United States
Department
of Agriculture

Forest Service

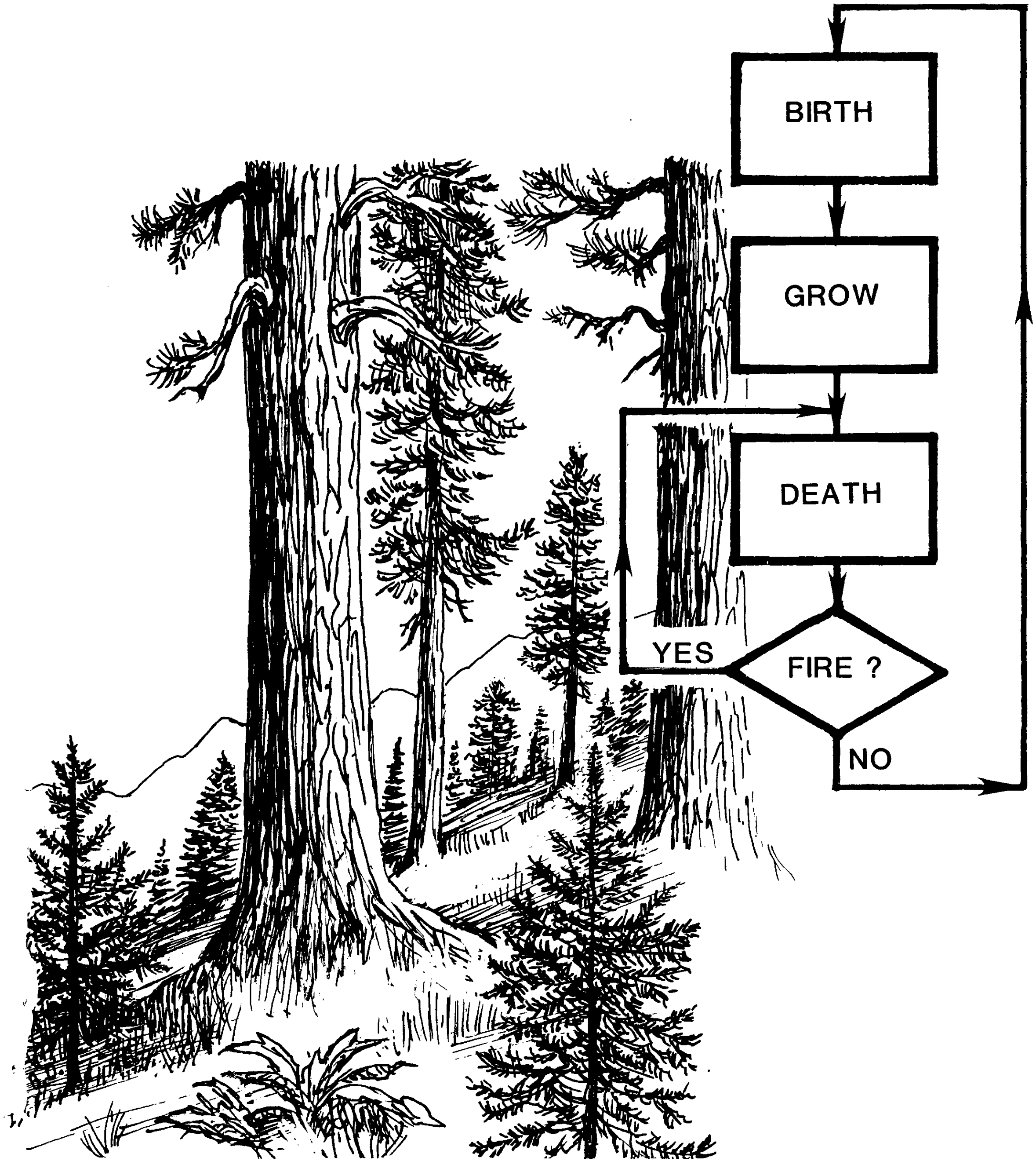
**Intermountain
Research Station**

General Technical
Report INT-266

September 1989

FIRESUM—An Ecological Process Model for Fire Succession in Western Conifer Forests

**Robert E. Keane
Stephen F. Arno
James K. Brown**



THE AUTHORS

ROBERT E. KEANE is a quantitative ecologist working under cooperative agreement with the Intermountain Research Station in Missoula, MT. He received his bachelor's degree from University of Maine in 1978 in forest engineering and his master's from University of Montana in 1985 in forestry. He has been working for the past 5 years developing ecological data analysis software for the Northern Region's Ecosystem Management Program and conducting computer modeling research with the Intermountain Research Station.

STEPHEN F. ARNO is a research forester in the Prescribed Fire and Fire Effects research work unit at the Intermountain Fire Sciences Laboratory, Intermountain Research Station, in Missoula, MT. He received a B.S. degree in forestry from Washington State University and master's and Ph.D. degrees from the University of Montana. He has studied various aspects of forest ecology since 1963.

JAMES K. BROWN is the leader of the Prescribed Fire and Fire Effects research work unit at the Intermountain Fire Sciences Laboratory in Missoula, MT. He received his bachelor's degree from the University of Minnesota in 1960, his master's degree from Yale University in 1961, and his Ph.D. degree from the University of Michigan in 1968, all in forestry. His past reseach has focused on fuel consumption, fire ecology of western forests, and development of prescribed burning guidelines.

RESEARCH SUMMARY

A successional process model has been developed to simulate long-term stand dynamics in forests of the Northern Rockies. The model can be used to evaluate fire effects differences for various fire regimes, including prescribed burning at different intervals, complete fire exclusion (fire suppression), and pre-1900 fire frequencies. The model, FIRESUM (a **FIRE SU**ccession **M**odel), simulates tree establishment, growth, and mortality, along with live and dead fuel accumulation, fire behavior, and fuel reduction on a 400-square-meter plot. The following influences on tree establishment and growth are included in the model: growing season warmth, water stress, light tolerance, and site quality. The model predicts basal area by species, duff and fuel accumulation, and fire intensities. All model algorithms are discussed, and corresponding parameters for several tree species are presented. The model is continually being tested and verified. Recent test results show FIRESUM underpredicts basal area by an average of 10 to 20 percent. A sensitivity analysis of FIRESUM showed that parameters associated with the growth algorithm are most critical. The model was designed so that it could be applied to different forest types with minimal modification of the computer code.

ACKNOWLEDGMENTS

We thank the following individuals who participated in workshops that provided critical analysis during model development: James R. Habeck, Steven W. Running, Ronald H. Wakimoto, and Hans R. Zuuring of the University of Montana (Missoula); Patricia L. Andrews, Jim D. Chew, William H. Frandsen, Arnold I. Finklin, Roger G. Hungerford, David Pierce, Elizabeth D. Reinhardt, Kevin C. Ryan, and Raymond C. Shearer of the Forest Service, U.S. Department of Agriculture, in Missoula.

CONTENTS

Page
Introduction 1
The Model 1
  Model Description 1
  Tree Growth 4
  Tree Regeneration 7
  Tree Mortality 8
  Fuel Accumulation 10
  Fire Characteristics 12
  Fuel Consumption 13
Model Output 13
Model Testing and Analysis 14
  Validation and Verification 14
  Sensitivity Analysis 15
Summary and Conclusions 16
References 17
Appendix A: Program listing for FIRESUM 21
Appendix B: Example of the TREE.DAT file 73
Appendix C: Example of the SITE.DAT file 75
Appendix D: Example of the CONTRL.DAT file 76

Intermountain Research Station
324 25th Street
Ogden, UT 84401

FIRESUM—An Ecological Process Model for Fire Succession in Western Conifer Forests

Robert E. Keane
Stephen F. Arno
James K. Brown

INTRODUCTION

The long-term effects on forest composition and structure of different fire management alternatives, such as complete suppression of all fires and prescribed fires of varying intervals and prescriptions, are often difficult to quantify. Although many researchers have studied successional communities arising after fire (Arno and others 1985; Kessell and Potter 1980; Steele and Geier-Hayes 1987; Stickney 1980; Means; 1981), investigations of the effects of successive fires—that is a "fire regime"—on vegetation are rare. The long time periods involved greatly complicate quantification of effects of successive fires based on field evidence. Computerized simulation models, however, offer an alternative means of comparing long-term effects of different fire regimes on forest vegetation. An additional benefit of developing such models is detection of areas where knowledge is deficient and future research is critically needed.

We developed a computer model, called FIRESUM (a **FIRE SU**ccession **M**odel), to simulate the effect of different fire regimes on tree composition, stand structure, and fuel loading in forests of the inland portion of the northwestern United States. Comparison of long-term fire effects predictions under different fire regimes could prove useful for developing fire management prescriptions to meet resource management objectives.

