explained where they are declared

*****************************************************************************/
#include <stdio.h>
#include "globals.h"
#include <string.h>
#include "vstructs.h"

void PrintVar()
{
  char Keyword[9];  /* Keyword from user */
  char OutString[16];  /* Output string for numbers */
  enum { NotFound, Found } Status; /* Used in searching for keywords */
  int  NperLine;  /* Number of entries per line */
  int  NtoList;   /* Number to list from each array */

  register int i, c;  /* Iteration counters */

  /* Read keyword:  return on EOF */
  if( ReadString( In, Keyword, "Print" ) == EOF )
    return;

  /* Set number of entries per line */
  NperLine = PageWdth / 20;
  if( NperLine < 4 )
    NperLine = 4;

  /* For keyword "all", print them all */
  if( ! strnicmp( Keyword, "all", 3 ) )
  {
    NtoList = NEnvVars;
    for( i = 0; i < NtoList; i++ )
    {
      sprintf( OutString, "%lf", Envt[i].FVal );
      /* Trim trailing zeros from floating point format */
      for( c = strlen( OutString ) - 1; OutString[c] == '0'; --c )
        OutString[c] = '\0';
      fprintf( Out, "%-8s %-9s", Envt[i].Name, OutString );
      if( (i + 1) % NperLine && (i + 1) < NtoList )
        fputs( " ", Out );
      else
        putc( '\n', Out );
    }

    NtoList = NRYears;
    for( i = 0; i < NtoList; i++ )
    {
      fprintf( Out, "%-8s %-9d", RYears[i].Name, RYears[i].IVal );
      if( (i + 1) % NperLine && (i + 1) < NtoList )
        putc( ' ', Out );
      else
        putc( '\n', Out );
    }

    for( i = 0; i < NRunFlags; i++ )
    {
      if( ! RunFlags[i].FlVal )
        fprintf( Out, "No" );
      fprintf( Out, "%-16s", RunFlags[i].Name );
      if( RunFlags[i].FlVal )
        fprintf( Out, "  " );
      if( (i + 1) % NperLine && (i + 1) < NRunFlags )
        putc( ' ', Out );
      else
        putc( '\n', Out );
    }
```

```
    /**
      I/O unit information is only printed for those not directed to
      their default places.
    **/
    for( i = 0; i < NIOUnits; i++ )
      if( IOUnits[i].CurPtr != IOUnits[i].DfltPtr )
      {
        fprintf( Out, "%s", IOUnits[i].Name );
        if( IOUnits[i].Direction == Reading )
          fprintf( Out, " reading from " );
        else
          fprintf( Out, " writing to " );
        fprintf( Out, "%s\n", IOUnits[i].Filename );
      }

    /* Write out the current parameter file name */
    fprintf( Out, "\nSpecies parameters read from %s\n", ParamFileName );

  } /* all */
  else
  {
    /* Get name of variable and print its value */
    Status = NotFound;
    for( i = 0; i < NEnvVars; i++ )
      if( ! strnicmp( Keyword, Envt[i].Name, 8 ) )
      {
        Status = Found;
        break;
      }
    if( Status == Found )
    {
      sprintf( OutString, "%lf", Envt[i].FVal );
      for( c = strlen( OutString ) - 1; OutString[c] == '0'; --c )
        OutString[c] = '\0';
      fprintf( Out, "%-8s  %s\n", Envt[i].Name, OutString );
    }
    else
    {
      for( i = 0; i < NRYears; i++ )
        if( ! strnicmp( Keyword, RYears[i].Name, 8 ) )
        {
          Status = Found;
          break;
        }
      if( Status == Found )
        fprintf( Out, "%-8s  %d\n", RYears[i].Name, RYears[i].IVal );
      else
        fprintf( Out, "%s not a recognized keyword\n", Keyword );
    }
  }  /* end of "all" test else block */

  /* In batch mode we don't want an extra line */
  if( ! BatchFlag )
    putc( '\n', Out );

}

/*****************************************************************************

  reptime.c

  Report current real world time

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
```

```
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601
    Australia

  Description:

  This function ReportTime() simply reports the current time to the
  output unit TimeOut (see setvars.c).

  Global variables, constants and required functions are documented
  in the header files globals.h, vstructs.h and trees.h

  Local variables are explained where they are declared

******************************************************************************/
#include <stdio.h>
#include "globals.h"
#include <time.h>
#include "vstructs.h"

void ReportTime()
{
#if ! (TC || ZTC)
  unsigned long t;
#else
  time_t t;
#endif
  struct tm *Now;

  time( &t );
  Now = localtime( &t );

  fprintf( TimeOut, "%02d:%02d:%02d  year %d\n", Now->tm_hour, Now->tm_min,
           Now->tm_sec, Year );
  fflush( TimeOut );
}

/******************************************************************************

  savefire.c

  Save fire details to memory and retrieve them

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601
    Australia

  Description:

    SaveFire() saves details of the current fire to a special struct in
      dynamically-allocated memory, linked into a list of them.

    ReportFires() retrieves all of those structs (freeing the memory) and
      writes out their contents to the unit FireOut (define in setvars.c).