FIRESUM was created by extensively modifying the process model SILVA (Kercher and Axelrod 1981), which simulates forest succession involving fire in mixed conifer forests of the California Sierra Nevada. Parameters and algorithms within SILVA were revised, deleted, or added to reflect current knowledge of ecologic processes inherent in various types of forests. Currently, FIRESUM can be applied to ponderosa pine/Douglas-fir and whitebark pine/ subalpine fir forests of the Inland Northwest and the Northern Rocky Mountains.

The purpose of this paper is to describe algorithms and routines used in FIRESUM along with related modeling assumptions. The parameters used to quantify each algorithm are also discussed.

THE MODEL

Model Description

FIRESUM is a deterministic model containing stochastic properties. Tree growth, woody fuel accumulation, and litterfall are simulated deterministically, whereas tree establishment and mortality are stochastic algorithms. The model simulates all processes on an individual tree level in a 400-square-meter area called the simulation plot. Because the particular combination of stochastic events occurring within a given FIRESUM simulation represent only one case among the set of many possible simulation outcomes, the model repeats simulations many times to obtain an average of simulated results.

FIRESUM is a gap-replacement model (Shugart and West 1980) following the approach used for JABOWA (Botkin and others 1972) in which individual trees are grown deterministically using an annual time step, difference equation. Tree growth is affected by several site factors, including available light, water stress, and growing season warmth. Tree establishment and mortality are modeled stochastically using Monte Carlo techniques. Fuel loadings are calculated yearly. Fires are introduced at various intervals, and effects of each fire are simulated by reduction of litter, duff, and down woody fuels; and by tree mortality and postfire tree regeneration and growth.

FIRESUM was programmed in the FORTRAN 77 language and contains over 2000 lines of code, with 43 subroutines and a main driver (appendix A). A generalized flow chart for FIRESUM execution is presented in figure 1. FIRESUM execution starts with tree and site parameters read into the program from external data files (TREE.DAT and SITE.DAT as shown in appendixes B and C) in subroutines TREE and SITE.DAT. External files allow efficient modification of parameters and facilitate the execution of simultaneous runs. The tree parameter file (appendix B) consists of numbers describing each tree species in terms of the model's algorithms. For example, the maximum height of each tree species used in growth algorithm of FIRESUM ($\mathbf{H}_m$ in appendix B) is represented in the tree parameter file. The site file (appendix C) contains parameters that describe the simulation site. Initial tree data for a sample plot are read from data file CONTRL.DAT (appendix D) into subroutine CONTRL and then these input trees are distributed on the plot in DIST. These data represent the simulation stand at the start of simulation. Parameters used to summarize site conditions are read from subroutine SITE.DAT and used to compute growth reduction factors in CALC and SITE. Frequency of cone crops and fire years are computed in CYCLES and RINGS, repectively. Establishment of new trees is done in BIRTH, trees are then grown in subroutine GROW and subject to mortality in KILL, thereby completing a normal tree life cycle.

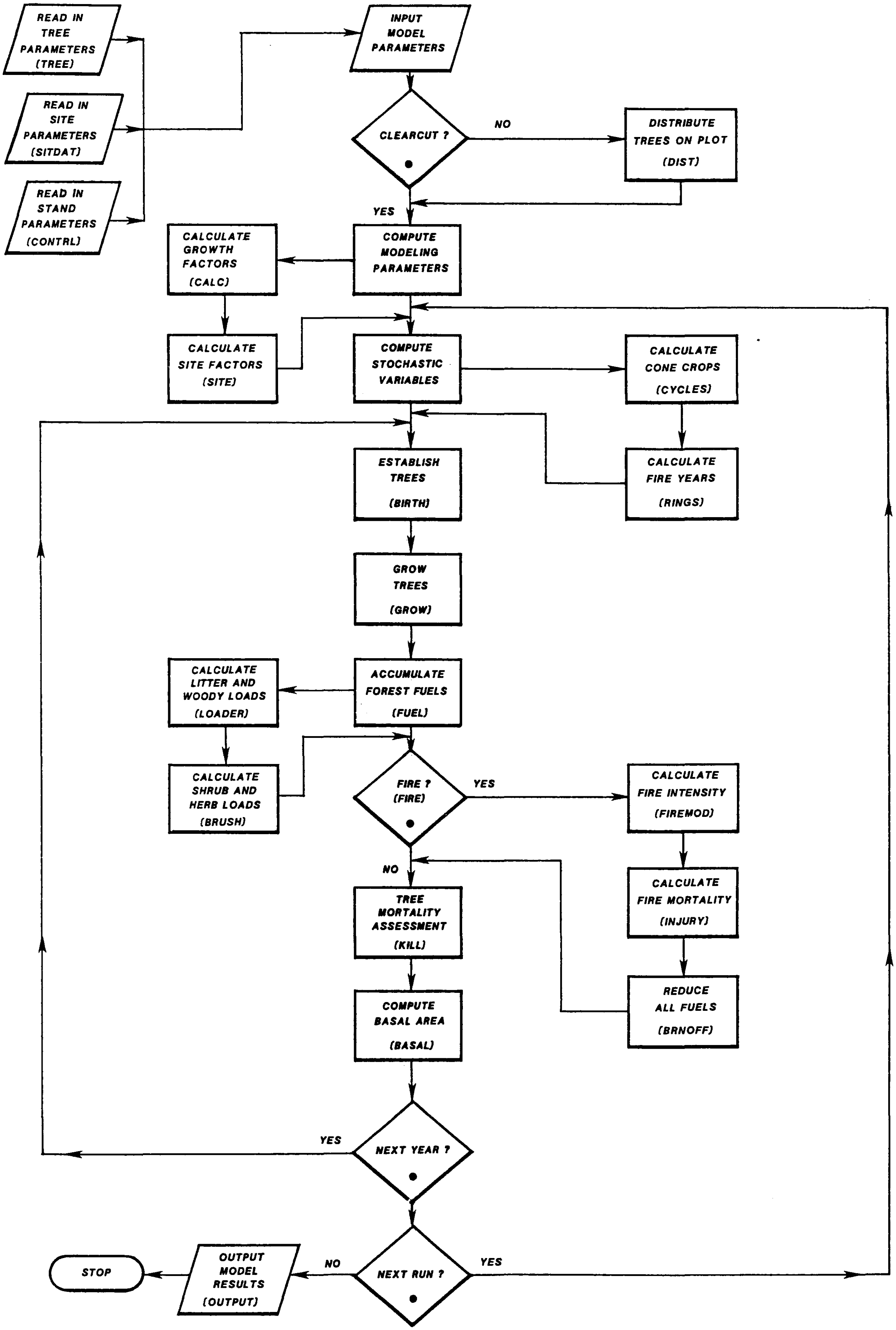


Figure 1—An instructional guide to program logic for the simulation process model FIRESUM. Program subroutines are noted in parentheses.

Fuel loadings are annually estimated in FUEL, LOADER, and BRUSH, and are passed to subroutine FIRE when a fire is initiated. Fire intensity is calculated in FIREMOD from these fuel loading predictions. Subsequent tree mortality from fire is estimated in INJURY using function RISK. Fuel reduction is performed in subroutine BRNOFF and the new loadings are passed back to FUEL. BASAL stores a running average annual basal area by species, which is then passed to subroutine OUTPUT at program termination. OUTPUT prints final results to external files.

Several subroutines not shown in figure 1 are also used in model execution. Subroutine SNAG estimates woody fuel contributed by recently dead trees and adds that amount to the fuel bed. FOLIAGE computes the leaf area of each tree on the simulation plot. Subroutines BEETLE and RUST are used to compute mortality caused by the mountain pine beetle and white pine blister rust. Crown fires are modeled in subroutine CROWN, which predicts when a ground or surface fire becomes hot enough to ignite tree crowns. This submodel is in the developmental stage and needs additional testing before implementation into FIRESUM. Subroutine RANDX is the random number generator. The growth reduction factor for water stress is computed in WRSTRS. The degree of shading based on leaf area is computed in SHADE, and the flame length is computed in FLTEMP.

The following are detailed descriptions of major simulation algorithms in FIRESUM. Values for parameters in these algorithms are shown in table 1.