  Global variables, constants and required functions are documented
  in the header files globals.h and vstructs.h

  Local variables are explained where they are declared

******************************************************************************/
```

```
#include <stdio.h>
#include "globals.h"
#if ! UNIX
# include <stdlib.h>
#endif
#include "vstructs.h"

typedef struct FireStruct {        /* Structure for each fire */
  double Flame;                    /* Flame height            */
  double ROS;                      /* Rate of Spread          */
  int Year;                        /* Year of fire            */
  int Killed;                      /* No. trees killed        */
  int Total;                       /* Total no. trees on plot */
  struct FireStruct *Next;         /* Pointer to next fire    */
} FireRecord;

static FireRecord *Fire1 = NULL;    /* Pointer to beginning of list */
static FireRecord *LastFire = NULL; /* Pointer to end of it */

void SaveFire( Flame, ROS, Killed, Total )
double Flame, ROS;  /* Flame height (m) and rate of spread (m/hr) */
int Killed, Total;  /* No. killed trees and total no. on plot */
{
  register FireRecord *CurrentFire; /* Single pointer */

  /* Allocate memory */
  if( (CurrentFire = (FireRecord *) malloc( sizeof( FireRecord ) )) == NULL )
    errz( FATAL, "No storage left for fire data" );

  /* Place at the end of the linked list */
  if( LastFire != NULL )
    LastFire->Next = CurrentFire;
  LastFire = CurrentFire;
  CurrentFire->Next = NULL;

  /* Fill up its fields */
  CurrentFire->Year = Year;
  CurrentFire->Flame = Flame;
  CurrentFire->ROS = ROS;
  CurrentFire->Killed = Killed;
  CurrentFire->Total = Total;

  /* Set beginning pointer if necessary */
  if( Fire1 == NULL )
    Fire1 = CurrentFire;
}

void ReportFires()
{
  register FireRecord *Fr, *DoomedFire;

  /* No fires? Then nick off */
  if( Fire1 == NULL )
    return;

  fputs( "\n               F I R E S\n", FireOut );
  fputs( " Year      Flame ht      RoS         Trees\n", FireOut );
  fputs( "              m          m/hr   Killed / Total\n", FireOut );
  fputs( "----------------------------------------------\n", FireOut );

  Fr = Fire1;
  while (Fr != NULL)
  {
    fprintf( FireOut, "%5d    ", Fr->Year);
    if( Fr->Flame == 1000.0 )
      fprintf( FireOut, "   crown " );
    else
      fprintf( FireOut, "%8.1lf ", Fr->Flame );
```

```
    fprintf( FireOut, "%11.2lf %7d %7d\n", Fr->ROS, Fr->Killed, Fr->Total );
    DoomedFire = Fr;
    Fr = Fr->Next;
    free( (void *) DoomedFire );
  }
  fputs( "--------------------------------------------\n", FireOut );

  Fire1 = NULL;
  LastFire = NULL;
}


/****************************************************************************

  siteenvt.c

  SiteEnvt() to read site environment commands from file

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601
    Australia

  Description:

    SiteEnvt calls ReadString() in simpleparse.c to get a filename from
    which to read, then scans that file (independent of the simpleparse.c
    functions).  The file can contain any of the commands in the Envt[]
    array (see setvars.c and vstructs.h).

  Global variables, constants and required functions are documented
  in the header files globals.h, vstructs.h and trees.h

  Local variables are explained where they are declared

****************************************************************************/
#include <stdio.h>
#include "globals.h"
#include <string.h>
#include "vstructs.h"

void SiteEnvt()
{
  char   Keyword[9],       /* Keyword looked up in Envt */
         SitEnvFile[32];   /* Filename from which to read */
  FILE * SiEnFPtr;         /* Pointer to the file */
  int    i;                /* Iteration counter */
  /* Status var while searching for keywords */
  enum   { NotFound, Found, Error, Skip } Status;

  /* Get file name from command line */
  if( ReadString( In, SitEnvFile, "Environment file" ) != 1 )
  {
    errz( WARNING, "Error reading environment file name" );
    return;
  }
  /* Open it for reading; report on failure */
  if( (SiEnFPtr = fopen( SitEnvFile, "r" )) == NULL )
  {
    errz( WARNING, "Open failure on environment file" );
    return;
  }

  /**
```

```
    Get a string.  If it's a keyword in Envt[] collect the new value for
    the variable.  Report errors to stderr.
  **/
  Status = NotFound;
  while( fscanf( SiEnFPtr, "%s", Keyword ) != EOF )
  {
    for( i = 0; i < NEnvVars; i++ )
      if( ! strnicmp( Keyword, Envt[i].Name, 8 ) )
      {
        Status = Found;
        break;
      }
    if( Status == Found )
      if( fscanf( SiEnFPtr, "%lf", &Envt[i].FVal ) != 1 )
        Status = Error;

    if( Status == NotFound )
    {
      fprintf( stderr, "'%s' not an environment variable\n", Keyword );
      Status = Skip;
    }
    else if( Status != Skip )
    {
      if( Status == Error )
        fputs( "Scanning error in input:  variable not set\n", stderr );
      Status = NotFound;
    }
  }

  /* Close file */
  fclose( SiEnFPtr );

  return;
}

/*******************************************************************************

  sitelist.c

  Function SiteList() to list all trees currently on the plot

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601
    Australia

  Description:

    SiteList() lists the current site to the Out unit, by species.
    Species and trees within species are listed according to their order
    in memory (see trees.c).  This means that species are listed in the
    order in which they appear in the species parameter file and trees
    are not sorted by DBH.

    Each species has its full name, no. of trees and biomass listed.
    DBH of each tree is shown.

    SiteList() is sensitive to the value of PageWdth, which can be
    changed by the user.

  Global variables, constants and required functions are documented
  in the header files globals.h, vstructs.h and trees.h

  Local variables are explained where they are declared
```

```
****************************************************************************/
#include <stdio.h>
#include "globals.h"
#include "trees.h"
#include "vstructs.h"

void SiteList()
{
  double   TotBiomass = 0.0;  /* Plot biomass accumulator */
  double   SpBiomass;         /* Species biomass accum'r */
  double   TotLeafArea = 0.0; /* Plot LAI accum'r */
  register int NperLine, i;   /* Number of trees to list per line */
  register Species *Sp;       /* Iteration ... */
  register Tree *Tr;          /* ... counters */

  /**
    Calculate the number of trees to list per line, min. 5 (for 80 col.
    PageWdth).
  **/
  NperLine = (PageWdth - 46) / 6;
  if( NperLine < 5 )
    NperLine = 5;

  /* Heading information */
  fprintf( Out, "\nY E A R   %d\n\n", Year );
  fprintf( Out, "%-25s  %s   %s       %s\n", "Species", "Number",
                "Biomass", "D i a m e t e r s (cm)" );
  /* Then a line */
  for( i = 0; i < (46 + NperLine * 6); i ++ )
    putc( '-', Out );

  /* Loop through species, ignoring those without trees */
  for( Sp = FirstSpecies(); Sp != NULL; Sp = NextSpecies() )
    if( Sp->NTrees > 0 )
    {
      /* Full name and no. of trees */
      fprintf( Out, "\n%-25s   %4d    ", Sp->Name, Sp->NTrees );
      /* Accumulate biomass and LAI */
      for( SpBiomass = 0.0, Tr = FirstTree(); Tr != NULL; Tr = NextTree() )
      {
        SpBiomass += Biomass();
        TotLeafArea += LeafArea();
      }
      TotBiomass += SpBiomass;
      /* Print biomass */
      fprintf( Out, "%7.2lf   ", SpBiomass );
      /* Then print tree DBHs, going to new line when necessary */
      for( Tr = FirstTree(), i = 1; Tr != NULL; Tr = NextTree(), i++ )
      {
        fprintf( Out, " %5.1lf", Tr->DBH );
        if( ! (i % NperLine) && Tr->Next != NULL )
          fprintf( Out, "\n%46s", " " );
      } /* for */
    } /* if */

  /* Another line */
  putc( '\n', Out );
  for( i = 0; i < (46 + NperLine * 6); i ++ )
    putc( '-', Out );

  /* Totals: # trees, biomass, LAI */
  fprintf( Out, "\nTOTALS :  %7d  trees\n", TotNTrees() );
  fprintf( Out, "          %7.2lf  t/ha biomass\n", TotBiomass );
  fprintf( Out, "          %7.3lf  leaf area index\n", TotLeafArea );

  /* Finish with a row of asterisks */
  for( i = 0; i < ((PageWdth - 2) / 3); i++ )
```

```
    fprintf( Out, "*  " );
  fprintf( Out, "\n\n" );

  fflush( Out );
}

/*****************************************************************************

  sitesumm.c

  Site summary functions

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601
    Australia

  Description:

    SiteSummary() is called by main() (newbrind.c) after LeadTime has
    elapsed if YSiteSum (year of site summary after lead time) is not
    negative.  SiteSummary() will produce a site summary and return Stop
    to stop this particular simulation if YSiteSum years has elapsed
    since:

      LeadTime    for FireProb = 0, and
      Last fire   otherwise.

    Thus with fire and YSiteSum = 50, there must be a 50-year fire-free
    period (all of which is after LeadTime) for SiteSummary() to report.

    SiteSummary() calls SiteSummLT() if the LifeTable flag is set,
    otherwise SiteSummOld().

  Global variables, constants and required functions are documented
  in the header files globals.h, vstructs.h and trees.h

  Local variables are explained where they are declared

*****************************************************************************/
#include <stdio.h>
#include "globals.h"
#include "trees.h"
#include "vstructs.h"

#define NCLASS 11
#define CLASSIZ 10.0

#if ! UNIX

int SiteSummOld( void );
int SiteSummLT( void );

#else

int SiteSummOld( );
int SiteSummLT( );

#endif

int SiteSummary()
{
  if( LifeTablFlag )
    return( SiteSummLT() );
```

```
  else
    return( SiteSummOld() );
}

static int SiteSummOld()
{
  /**
    This version will write site summary data to SummOut once for each
    run.  If FireProb > 0.0, then at ysitesum years since fire data are
    written.  Otherwise, at ysitesum years after LeadTime they are
    written.

    After the data are written, Stop is returned to the main program
    to break the annual loop.