Table 1—A summary table of parameter values for all species currently implemented in FIRESUM

Parameter symbol[1] (units)	Tree species[2] PIPO	ABGR	PSME	PICO	LAOC	ABLA	PIEN	PIAL	PICO	LALY
Hm (cm)	6,562.5	5,333.7	5,715.0	4,115.0	6,857.5	4,175.5	5,456.0	3,657.0	4,115.0	3,048.0
Dm (cm)	250.5	139.4	208.8	110.0	250.0	126.7	234.4	182.0	110.0	168.0
AGEMAX (years)	450.0	275.0	350.0	220.0	450.0	180.0	320.0	1,000.0	350.0	800.0
DMIN (deg-days)	2,249.9	2,496.6	1,810.4	1,215.3	1,817.4	801.8	801.4	800.0	1,500.0	800.0
DOPT (deg-days)	4,010.0	4,200.0	4,200.0	4,200.0	4,200.0	3,800.0	3,800.0	3,000.0	3,000.0	3,000.0
DMAX (deg-days)	8,608.0	7,194.0	7,194.0	7,194.0	7,194.0	6,200.0	6,200.0	5,200.0	6,500.0	5,200.0
Shade tolerance[3]	I	T	M	I	I	T	M	M	I	I
SV (cm^2/cm^2)	57.6	72.9	69.1	64.7	184.0	70.0	54.2	57.6	64.7	184.0
PLA (m/m)	3.54	2.04	2.85	3.54	3.54	2.04	2.04	3.54	3.54	3.54
WSO (proportion)	.25	.47	.32	.38	.38	.65	.65	.33	.40	.75
Pc (probability)	.395	.333	.446	.318	.438	.333	.167	N/A	.318	.368
hc (years)	2.0	2.0	1.0	2.0	2.0	2.0	3.0	1.0	2.0	1.0
BC (proportion)	.070	.033	.065	.014	.069	.015	.022	.015	.014	.031
DKF (proportion)	.0575	.0339	.0339	.0460	.1310	.0339	.0339	.057	.044	.201
DKL (proportion)	.1116	.0667	.1167	.1186	.2000	.0667	.0667	.112	.112	.200
LTD (proportion)	.5500	.6500	.6500	.6600	.8500	.6500	.6500	.550	.660	.650
DKD (proportion)	.2280	.2210	.2210	.2210	.2210	.2210	.2210	.221	.221	.321
AINC (centimeters)	.012	.005	.007	.015	.016	.008	.008	.006	.016	.007
Lc (percent)	40.0	80.0	80.0	40.0	40.0	80.0	80.0	50.0	40.0	45.0
NYR (years)	4.0	7.0	5.0	3.0	1.0	7.0	6.0	7.0	3.0	1.0

[1] Hm = Maximum attainable height
Dm = Maximum attainable diameter
Agemax = Maximum attainable age
DMIN = Minimum number degree-days
DOPT = Optimum number of degree-days
DMAX = Maximum number of degree-days
Shade tolerance = Shade tolerance class
SV = Surface to volume ratio of foliage
PLA = Projected leaf area conversion factor
WSO = Minimum AET:PET ratio
Pc = Probability of good cone crop
hc = Years blocked after good cone crop
BC = Bark thickness conversion factor
DKF = Decomposition loss from needlefall
DKL = Decomposition loss from litter
LTD = Decomposition loss from litter to duff
DKD = Decomposition loss from duff
AWC = Minimum diameter growth for mortality
Lc = Live crown ratio
NYR = Years needles remain on tree

[2] Species codes are: PIPO = ponderosa pine, ABGR = grand fir, PSME = Douglas-fir, LAOC = western larch, ABLA = subalpine fir, PIEN = Engelmann spruce, PIAL = whitebark pine, PICO = lodgepole pine, LALY = subalpine larch.

[3] Shade tolerance categories are I-shade intolerant, M-moderately shade tolerant, and T-shade tolerant.

Tree Growth (Subroutine GROW)

Growth is modeled by an annual increase in tree diameter measured at breast height (d.b.h.) [1.37 meters above ground line] (Botkin and others 1972). Diameter increment growth (*dD/dt*) is calculated from the time step equation:

$$\frac{dD}{dt} = \frac{G\,D\,[1-(DH)/(D_m H_m)]}{274 + 3b_2D - 4b_3D^2}\,[rAL\; rN\; rW\; rDEGD] \quad (1)$$

where D is the diameter (d.b.h. in centimeters) and H is the height of the tree (centimeters), D_m and H_m are maximum attainable d.b.h. and height (centimeters) for the tree species in the Northern Rocky Mountain region. Values for D_m and H_m (table 1) are taken from Patterson and others (1985), Fowells (1965), Pando (1973), Pfister and others (1977), Hunt (1986), and other studies of old-growth forests. Tree height (H) is computed from:

$$H = 137 + b_2D - b_3D^2 \quad (2)$$

where b_2 and b_3 are species-dependent constants. Constants G, b_2, and b_3 are estimated using equations 3 and 5 in Botkin and others (1972), which have D_m, H_m, and maximum attainable age (AGEMAX in years) as independent variables.

The remaining variables in the equation are growth reduction factors (values between 0.0 and 1.2) that represent the total effect of surrounding environment on tree growth. These factors are modeled as tree growth response to available light (*rAL*), nutrient supply (*rN*), water relations (*rW*), and temperature regime (*rDEGD*). Optimal growth is only possible when all factors equal 1.0.

Available light (*AL*) for an individual tree is calculated according to Beer's Law (Kercher and Axelrod 1982) using the equation:

$$AL = AL_o e^{(-k_j\ \Sigma LAI)} \quad (3)$$

where ΣLAI is the sum of leaf area indexes for all trees taller than the tree under consideration and AL_o is available light at full sunlight (standardized to 1.0). Variable k_j is the extinction coefficient per meter for canopy type j. Because forest canopy characteristics differ by tree composition, that is forest community, it was necessary to stratify extinction coefficient (k) (and many other simulation parameters mentioned later in this paper) by a classification of fire groups (Davis and others 1980) synthesized from the Montana habitat types of Pfister and others (1977). In their classification, habitat types were grouped into similar categories based on vegetation composition, tree ecology and fire histories (table 2). Canopy extinction coefficients by fire group are presented in table 3.

Because utilization of available light by a tree depends on degree of shade tolerance for that species, light response equations were stratified by shade tolerance class (shade intolerant, moderately shade tolerant, and shade tolerant as shown in table 1). These equations, from Botkin and others (1972), are

$$\text{Shade tolerant:}\quad rAL = 1 - e^{[-4.64\,(AL-0.05)]} \quad (4)$$

$$\text{Shade intolerant:}\quad rAL = 2.24\,[1 - e^{[-1.136\,(AL-0.08)]}] \quad (5)$$

where *rAL* is a dimensionless number between zero and 1.0 (1.2 for shade intolerant species), and *AL* expresses available light (also dimensionless). Shade-tolerant species are able to attain higher growth rates in heavily shaded conditions (fig. 2). But light saturation for tolerant species occurs at a much lower level of photosynthetic activity than for the shade intolerants. Although three tolerance classes are recognized in FIRESUM, the tolerant equation (4) includes species that are tolerant and moderately tolerant of shade.

Leaf area was difficult to calculate due to the absence of leaf area equations for Inland Northwest tree species. In FIRESUM we estimated leaf area (*LA* in square centimeters) from:

$$LA = \frac{[(CW{*}PFOL)/CD] * SV_i}{PLA_i} \quad (6)$$

where *CW* is crown weight in grams, *PFOL* is proportion of crown that is foliar weight, *CD* is needle density in grams per cubic centimeter (assumed to be 0.5 for all species based on the authors' unpublished data), SV_i is

Table 2—Fire groups implemented in FIRESUM. Tree species prevalent in the ponderosa pine/Douglas-fir forests are capable of attaining dominance in any of these fire groups

Number		Fire group name[1]	Predominant overstory	Fire frequency
1		Warm, dry ponderosa pine	Pure ponderosa pine	3-8 year intervals
2	**	Grand fir	Larch, Douglas-fir, ponderosa pine, grand fir	20-200 years
3	**	Warm, dry Douglas-fir	Ponderosa pine, Douglas-fir	5-20+ years
4		Cool, dry Douglas-fir	Douglas-fir	35-40 years
5	**	Moist Douglas-fir	Douglas-fir, lodgepole pine, ponderosa pine	around 40 years
6		Cool habitat types	Lodgepole pine	100-500 years
7		Dry, lower subalpine types	Douglas-fir, lodgepole pine, spruce	50-130 years

[1] Fire groups are from Davis and others (1980).

** Only these fire groups have ponderosa pine/Douglas-fir ecosystems. The other groups are included in FIRESUM for future research. All fire groups contain the seven species implemented in FIRESUM.

Table 3—FIRESUM parameter values stratified by fire group. Descriptions of the fire groups are provided in Table 2 (Davis and others 1980)

Parameter symbol[1]	Fire group number 1	2	3	4	5	6	7
k	0.426	0.525	0.426	0.426	0.426	0.426	0.525
BARMAX	.0071	.0149	.0074	.0091	.0107	.0083	.0111
SPM	1.0	6.0	2.0	4.0	3.0	5.0	5.0
PRO	.990	.717	.668	.768	.768	.985	.852
LBULK	15.8	41.6	21.9	25.3	36.0	43.3	38.1
DBULK	76.9	45.8	76.9	110.6	110.6	139.5	142.7

[1] Parameter descriptions:
k = Extinction coefficient (dimensionless)
BARMAX = Maximum attainable basal area (m^2/m^2)
SPM = Maximum seedling density (seedlings/m^2)
PRO = Dead shrubby fuel in shrub biomass (proportion)
LBULK = Bulk density of litter (kg/m^3)
DBULK = Bulk density of duff (kg/m^3)

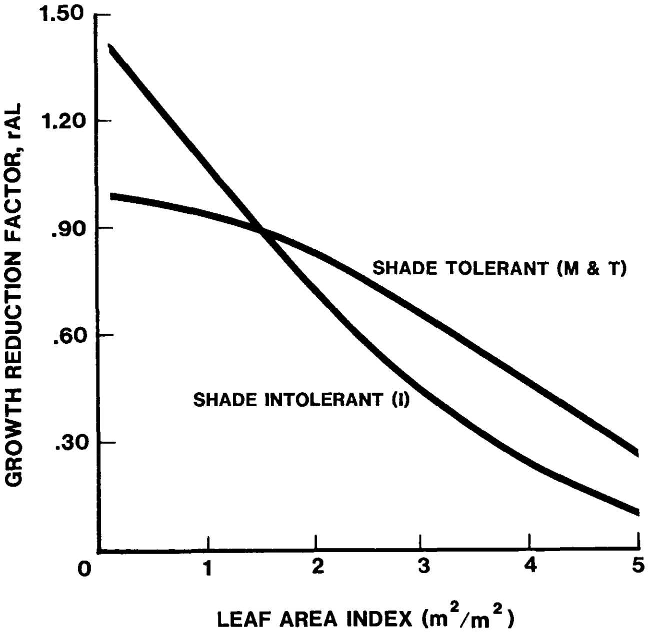


Figure 2—Relationship of the growth reduction factor for shading to leaf area index (equations 4, 5). This range of leaf area indexes is commonly found in ponderosa pine/Douglas-fir forests of the Inland Northwest. Shade tolerant categories *M* (moderately shade tolerant) and *T* (shade tolerant) are represented by the same function. Shade intolerant species (*I*) have a different function.

needle surface-to-volume ratio for species *i* (values are from Lopushinsky [1970], Brown [1970], and Minore [1979]), and PLA_i is a conversion factor to estimate projected leaf area from all-sided leaf area for species *i* (values calculated from Kaufmann and others [1982], Smith [1972], and unpublished data). Crown weight and proportion foliar weight are estimated from regression equations (Brown 1976, 1978; Moeur 1981) that use d.b.h. as the independent variable. All other variables are constants (table 1).

The effect of resource availability (tree crowding) on tree growth was indirectly modeled as a function of stand basal area with the equation:

$$rN = 1 - (BAR/BARMAX_j) \qquad (7)$$

where *BAR* is basal area (square meters) of simulation stand and $BARMAX_j$ is maximum attainable basal area (square meters) for stands in fire group *j*. Values for $BARMAX_j$ (table 3) are estimated from Pfister and others (1977) and Arno and others (1985). The factor *rN* goes to zero as *BAR* approaches $BARMAX_j$ (fig. 3).

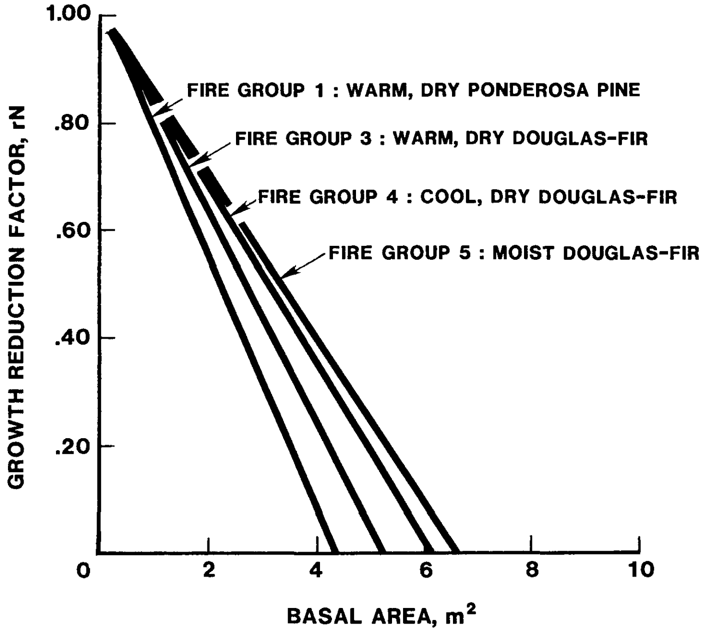


Figure 3—Growth reduction factor (*rN*) relationship to plot basal area in four fire groups.

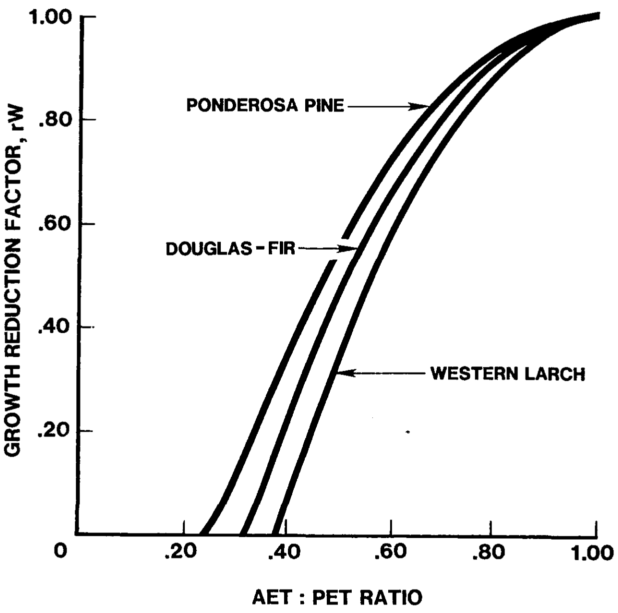


Figure 4—Relationship of the growth reduction factor *rW* to the simulation plot's actual to potential evapotranspiration ratio (AET:PET).

Modeling growth response to water stress (*rW*) in FIRESUM is taken from Reed (1980) and Reed and Clark (1979) where the ratio of actual to potential evapotranspiration (AET:PET) is the driving variable. This ratio indicates the relative aridity of the simulation climate. The water response (*rW*) equation is:

$$rW = 1 - [(1 - APR)/(1 - WSO_i)]^2 \qquad (8)$$

where *APR* is the annual AET:PET ratio for the site and *WSO* is the lower limit of tolerance in *APR* for species *i*. This parabolic function (fig. 4) reaches maximum when *APR* equals 1.0, which assumes growth is not inhibited when annual AET exceeds annual PET. Values of WSO_i for each species were calculated from weather data collected at or near near the edge of species *i*'s natural distribution where water becomes the limiting factor (Little 1971). Actual evapotranspiration is calculated using the water balance equations presented in Kercher and Axelrod (1981), which use monthly precipitation (*BASEP*), soil water-holding capacity (*TEXT*), soil depth (*TILL*), and a runoff constant (*EXCESS*) as variables (values shown in appendix C). Potential evapotranspiration is calculated from the Thornthwaite and Mather (1957) equations.

Climatic influence on diameter growth was modeled as a function of temperature expressed as growing degree-days (Botkin and others 1972; Shugart and Nobel 1981). The parabolic equation taken from Reed and Clark (1979) is given as

when $DMIN_i < DEGD < DMAX_i$:

$$rDEGD = \frac{[(DEGD - DMIN_i)\,(DMAX_i - DEGD)]^V}{[(DOPT_i - DMIN_i)\,(DMAX_i - DOPT_i)]^V} \qquad (9)$$

where $V = (DMAX_i - DOPT_i)/(DOPT_i - DMIN_i)$

when $DEGD < DMIN_i$ or $DEGD > DMAX_i$:

$$rDEGD = 0.0 \qquad (10)$$

where *rDEGD* is a number between 0 and 1.0, *DEGD* is number of degree-days calculated from weather data submitted as input for a simulation run; $DMIN_i$ and $DMAX_i$ are the maximum and minimum degree-days defining the geographic range of species *i*; and $DOPT_i$ is number of degree-days for optimum growth of species *i*.