    The data written are:
      # & biomass for each species
      number of trees in 11 DBH classes 10.0 cm wide (with >100 cm one class)

    Format:
      All data on one line for each call, separated by tabs
  **/

  double   SpBiom;          /* Species biomass */
  int      RepTime;         /* Year in which to report */
  int      Count[NCLASS];   /* Counters in DBH classes */
  register int i;           /* Iteration counter */
  register Species *Sp;     /* ditto */
  register Tree *Tr;        /* ditto */

  /* Calculate the year in which to report from LeadTime and FireProb */
  if( FireProb > 0.0 )
    if( LastFrYr > LeadTime )
      RepTime = Year - LastFrYr;
    else
      RepTime = 0;
  else
    RepTime = Year - LeadTime;

  /* Report starts here */
  if( RepTime == YSiteSum )
  {
    fprintf( SummOut, "%d", Year );

    for( i = 0; i < NCLASS; i++ )
      Count[i] = 0;

    /**
      Write out species biomass values, and also tally size-class
      information.
    **/
    for( Sp = FirstSpecies(); Sp != NULL; Sp = NextSpecies() )
    {
      SpBiom = 0.0;
      for( Tr = FirstTree(); Tr != NULL; Tr = NextTree() )
      {
        SpBiom += Biomass();
        if( Tr->DBH < 100.0 )
          ++Count[(int) (Tr->DBH / CLASSIZ)];
        else
          ++Count[NCLASS-1];
      }
      fprintf( SummOut, "\t%.1lf", SpBiom );

    }

    /**
      Then write out species numbers and size class information.
```

```
    **/
    for( Sp = FirstSpecies(); Sp != NULL; Sp = NextSpecies() )
      fprintf( SummOut, "\t%d", Sp->NTrees);

    for( i = 0; i < NCLASS; i++ )
      fprintf( SummOut, "\t%d", Count[i] );

    /* Finish the line */
    putc( '\n', SummOut );

    /* Flush output buffer */
    fflush( SummOut );

#if 0
    /* Debugging code to print out G values of surviving trees */
    if (Debug)
    {
      /* Write out year first */
      fprintf(Debug, "%d", Year);

      for(Sp = FirstSpecies(); Sp != NULL; Sp = NextSpecies())
        fprintf(Debug, "\t%.1lf", (Sp->nG == 0) ? 0.0 : Sp->SumG / Sp->nG);

      for(Sp = FirstSpecies(); Sp != NULL; Sp = NextSpecies())
        fprintf(Debug, "\t%d", Sp->nG);

      /* EOL and flush buffer */
      putc('\n', Debug);
      fflush(Debug);
    }
#endif

    return( Stop );  /* Stop is defined in globals.h */

  }  /* End of RepTime == YSiteSum block */

  return( Run );  /* Run is defined in globals.h */
}

/* Write out species' life tables */
static int SiteSummLT()
{
  int sum;
  register int d, f;
  register Species *Sp;

  if( Year - LeadTime != YSiteSum )
    return( Run );

  /**
    Write out mean biomass from all the years since LeadTime.
  **/
  fprintf( SummOut, "%.0lf\n", RunningTotalBiomass / YSiteSum );

  /**
    For each species...
  **/
  for( Sp = FirstSpecies(); Sp != NULL; Sp = NextSpecies() )
  {
    for( sum = 0, d = 0; d < MAXDCLASS; d++ )
      for( f = 0; f < MAXFATE; f++ )
        sum += Sp->LifeTable[d][f];

    if( sum )
    {
      fprintf( SummOut, "**** %s ****\n", Sp->Name );
      for( d = 0; sum && d < MAXDCLASS; d++ )
      {
```

```
        for( f = 0; f < MAXFATE; f++ )
        {
          fprintf( SummOut, "%d\t", Sp->LifeTable[d][f] );
          sum -= Sp->LifeTable[d][f];
        }
        putc( '\n', SummOut );
      }  /* DBH class loop */
    }  /* block for a species with some individuals */
  }  /* species loop */

  return( Stop );
}

/*****************************************************************************

  spparams.c

  Functions to load up species parameters from file

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601
    Australia

  Description:

    Species parameter files have items separated by tabs, with labels on
    the first line.  This format can be easily manipulated by Microsoft
    Excel.

  Global variables, constants and required functions are documented
  in the header files globals.h, vstructs.h and trees.h

  Local variables are explained where they are declared

*****************************************************************************/
#include <stdio.h>
#include "globals.h"
#include <math.h>
#include <string.h>
#include "trees.h"
#include "vstructs.h"

#if GRAPHICS
# include "brgraph.h"

/**
  Colours are defined in graphics.h, but strings with their names are
  used by spparams (note Oz/Brit spellings!).
**/
# define NCOLOURS 16
char *SpeciesColours[NCOLOURS] = {
  "BLACK",
  "BLUE",
  "GREEN",
  "CYAN",
  "RED",
  "MAGENTA",
  "BROWN",
  "LIGHTGREY",
  "DARKGREY",
  "LIGHTBLUE",
  "LIGHTGREEN",
```

```
  "LIGHTCYAN",
  "LIGHTRED",
  "LIGHTMAGENTA",
  "YELLOW",
  "WHITE"
};

/* Shapes also used by spparams */
# define NSHAPES 2
char *SpeciesShapes[NSHAPES] = {
  "Diamond",
  "Square"
};

#endif  /* GRAPHICS */

/**
  Local function ReadParam() to read one param at a time and
  signal fatal error if it is misread.
**/
void ReadParam( fp, fmt, ptr, desc, spname )
FILE * fp;      /* File pointer from which to read */
char * fmt;     /* Format string for fscanf (one item) */
void * ptr;     /* Pointer to item read */
char * desc;    /* Description of item */
char * spname;  /* Species name */
{

  char errstr[64];

  if( fscanf( fp, fmt, ptr ) != 1 )
  {
    sprintf( errstr, "Error reading %s for %s", desc, spname );
    errz( FATAL, errstr ); /* Doesn't return */
  }