Figure 5 illustrates the ability of Douglas-fir to grow in colder environments (lower number of degree-days) as compared with ponderosa pine. Note the value *rDEGD* equals 1.0 at $DOPT_i$. Growing degree-days are calculated using equation 9 in Botkin and others (1972). This equation employs a base temperature of 4 °C to define growing season and uses mean monthly temperatures for January and July as minimum and maximum yearly temperatures. $DMAX_i$ and $DMIN_i$ were estimated from weather data collected at extremes of species *i*'s geographical distribution in the Inland Northwest (Shugart 1984). $DOPT_i$ was calculated from weather data at stations that were at or near areas where site index values for species *i* were maximum. Additional information from Alexander and others (1984), Dale and Hemstrom (1984), Fowells (1965), Hellmers and others (1970), and Little (1971) was used to further quantify these three parameters (table 1).

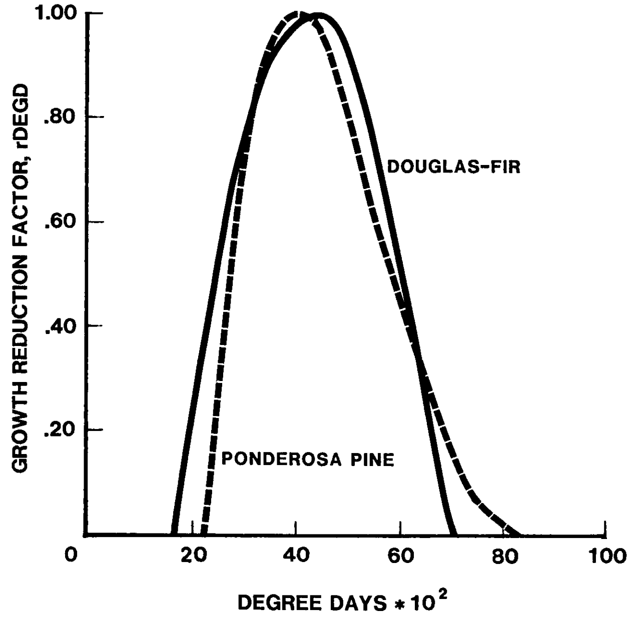


Figure 5—The relationship of degree-days for the simulation plot to the growth reduction factor representing growing season warmth and its effect on tree growth (*rDEGD*).

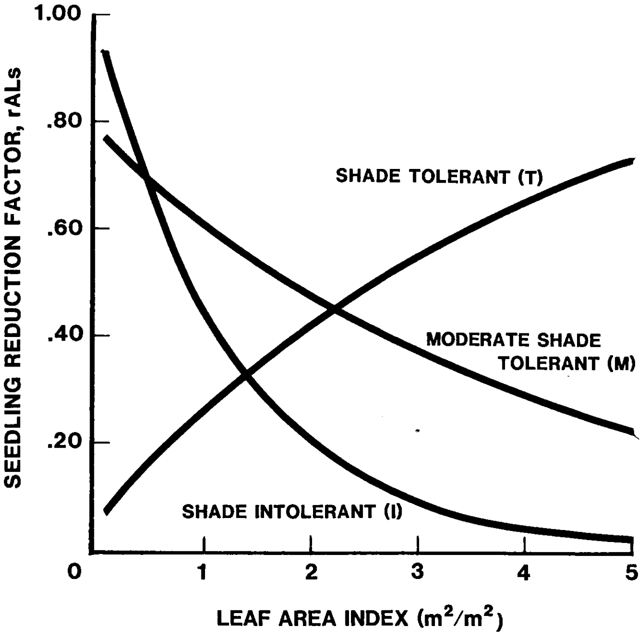


Figure 6—Effects of shading (leaf area index) on the seedling reduction factor *rALs* for three shade tolerance categories.

Tree Regeneration (Subroutine BIRTH)

Trees were established on the simulation plot if two criteria were met. First, simulated growing degree-days (*DEGD*) had to exceed $DMIN_i$ for the species under consideration (Knapp and Smith 1982; Shugart 1984; Weinstein and others 1982); and second, the threshold AET:PET ratio (defined earlier as *APR* in subroutine GROW) had to be greater than WSO_i (Brix 1979; Kercher and Axelrod 1984; Lopushinsky and Klock 1974). If the above criteria were true, then size of cone crop was evaluated.

Each year a species can have a good or poor cone crop, but trees are established only in good seed years. The Monte Carlo method discussed in Kercher and Axelrod (1984) was used to determine good cone crop years. In this stochastic method, p_c is the probabilty of a good cone crop. Each year a random number is generated and, if it is less than p_c, a good cone crop is simulated. But this process is blocked for a number of years after a good cone crop (Kercher and Axelrod 1981). The number of blocked years (h_c–1) is based on the assumption that trees must store sufficient energy reserves before generating another good cone crop. Good seed years are determined at the beginning of each simulation run and remain constant for each replicate run within a simulation. Values for parameters p_c and h_c (table 1) are from Boe (1954), Eis and Craigdallie (1983), Lotan and Perry (1983), Shearer (1985), and Shearer and Schmidt (1970).

The number of trees established on the simulation plot is calculated from the equation:

$$FNJ_i = SPM_j * PTREE_i * PSUR_i * rAL_s * rSRF_i \quad (11)$$

where FNJ_i is the number of seedlings established for species *i*, SPM_j is the maximum number of seedlings (includes all species) that can become established on 1.0 m² for fire group *j*, $PTREE_i$ is proportion of seed trees of species *i*, $PSUR_i$ is the probability of seedling survival considering the duff depth, rAL_s is a reduction factor accounting for effects of limited light on seedling establishment, and *rSRF* is a reduction factor that models the effect of distance of seed source on tree establishment.

The factor rAL_s ranges from 0 to 1.0, depending on three levels of shade tolerance. The set of equations for calculating rAL_s are:

Shade intolerant: $rAL_s = e^{(-0.8*LAI)}$ (12)

Moderate shade tolerant: $rAL_s = e^{(-0.25*LAI - 1.0)}$ (13)

Shade tolerant: $rAL_s = 1 - e^{(-0.25*LAI - 0.2)}$ (14)

where *LAI* is plot leaf area index (square meters of leaf area per square meter of plot area). The coefficients were derived by the authors, based on unpublished data about the dynamics of seedling establishment. Shade-tolerant species are able to establish the most seedlings at low light levels (fig. 6) (Grime and Jeffery 1965).

Values for SPM_j (table 3) were taken from Alexander (1984), Arno and others (1985), Knapp and Smith (1982), Pfister and Shearer (1977), Schimdt and others (1976), Shearer (1974), Shearer (1975), and Shearer (1985). Seed trees were defined as any tree greater than 10 cm d.b.h. or having an age greater than some minimum threshold (variable *YSC*, values shown in appendix B). This assumes only trees meeting these criterion are able to produce appreciable quantities of seed. The variable $PTREE_i$ roughly estimates a species contribution to the seedbank; it is calculated by dividing the number of seed trees for species *i* by the total number seed trees. To account for off-site seed dispersion, tree species not represented on the plot were assigned a value of 0.05 for $PTREE_i$.

$PSUR_i$ was calculated using regression equations developed from a study on litter and duff depth reduction in north Idaho (Boyce 1985). The independent variable in these equations is depth of litter and duff in centimeters (DEPTH). These equations are:

Ponderosa pine:
$$PSUR = 1.0 - 0.164 * DEPTH \qquad R^2 = 0.94 \qquad (15)$$

Grand fir:
$$PSUR = 1.0 - 0.149 * DEPTH \qquad R^2 = 0.90 \qquad (16)$$

Douglas-fir:
$$PSUR = 1.0 - 0.160 * DEPTH \qquad R^2 = 0.99 \qquad (17)$$

Lodgepole pine:
$$PSUR = 1.0 - 0.161 * DEPTH \qquad R^2 = 0.93 \qquad (18)$$

Western larch:
$$PSUR = 1.0 - 0.177 * DEPTH \qquad R^2 = 0.99 \qquad (19)$$

In the absence of specific data for subalpine fir and Engelmann spruce, the grand fir equation was used to represent *PSUR* for those species. Negative values for *PSUR* were equated to zero.

The distance the simulation plot is from seed sources directly influences the number of trees established. Factor *rSRF* attempts to simulate this relationship. Reduction equations are of the form:

$$Y_i = \frac{e^{(a+bX_i)}}{e^a} \qquad (20)$$

where Y_i is the *rSRF* for species *i* with values between 0 and 1, X_i is the distance from species *i*'s seed source, which is input into FIRESUM (value *DIST* in appendix D); and *a* and *b* are species-derived constants based on data provided by McCauley and others (1985). These values (variable *DISEQU*(1,*i*) for *a*, variable *DISEQU*(2,*i*) for coefficient *b*) are shown in appendix B.

All new trees are established as saplings of 1.0 cm diameter at breast height (d.b.h.) and 1.37 m tall. These new trees are added to the simulation after a lag period of 25 to 50 years, depending on the site (value LAG in appendix B).

The absence of seed trees for a species on the plot presents a special case in FIRESUM. Distances to seed source from simulation plot by species are input into the model. The factor *rSRF* and the number of seed trees are computed annually for each species. But the value of *rSRF* only enters into the seedling equation(s) when all seed trees of that species are eliminated from the simulation plot, because of beetle epidemic or successional replacement, for example. It is assumed in FIRESUM that the seed source of eliminated species composes 5 percent of the total seed crop trees outside the simulation plot for all tree species but whitebark pine. If all trees are killed on the plot, such as after a crown fire, the seed source stand is assumed to be identical to the preburn simulation stand.

Whitebark pine regeneration is computed in subroutine PINALB (appendix A), which models the effects of seed crop, Clark's nutcrackers, and light on whitebark regeneration (Keane and others 1989b). This routine is very different from that used for other species and shows how FIRESUM can be modified to simulate life cycles for any tree species. A complete discussion on the whitebark pine regeneration algorithm is presented in Keane and others (1989b).

Tree Mortality (Subroutine KILL)

Four types of tree mortality—random, stress, fire, and insects and disease—are recognized in FIRESUM and are modeled as stochastic functions. "Random mortality" is the chance of death, from endemic insect attack, windthrow, or other local perturbations that a tree experiences throughout its lifetime. The probability of random mortality (P_r) is calculated by the equation:

$$P_r = 4/AGEMAX_i \qquad (21)$$

where $AGEMAX_i$ is the maximum attainable age for species *i*. It was assumed that only 2 percent of the trees survive to $AGEMAX_i$ to derive equation 21 (Botkin and others 1972). Analysis of stand data from Montana, Idaho, and eastern Oregon (Arno and others 1985; Keen 1940; Seidel 1975) suggests that 2 percent is reasonable.

"Stress mortality" is tree death resulting from severe stress over periods of 2 to 50 years. Stress mortality can be caused by water scarcity, insufficient light, or tree crowding (Kercher and Axelrod 1984, Shugart and Noble 1981). The probability of stress mortality (P_s) is a function of growth increment. If a tree's annual growth increment (*DINC*) is less than a threshold value (*AINC*) for that species, the following equation is executed:

$$P_{s(n+1)} = P_{s(n)} + 0.2 - 0.2\, P_{s(n)} \qquad (22)$$

where *n* is the number of stressful years.

A new P_s is calculated each year *DINC* is less than *AINC*. P_s will eventually equal 0.997 after 30 years in this stressed condition. Values for *AINC* were estimated by examining the cross-sections of numerous severely suppressed trees.

Mortality due to fire is modeled as a function of fire intensity. When a fire spreads through an area it kills trees by scorching foliage and killing bole cambium. The degree of crown scorch and cambial kill depends on fire intensity and duration. Ryan and Reinhardt (1988) developed an empirical mortality equation that implicitly accounts for both causes of fire death (fig. 7). The equation is:

$$P_{fk} = \frac{1}{1 + \mathrm{EXP}[-1.941 + 6.32(1 - \mathrm{EXP}(BC_i D_k)) + 0.000535\, CK_k^{\,2}]} \qquad (23)$$

where P_f is the probability of mortality from fire for tree *k*, BC_i is a bark conversion factor for species *i*, which multiplied by D_k (d.b.h. of tree *k* in centimeters) provides an estimate of bark thickness for tree *k*, and CK_k is percentage of scorched crown volume for tree *k*. Values for BC_i are taken from Faurot (1977), Lange (1971), Lynch (1959), Myers and Alexander (1972), and Ryan and Reinhardt (1988).

Assuming crown shape approximates a paraboloid (Peterson 1985; Ryan and Reinhardt 1988), scorched crown volume was estimated using:

$$CK_k = 100\ [B(2L-b)\ /\ L^2] \qquad (24)$$

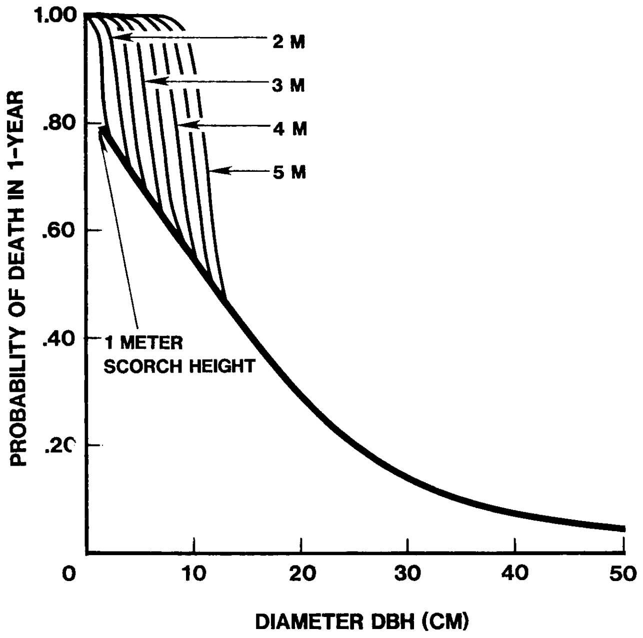


Figure 7—Mortality of ponderosa pine as related to tree diameter (d.b.h.) and crown scorch height. Taken from Ryan and Reinhardt (1988).

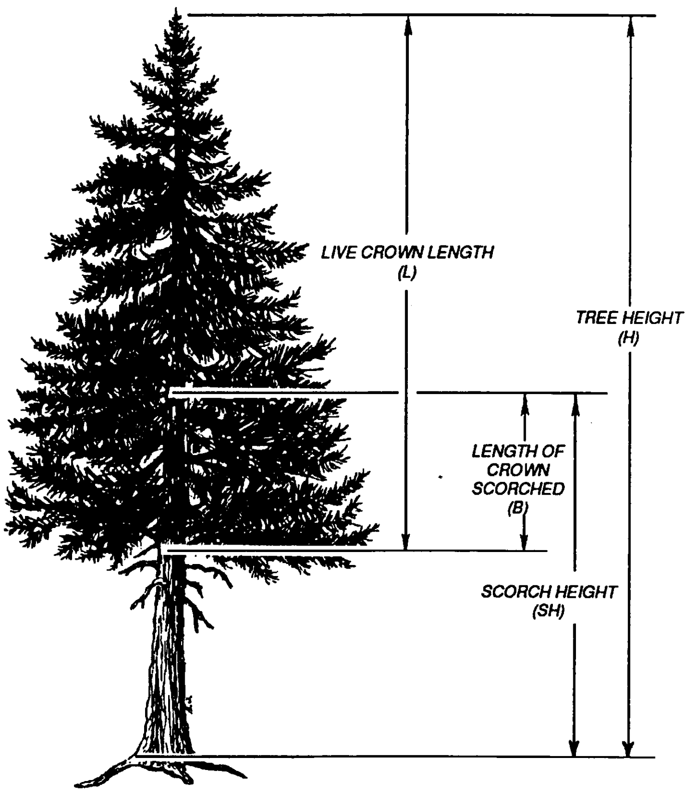


Figure 8—Diagram of important tree dimensions used in the calculation of percent crown scorched. This variable is used to compute tree mortality.

The dimensions B(length of scorched crown in centimeters) and L (length of crown in centimeters) are calculated from tree height and crown length (fig. 8). Tree height (H_k in centimeters) is calculated from the equation:

$$H_k = 137 + b_2 D_k - b_3 D_k^2 \qquad (25)$$

where b_2 and b_3 are the species-dependent constants described in the Tree Growth section. The length of crown (L) is the product of live crown ratio (L_c) and tree height. The dimension B is solved by the equation:

$$B = SH - [H - L] \qquad (26)$$

where SH is scorch height. Scorch height in meters is calculated from an empirical expression developed by Van Wagner (1973):

$$SH = \frac{C_1(FI)^{7/6}}{[C_2(FI) + (C_3(WIND)^3]^{1/2}\,(TKILL - T)} \qquad (27)$$

where FI is fire intensity (kilowatts per meter of fireline), $WIND$ is wind speed (kilometers /hour) at midflame height, T is ambient temperature (degrees Celsius), and $TKILL$ is the lethal temperature for tree foliage (assumed as 60 °C). The constants C_1, C_2 and C_3 were derived empirically and are 0.742 m/°C, 0.0256 $(\mathrm{kW/m})^{4/3}$, and 0.278 h/km $(\mathrm{kW/m})^{7/9}$, respectively. Fire intensity is discussed later. Ambient temperature (T) was assumed to be 20 °C, a typical temperature for prescribed fire. Kercher and Axelrod (1984) found equation (26) to be very sensitive to windspeed at high fuel loadings and insensitive to windspeed at low loadings.