} /* ReadParam */

void SpParams()
{
  char Line[26], ColourShape[16];
  double AgeMax, SeedLife;
  FILE *SpPaFPtr;
  register int i;
  Species *Sp;
  unsigned BoolParam;

  /**
    Get parameter file name into global variable, then open file.
    All problems with parameter files are fatal so the user knows
    that the desired parameter values were not loaded.
  **/
  if( ReadString( In, ParamFileName, "Spp. param. file" ) != 1 )
  {
    errz( FATAL, "Error reading parameter file name" );
    return;
  }
  if( (SpPaFPtr = fopen( ParamFileName, "r" )) == NULL )
  {
    errz( FATAL, "Open failure on parameter file" );
    return;
  }

  /* First, get rid of any species & trees */
  KillAll();

  /**
    New param file format:  tab-separated entries, all for one species
```

```
  on a single line.  First line of file has column headings for the
  params (this makes the file easier to read, and can be read by
  Microsoft Excel.
**/

/* Skip first line */
while(getc(SpPaFPtr) != '\n')
  ;

while( fscanf( SpPaFPtr, "%s", Line ) == 1)
{
  Sp = CreateSpecies();

  strcpy( Sp->Name, Line );
  fscanf( SpPaFPtr, "%s", Line );
  strcat( Sp->Name, " " );
  strcat( Sp->Name, Line );

  ReadParam( SpPaFPtr, "%lf", &Sp->DBHMax, "DBHMax", Sp->Name );

  ReadParam( SpPaFPtr, "%lf", &Sp->HeightMax, "HeightMax", Sp->Name );
  Sp->B2 = 2 * (Sp->HeightMax - 137.0) / Sp->DBHMax;
  Sp->B3 = (Sp->HeightMax - 137.0) / (Sp->DBHMax * Sp->DBHMax);

  Sp->HtB = log( Sp->HeightMax - 137.0 );

  ReadParam( SpPaFPtr, "%lf", &AgeMax, "max. age", Sp->Name );
  Sp->SizeIndMort = 1.0 - exp( log( SurvProp ) / AgeMax );

  ReadParam( SpPaFPtr, "%lf", &Sp->GMean, "mean of G", Sp->Name );

  ReadParam( SpPaFPtr, "%lf", &Sp->GSD, "SD of G", Sp->Name );

  ReadParam( SpPaFPtr, "%lf", &Sp->SeedDBH, "min. seed DBH", Sp->Name );

  ReadParam( SpPaFPtr, "%lf", &SeedLife, "seed life span", Sp->Name );
  Sp->SeedSurv = exp( -2.3 / SeedLife );

  ReadParam( SpPaFPtr, "%d", &Sp->MaxFrosts, "min. seed DBH", Sp->Name );

  ReadParam( SpPaFPtr, "%d", &BoolParam, "fire trigger flag", Sp->Name );
  Sp->GermNeedFire = BoolParam;

  ReadParam( SpPaFPtr, "%d", &BoolParam, "seed dispersal flag", Sp->Name );
  Sp->WellDispSeeds = BoolParam;
  Sp->SeedDisp = 0.01 * (1.0 - Sp->SeedSurv);
  if( Sp->WellDispSeeds )
    Sp->SeedDisp *= 4.0;

  ReadParam( SpPaFPtr, "%d", &BoolParam, "enhanced germination flag",
      Sp->Name );
  Sp->GermFireEnhanced = BoolParam;

  ReadParam( SpPaFPtr, "%d", &BoolParam, "fire resistance flag", Sp->Name );
  Sp->FireResistant = BoolParam;

  ReadParam( SpPaFPtr, "%d", &BoolParam, "ligno. seedling flag", Sp->Name );
  Sp->HasLigSdlgs = BoolParam;

  ReadParam( SpPaFPtr, "%lf", &Sp->GrowthForm, "growth form", Sp->Name );

  ReadParam( SpPaFPtr, "%lf", &Sp->Fire2Const, "fire parameter", Sp->Name );

  ReadParam( SpPaFPtr, "%lf", &Sp->Fire2DBH, "fire parameter", Sp->Name );

  ReadParam( SpPaFPtr, "%lf", &Sp->MoistMin, "min. moisture index parameter",
      Sp->Name );
```

```
    ReadParam( SpPaFPtr, "%lf", &Sp->BirthConst, "est't GLM constant term",
        Sp->Name );

    for( i = 0; i < 3; i++ )
      ReadParam( SpPaFPtr, "%lf", &Sp->BirthTemp[i],
          "est't GLM temperature term", Sp->Name );

    for( i = 0; i < 2; i++ )
      ReadParam( SpPaFPtr, "%lf", &Sp->BirthMoist[i],
          "est't GLM moisture term", Sp->Name );

    ReadParam( SpPaFPtr, "%lf", &Sp->BirthTempMoist,
          "est't GLM interaction term", Sp->Name );

    ReadParam( SpPaFPtr, "%lf", &Sp->BirthSediments,
          "est't GLM sediments term", Sp->Name );

    ReadParam( SpPaFPtr, "%lf", &Sp->BirthGranites,
          "est't GLM granites term", Sp->Name );

    ReadParam( SpPaFPtr, "%s", Sp->ShortName, "short name", Sp->Name );

    ReadParam( SpPaFPtr, "%s", Line, "plot symbol", Sp->Name );
    Sp->PlotSym = Line[0];

    ReadParam( SpPaFPtr, "%lf", &Sp->PlotLowerBound, "plot lower bound",
        Sp->Name );

    ReadParam( SpPaFPtr, "%lf", &Sp->PlotUpperBound, "plot upper bound",
        Sp->Name );

    /**
      Read colour even for non-graphics versions so the
      parameter files can be the same (i.e., ignore what is
      read).
    **/
    ReadParam( SpPaFPtr, "%s", ColourShape, "graphic tree colour", Sp->Name );

#if GRAPHICS

    for( i = 0; i < NCOLOURS; ++i )
      if( ! strnicmp( ColourShape, SpeciesColours[i], 12 ) )
        break;
    if( i == NCOLOURS )
      errz( FATAL, "Error:  invalid tree colour specified" );
    else
      Sp->GraphColour = i;

#endif

    /* Ditto for shape */
    ReadParam( SpPaFPtr, "%s", ColourShape, "graphic tree shape", Sp->Name );

#if GRAPHICS

    for( i = 0; i < NSHAPES; ++i )
      if( ! strnicmp( ColourShape, SpeciesShapes[i], 12 ) )
        break;
    if( i == NSHAPES )
      errz( FATAL, "Error:  invalid tree shape specified" );
    else
      Sp->GraphShape = i;

#endif

  }
  fclose( SpPaFPtr );
  InitLifeTable();
```

```
  return;
}


/******************************************************************************

  strnicmp.c

  Case-insensitive, limited-length string comparison function

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601
    Australia

  Description:

    Some compiler libraries don't have this function, so here it is.

******************************************************************************/
#include <stdio.h>
#include <ctype.h>

int strnicmp( str1, str2, maxleng )  /* String comparison up to length */
char *str1, *str2;                   /* maxleng:  case insensitive     */
int maxleng;
{
  for( ; maxleng--; ++str1, ++str2 )
    if( (islower( *str1 ) ? toupper( *str1 ) : *str1) !=
        (islower( *str2 ) ? toupper( *str2 ) : *str2) )
      return( (int) (*str1 - *str2) );
    else if( ! *str1 )
      break;

  return( 0 );
}

/******************************************************************************

  strtsite.c

  StrtSite() function to load a starting site description from file

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601
    Australia

  Description:

    StrtSite() gets a file name from the user.

    Basic file format is species short name followed by number of trees
    then DsBH of those trees, all separated by whitespace.  In this case
    the G values of the individual trees calculated by CreateTree()
    (trees.c) are kept.  Example:
```

```
      E.dalr 4 3 3 3 3
      E.dele 4 3 3 3 3
      E.pauc 4 3 3 3 3

    If the file contains the keyword "ReadG", then each tree's G value is
    read before its DBH.  Example:

      E.dalr 2 56 3.5 70 23.5
      E.dele 2 164 72.5 140 102.6

  Global variables, constants and required functions are documented
  in the header files globals.h, vstructs.h and trees.h

  Local variables are explained where they are declared

*****************************************************************************/
#include <stdio.h>
#include "globals.h"
#include "trees.h"
#include "vstructs.h"

#define OFF 0
#define ON  1

void StrtSite()
{
  char     Keyword[7],     /* Keyword for finding sp. short name */
           StrSitFile[32]; /* Filename */
  FILE     *StSiFPtr;      /* File stream pointer */
  double   DBH,            /* A DBH */
           G,              /* A value of G read from file */
           TotBiom = 0.0;  /* Plot biomass accumulator */
  int      NoMatchName,    /* Flag set if a name is not recognised */
           NoMatchRead,    /* Flag set if "ReadG" is not recognised */
           NoMatFlag = OFF,/* Flag to stop reporting of every wrong name */
           ReadG = OFF,    /* Flag for reading G values */
           ntrees;         /* Counter of trees in a sp */
  Species *Sp;             /* Iteration counter */
  Tree    *Tr;             /* ditto */

  /**
    Remove all existing trees and zero GotMatureTrees flags
  **/
  for( Sp = FirstSpecies(); Sp != NULL; Sp = NextSpecies() )
  {
    for( Tr = FirstTree(); Tr != NULL; Tr = KillTree() )
      ;
    Sp->GotMatureTrees = 0;
  }

  /* We must have species to be able to create a site */
  if( FirstSpecies() == NULL )
  {
    errz( WARNING, "You have no species" );
    return;
  }

  /* Get file name */
  if( ReadString( In, StrSitFile, "Site file" ) != 1 )
  {
    errz( WARNING, "Error reading starting site filename" );
    return;
  }
  if( (StSiFPtr = fopen( StrSitFile, "r" )) == NULL )
  {
    errz( WARNING, "Open failure on starting site file" );
    return;
  }
```