Although the mortality equation (23) includes a wide range of diameters and species, data for small diameter tree mortality were lacking. Because the majority of simulated trees are less than 10 cm d.b.h., additional validation of the equation with small diameter tree mortality is needed.

Insect and disease mortality is the fourth type of tree mortality represented in FIRESUM. Each insect and disease mortality algorithm was developed from empirical data using regression analysis. In the regression equations, probability of mortality (Y-variable) is computed from many types of independent variables (X-variables) including tree diameter, tree densities, proportion of trees infested, and some site variables. Each type of insect or disease is represented by regression equations for each species it may affect. Also, these equations are stratified by fire group. Currently, FIRESUM models mountain pine beetle-caused mortality on lodgepole pine and whitebark pine, and white pine blister rust-caused mortality on whitebark pine in whitebark pine/subalpine fir forests (Keane and others 1989b). Additional insect and disease mortality equations will be included as they are needed.

Each tree that dies, regardless of the cause of mortality, contributes a portion of its woody branchwood and all of its needles to the fuel bed. Weight of branchwood less than 3 inches in diameter for dead trees is calculated from equations by Brown (1978) and divided equally into the three smallest fuel components (discussed in the next section). It is assumed scorched foliage is not consumed by the fire and is added to the fuel bed, unless the fire was a crown fire. It is assumed all foliage is consumed by a crown fire. Needle weight is computed from equations presented in Brown (1978).

Fuel Accumulation (Subroutine FUEL)

Six dead and two live fuel components are recognized in FIRESUM (table 4). Loadings for these eight fuel components are computed annually, and if a fire is simulated, all fuel loadings are passed to subroutine FIRE, where they are used to estimate fire intensity. Accumulation algorithms are used to represent annual fuel increments for (1) dead woody fuel components, (2) litter and duff components, and (3) live and dead shrub and herbaceous components.

The 1-, 10-, and 100-hour timelag dead woody fuel components are updated each year using the following equations:

if $WOOD_{fy} < WOODMAX_{fj}$ then

$$WOOD_{fy+1} = WOOD_{fy} + WOODMAX_{fj}/WYR_{fj} \qquad (28)$$

else if $WOOD_{fy} > WOODMAX_{fj}$ then

$$WOOD_{fy+1} = WOODMAX_{fj} \qquad (29)$$

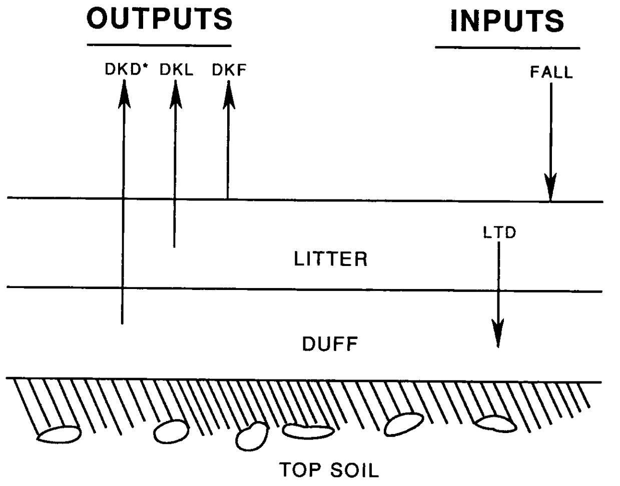
where $WOOD_{fy}$ is fuel loading (kilograms per square meter) for woody fuel component f at year y, $WOODMAX_{fj}$ is the maximum attainable fuel loading for component f in fire group j, and WYR_f is the number of years to reach $WOODMAX_f$ in an undisturbed forest for component f in fire group j. Parameter values (table 5) were taken from Bevins (1977), Brown and Bevins (1986), Brown and See (1981), Jeske and Bevins (1976), Mathews (1972), van Wagtendonk (1972). These equations operate under the assumption of constant accumulation and decomposition rates.

Table 4—Fuel components included in FIRESUM. Timelag woody branchwood categories are described in Fosberg (1970)

Number	Fuel component	Description
		Dead Fuel
		Litter Fuel
1	Litter	Downed tree foliage, no duff material contributes to fire
		Downed Woody Branchwood
2	1-hour time lag	Twigs and branches 0 to 1/4 inch in diameter
3	10-hour time lag	Twigs and branches 1/4 to 1 inch in diameter
4	100-hour time lag	Twigs and branches 1 to 3 inches in diameter
		Shrub and Herbaceous Fuel
5	Shrub	Shrub stemwood 0 to 1 inch diameter
6	Herbaceous	Grass and forbs
		Live Fuel
		Shrub and Herbaceous Fuel
7	Shrub	Foliage and small stemwood on live shrubs
8	Herbaceous	Grass and forbs living on forest floor

Table 5—Parameter values for woody fuel accumulation equations (28) and (29). WOODMAX is the maximum fuel loading and WYR is the time required to reach maximum fuel loading

	Fire group number						
Parameter symbol	1	2	3	4	5	6	7
			1-hour dead woody branchwood				
WYR (years)	40.0	40.0	40.0	30.0	30.0	40.0	50.0
WOODMAX (kg/m²)	.0121	.2710	.0638	.0520	.0520	.1776	.075
			10-hour dead wood branchwood				
WYR (years)	40.0	40.0	40.0	30.0	30.0	40.0	50.0
WOODMAX (kg/m²)	.833	.1548	.2619	.1879	.1879	.4294	.196
			100-hour dead woody branchwood				
WYR (years)	40.0	40.0	40.0	30.0	30.0	40.0	50.0
WOODMAX (kg/m²)	.1546	.1055	.5484	.5365	.5365	.7022	.546



* DKD – portion duff lost to microbial and faunal respiration
DKL – portion litter lost from microbial and faunal respiration
DKF – portion litter lost from overwinter decomposition
LTD – portion litter incorporated into duff
FALL – needlefall from conifer species on the plot

Figure 9—Diagram of litter and duff components. Inputs are noted by the downward-pointing arrows; outputs or losses are shown with upward-pointing arrows. This dynamic system is modeled using the coefficients DKD, DKL, DKF, FALL, and LTD.

Litter and duff loadings are calculated using annual dynamic equations in Kercher and Axelrod (1981). These equations are diagrammed in figure 9. The amount of annual needlefall ($FALL_i$) is calculated from the equation:

$$FALL_i = \Sigma(CW_i) * PFOL_i / NYR_i \qquad (30)$$

where $\Sigma(CW_i)$ is the sum of crown weight over all trees of species i, $PFOL_i$ is the proportion of crown that is needles, and NYR_i is number of years needles remain on a tree of species i. Crown weight and $PFOL_i$ are estimated using regression equations provided by Brown (1976 and 1978). *NYR* values (table 1) are from Fowells (1965), Gottfried and Ffolliot (1983), Harlow and Harrar (1969), Smith (1972), Turner and Long (1975).

Needlefall (kilograms per square meter), the only input to litter-duff equations, is reduced by a species-dependent proportion (DKF_i) to account for overwinter decomposition (fig. 9). The remaining needlefall is added to the litter and subjected to further decomposition losses. A portion of the litter (DKL_i) is lost to the system while another portion (LTD_i) is added to the duff. Duff loading is then decreased by a decomposition proportion (DKD_i), and this decrement is also lost from the system. Decomposition losses in both litter and duff components are due to microbial and microfauna respiration. Each component is updated annually, and the total litter and duff loading for the stand is calculated by summing across all species. Values for DKF_i, DKL_i, LTD_i, and DKD_i for species i (table 1) are taken from Allison and Klein (1961), Edmonds (1979), Fahey (1983), Fogel and Cromack (1977), Jenny and others (1949), Kercher and Axelrod (1984), Klemmedson and others (1985), Gottfried and Ffolliot (1983), Maclean and Wein (1978), Means and others (1985), Meetenmeyer (1978), Piene and Van Cleve (1978), and Yoneda (1975).

Biomass for shrub and herbaceous fuel types are estimated separately using a function provided by Kercher and Axelrod (1984). The shrub and herbaceous equations are identical, except for internal parameters, and assume shrub and herb biomass on a site has an upper limit dependent on stand productivity. Annual change in biomass is a product of current biomass and a factor that limits growth as maximum biomass is approached. The equation is:

$$MASS_{m(y+1)} = MASS_{m(y)} + n * MASS_{m(y*)}[1 - MASS_{m(y)} / BIOMAX_{m(j)}] * rAL \qquad (31)$$

where $MASS_{m(y)}$ is biomass (kilograms per square meter) of fuel component m (shrub or herb) at year y, n is a growth constant for small biomass (per year), $BIOMAX_{m(j)}$ is maximum attainable biomass (kilograms per square meter) for fire group j in fuel component m, and rAL is the light response function presented in the tree growth section (equations 4 and 5). Values for n were taken as 1.44 per year for shrubs (Sampson 1944) and 10.842 per year for herbaceous fuel (from unpublished data collected by the authors). $BIOMAX_{m(j)}$ values (table 6) are from Brown and Bevins (1986), Irwin and Peek (1979), and Martin (1982).