```
/* Scan for a species short name or the keyword "ReadG" */
while( fscanf( StSiFPtr, "%s", Keyword ) != EOF )
{
  for( Sp = FirstSpecies(); Sp != NULL &&
       (NoMatchName = strnicmp( Keyword, Sp->ShortName, 6 )) != 0 &&
       (NoMatchRead = strnicmp( Keyword, "ReadG", 6 )) != 0;
       Sp = NextSpecies() )
    ;

  if( ! NoMatchName )     /* We have a species */
  {
    if( fscanf( StSiFPtr, "%d", &ntrees ) != 1 )
    {
      fputs( "Error reading no. trees:  ", Out );
      fputs( "skipping to next recognized keyword\n", Out );
      NoMatFlag = ON;
    }
    else
    {
      Sp->Biomass = 0.0;

      while( ntrees-- )
      {
        Tr = CreateTree();
        /**
          Read G value for each tree from file.  If value can't be read
          it is ignored and the value calc'd in CreateTree() used.
        **/
        if( ReadG )
          if( fscanf( StSiFPtr, "%lf", &G ) != 1 )
            fputs( "Error reading tree G:  value ignored\n", Out );
          else
            Tr->G = G;

        if( fscanf( StSiFPtr, "%lf", &DBH ) != 1 )
        {
          fputs( "Error reading tree DBH:  set to 3.0\n", Out );
          DBH = 3.0;
        }

        /* Set up tree stuff */
        AddDBH( DBH );
        Sp->Biomass += Biomass();
        TotBiom += Biomass();
        if( DBH >= Sp->SeedDBH )
          Sp->GotMatureTrees = 1;
      }
    }
    NoMatFlag = OFF;
  }
  else if( ! NoMatchRead )
  {
    ReadG = ON;
    NoMatFlag = OFF;
  }
  else if( ! NoMatFlag )
  {
    fprintf( Out, "'%s' not a recognized keyword\n", Keyword );
    NoMatFlag = ON;
  }
}  /* End of Keyword reading while loop */

/**
  Set up Sp->Seeds as prop'n of total biomass if any trees present, 0.0
  otherwise.
**/
for( Sp = FirstSpecies(); Sp != NULL; Sp = NextSpecies() )
```

```
    if( Sp->Tree1 != NULL )
      Sp->Seeds = Sp->Biomass / TotBiom;
    else
      Sp->Seeds = 0.0;

  fclose( StSiFPtr );
  return;
}

/*****************************************************************************

  brgraph.c

  Graphics pictures of trees using Borland BGI graphics routines.

  Written by:

    Mike Strasser
    Ecosystem Dynamics Group
    Research School of Biological Sciences
    The Australian National University
    P.O. Box 475
    Canberra A.C.T.  2601
    Australia

  Description:

    Functions to draw the forest plot, draw individual trees and place
    text on the graphics screen in a specified colour.

  Functions:

    InitGraphics() detects the graphics board present and loads the
      appropriate graphics driver, then stores info about X & Y
      resolution and scaling for drawing trees.

    DrawPlot() draws the plot frame and height scale, and sets the mode
      for tree drawing to XOR.

    ClearGraphics() changes back to text screen.

    GraphTree() puts a single tree on the screen.

    GraphAllTrees() puts all trees on the screen.

    Picture() calls DrawPlot and GraphAllTrees to show the current plot
      state as a single picture.

    gcprintf() writes a string with the specified colour to the specified
      place on the screen.

    MinXPos() returns the minimum x position on the screen, and
    RangeXPos() returns the range of x positions.

*****************************************************************************/
#include <stdio.h>
#include "globals.h"

#if GRAPHICS

#include <stdarg.h>
#include <graphics.h>
#include "brgraph.h"
#include "trees.h"
#include "vstructs.h"

#define MAXHEIGHT 4000.0  /* Height of tree plot area in cm */

/**
```

```
  Current screen mode: globally-known for speed of access
**/
enum { Text, Graph } CurrentState = Text;

/**
  Locally-known static variables
**/
static int GrDriver = DETECT;   /* Graphics driver instruction */
static int GrMode;              /* Current mode */
static int ErrCode;             /* Current error code */
static char ErrStr[161];        /* Error message string */
static int MaxX, MaxY;          /* Screen resolution */
static int GrHeight;            /* Height of tree plot area in pixels */
static double YScale;           /* Vertical scaling factor for trees */
static int GrMinX, GrMaxX;      /* Min and max X-pixels for tree trunk location */

/**
  Local routine to check-out graphics info.  It needs to go into and out
  of graphics mode to get info about X- and Y-resolution.
**/
static void InitGraphics( void )
{
  /* Detect graphics driver present */
  initgraph( &GrDriver, &GrMode, GrDriverPath );
  ErrCode = graphresult();
  if( ErrCode != grOk )
  {
    sprintf( ErrStr, "Graphics system error:  %s\nNo graphics available",
             grapherrormsg( ErrCode ) );
    errz( WARNING, ErrStr );
    /* Reset NEWBRIND Graphics flag */
    Graphics = 0;
    return;
  }

  MaxX = getmaxx();
  MaxY = getmaxy();

  /**
    GrHeight is the height in pixels taken up by MAXHEIGHT (which is in cm).
    It is defined so that there is room for 3 - 4 lines of text below it,
    and to be a multiple of 4.
  **/
  settextstyle( DEFAULT_FONT, HORIZ_DIR, 1 );
  GrHeight = MaxY - 4 * (textheight( "H" ) + 2);
  GrHeight += GrHeight % 4;

  /**
    YScale is multiplied by a tree's height in cm to give its height in
    pixels.
  **/
  YScale = GrHeight / MAXHEIGHT;

  /**
    GrXMin is the lowest X pixel value on which trees are plotted (their trunk
    locations), and GrXMax the highest.  They are 20 pixels in from the left
    and right sides of the screen, respectively.
  **/
  GrMinX = 40;
  GrMaxX = MaxX - 20;

  closegraph();
}

/**
  Draw up the plot -- initialize plotting.  Set CurrentState to Graph.
**/
void DrawPlot( void )
```

```
{
  double        h;  /* Height... */
  register int  i;  /* ... and pixel temp variables used in drawing ticks */

  /* Call InitGraphics if necessary */
  if( GrDriver == DETECT )
    InitGraphics();
  initgraph( &GrDriver, &GrMode, GrDriverPath );
  CurrentState = Graph;

  /* Draw a bright green base line as a thick line */
  setlinestyle( SOLID_LINE, 0, THICK_WIDTH );
  setcolor( LIGHTGREEN );
  line( GrMinX, GrHeight + 1, GrMaxX, GrHeight + 1 );

  /* Draw a vertical axis and scale ticks for each 10 m height */
  setlinestyle( SOLID_LINE, 0, NORM_WIDTH );
  setcolor( WHITE );
  line( GrMinX - 1, GrHeight, GrMinX - 1, 0 );

  for( h = 0.0; h < MAXHEIGHT + 10.0; h += 1000.0 )
  {
    i = GrHeight - h * YScale;
    line( GrMinX - 1, i, GrMinX - 8, i );
  }

  /**
    Write vertical legend and tick values.
  **/
  settextstyle( DEFAULT_FONT, VERT_DIR, 1 );
  ErrCode = graphresult();
  if( ErrCode != grOk )
  {
    closegraph();
    sprintf( ErrStr, "Graphics system error: %s\nNo graphics available",
             grapherrormsg( ErrCode ) );
    errz( WARNING, ErrStr );
    Graphics = 0;
    return;
  }
  settextjustify( LEFT_TEXT, CENTER_TEXT );
  outtextxy( 10, GrHeight / 2, "Height (m)" );

  settextstyle( DEFAULT_FONT, HORIZ_DIR, 1 );
  settextjustify( RIGHT_TEXT, CENTER_TEXT );
  for( h = 0.0; h < MAXHEIGHT + 10.0; h += 1000.0 )
  {
    sprintf( ErrStr, "%.0lf", h / 100.0 );
    outtextxy( GrMinX - 9, GrHeight - h * YScale, ErrStr );
  }

  /* We'll use XORing for drawing trees */
  setwritemode( 1 );
}

/**
  Clear up graphics and return to text screen.
**/
void ClearGraphics( void )
{
  closegraph();
  CurrentState = Text;
}

/**
  Show a tree on the screen at the specified position.
**/
void GraphTree( int XPos, double Height, int Colour, int Shape )
```

```
{
  int           canopycoords[10];  /* Array for drawing canopy shape */
  register int  canopyheight,      /* Pixel value */
                halfcanopy,        /* Ditto */
                iHeight;           /* Ditto */

  iHeight = Height * YScale;
  canopyheight = iHeight / 3;
  halfcanopy = canopyheight / 2;

  setcolor( Colour );
  setlinestyle( SOLID_LINE, 0, NORM_WIDTH );
  line( XPos, GrHeight, XPos, GrHeight - iHeight + canopyheight  );
  if( Shape == Diamond )
  {
    canopycoords[0] = canopycoords[4] = canopycoords[8] = XPos;
    canopycoords[2] = XPos + halfcanopy;
    canopycoords[6] = XPos - halfcanopy;
    canopycoords[1] = canopycoords[9] = GrHeight - iHeight;
    canopycoords[3] = canopycoords[7] = GrHeight - iHeight + halfcanopy;
    canopycoords[5] = GrHeight - iHeight + canopyheight;
    drawpoly( 5, canopycoords );
  }
  else
    rectangle( XPos - halfcanopy, GrHeight - iHeight,
               XPos + halfcanopy, GrHeight - iHeight + canopyheight );
}

/**
  Show all trees.
**/
void GraphAllTrees( void )
{
  register Species *Sp;
  register Tree *Tr;

  for( Sp = FirstSpecies(); Sp != NULL; Sp = NextSpecies() )
    for( Tr = FirstTree(); Tr != NULL; Tr = NextTree() )
    {
      Tr->GraphHeight = Tr->Height;
      GraphTree( Tr->XPos, Tr->GraphHeight, Sp->GraphColour, Sp->GraphShape );
    }
}

/**
  Draw a picture of the plot.
**/
void Picture( void )
{
  DrawPlot();
  GraphAllTrees();
  gcprintf( 100, MaxY, WHITE, "Press any key to continue..." );
  getch();
  ClearGraphics();
}

/**
  gcprintf

  Graphics mode printf with colour specified.  After gprintf in BGIDEMO.C,
  but X and Y coords passed by value and not altered.
**/
int gcprintf( int x, int y, int colour, char * fmt, ... )
{
  va_list argptr;  /* Argument list pointer */
  char str[140];   /* Buffer for string */
  int counter;     /* Result of sprintf() for return */
```