Table 6—Parameter values for maximum biomass (BIOMAX) used to compute loadings of live and dead shrub and herbs in equation (31)

	Fire group number						
Component	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Shrub (kg/m²)	0.027	0.086	0.076	0.069	0.070	0.016	0.054
Herb (kg/m²)	.029	.048	.043	.102	.101	.142	.197

Using light response functions (*rAL*) from the Growth section, shrub and herbaceous loadings are divided into tolerant and intolerant categories. For example, the value for shade intolerant *rAL* (number between 0 and 1) is multipled by total shrub biomass to compute intolerant shrub biomass. Biomass estimates for the herbaceous shade tolerance categories are averaged, and then it is assumed that 90 percent of the average is dead at fire incidence. The remaining 10 percent is treated as live fuel. Shrubby biomass is also averaged across shade tolerance categories, but calculations of dead (*SDEAD*) and live (*SLIVE*) loadings (kilograms per square meter) are accomplished using these equations:

$$SDEAD = SAVE *(1 - PRO_j)/PLOTSIZ\,;\ \text{dead shrubby fuels (kg/m}^2\text{)} \qquad (32)$$

$$SLIVE = SAVE *(PRO_j)/PLOTSIZ\,;\ \text{live shrubby fuels (kg/m}^2\text{)} \qquad (33)$$

where *SAVE* is the average loading (kilograms per square meter) for shade-tolerant and intolerant shrubs, PRO_j is the proportion of dead shrubby fuel in the total shrub biomass for fire group *j*, and *PLOTSIZ* is the simulation plot size (square meters). Values for PRO_j (table 3) are from Brown and Bevins (1986).

Total depth of duff and litter (centimeters) is also calculated in subroutine FUEL using the equation:

$$DEPTH = 100* [(LITT / LBULK_j) + (DUFF / DBULK_j)] \qquad (34)$$

where *DEPTH* is depth of duff and litter (centimeters), *LITT* and *DUFF* are the loadings (kilograms per square meter) of the litter and duff respectively, and $LBULK_j$ and $DBULK_j$ are the bulk densities of the litter and duff strata (respectively) for fire group *j*. Table 3 shows values of $LBULK_j$ and $DBULK_j$ taken from Brown (1981). This depth is then passed to subroutine BIRTH for use in the Boyce (1985) regression equations (equations 15 to 19).

Fire Characteristics (Subroutine FIRE)

Fire frequency is an input to FIRESUM. The user can specify number of years between fires (fire interval), an actual fire history for the stand consisting of variable fire intervals, or a stochastic function that computes fire interval as a dynamic variable using fire frequency probabilities (Kercher and Axelrod 1984). Fire year information is kept in a program array for reference during each year of program execution, similiar to the cone crop array mentioned in the Tree Regeneration section. This array remains unchanged between simulation runs. If the current simulation year is a fire year, fuel loadings and other input parameters are passed to subroutine FIREMOD and fire intensity is computed, then used to calculate scorch height for use in the fire mortality equation. Subroutine FIREMOD was developed by Albini (1976b) using Rothermel's (1972) model for predicting wildland fire spread. FIREMOD calculates Byram's fire line intensity (kilometers per hour) from a multivariate function comprised of the following user-specified parameters.

1. *WIND* = windspeed at midflame height (kilometers per hour)
2. *SLOPE* = slope of stand (degrees)
3. $MOIS_i$ = fractional moisture content of fuel type *i*
4. $RHOP_i$ = ovendry particle density for fuel type *i* (grams per cubic centimeter)
5. $BULK_j$ = bulk density of fuel bed in fire group *j* (grams per cubic centimeter)
6. SVR_{ij} = mean surface to volume ratio for fuel type *i* in fire group *j* (per centimeter)
7. LHV_i = heat content of fuel type *i* (kilojoules per kilogram)
8. ST_i = mineral content fraction of fuel type *i*
9. SE_i = silica-free mineral content fraction of fuel type *i*
10. $MEXT_i$ = moisture of extinction for fuel type *i* (fraction of weight)
11. $FLOAD_i$ = loading of fuel type *i* (kilograms per square meter)

Parameters having constant values across fuel components are *WIND* (kilometers per hour), *MEXT*, and *SLOPE* (degrees) taken from actual stand and site data and input into the model (appendixes B and C), and *LHV* (= 18586.7 kj/kg), *ST* (= 0.055), and *SE* (= 0.011) taken from Albini (1976a) and Anderson (1969). Other parameter values are in tables 7, 8, and 9. Variable $FLOAD_i$ is the only dynamic variable in the multivariate function, computed during program execution and passed to FIREMOD. Values for bulk densities and surface to volume ratios are taken from Brown (1970, 1981); particle densities from Brown (1970) and Anderson (1969); and moisture of extinction values from Frandsen and Andrews (1979) and Rothermel (1972). Values for moisture contents and windspeed are specified by the user and are usually taken from a typical fire prescription or fire weather prediction.

Table 7—Values for input parameters to subroutine FIREMOD stratified by fuel type component. MOIS is the fuel moisture content and RHOP is the surface to volume ratio

	Fuel component number							
Parameter symbol	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
MOIS (proportion)	0.08	0.08	0.10	0.14	0.10	0.08	1.00	1.50
RHOP (g/cm²)	.51	.39	.39	.39	.51	.51	.51	.51

Table 8—Bulk densities (BULK) used in subroutine FIREMOD stratified by fire group

	Fire group number						
Parameter symbol	**1**	**2**	**3**	**4**	**5**	**6**	**7**
BULK (g/cm³)	0.0158	0.0068	0.0080	0.0115	0.0071	0.0126	0.008

Table 9—Surface area to volume ratios (SVR) used in FIREMOD by fire group and fuel component

	Fuel component							
Fire group	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	57.41	8.89	3.48	0.95	3.156	91.86	49.20	91.86
2	57.41	11.76	2.88	.98	3.156	91.86	49.20	91.86
3	57.41	16.00	3.08	.98	3.156	91.86	49.20	91.86
4	57.41	16.00	3.08	.98	3.156	91.86	49.20	91.86
5	57.41	16.00	3.08	.98	3.156	91.86	49.20	91.86
6	57.41	16.00	3.08	.98	3.156	91.86	49.20	91.86
7	57.41	11.76	2.88	.98	3.156	91.86	49.20	91.86
8	57.41	11.76	2.88	.98	3.156	91.86	49.20	91.86

Fuel Consumption (Subroutine BRNOFF)

Fuel reduction by fire is computed using equations from Brown and others (1985), Norum (1974), and Sandberg (1980). The woody fuel reduction equations are:

1 and 10 hour timelag: $WOOD_{consumed} =$

$$0.890\,(WOOD_{pre}) - 0.0060 \qquad (34)$$

100 hour timelag: $WOOD_{consumed} =$

$$0.845\,(WOOD_{pre}) - 0.0150 \qquad (35)$$

Woody fuel reduction equations use preburn fuel loadings ($WOOD_{pre}$ in kilograms per square meter) to estimate fuel consumption ($WOOD_{consumed}$ in kilograms per square meter) independent of fire intensity or moisture content. The proportion of duff reduction, however, is based on preburn duff moisture content (*DMOIST* in percent). The equation for duff reduction is:

$DUFF_{post} =$

$$DUFF_{pre}\,[(83.7 - 0.426 * DMOIST)\,/100.0] \qquad (36)$$

where $DUFF_{post}$ and $DUFF_{pre}$ are duff loadings (kilograms per square meter) postburn and preburn, respectively. A duff moisture content of 50 percent, typical of many fire prescriptions, was used in simulation runs. All litter, dead shrub, and herbaceous biomass is assumed to be consumed by fire, and the live shrub fuel loading was assumed to be reduced by 90 percent of preburn weight.

MODEL OUTPUT

FIRESUM stores average basal area for each tree species by simulation year in an external file. The program also stores fuel component loadings, duff depths, number of established seedlings, and fire behavior statistics. Any of these variables can be graphed against simulation time using various graphic software packages and related hardware (plotters). Figure 10 presents the graphed results of three contrasting simulation runs. The first run (10a and 10b) had fires occurring at 20-year fixed intervals, which could represent a typical prescribed burning scenario. The second run (10c and 10d) had fires occurring at an 8-year stochastic interval, which simulates pre-1900 fire frequency. And the last run (10e and 10f) is the result of a no-fire scenario (fire suppression). Tree species basal area, and fuel loadings are shown for the simulation plot.

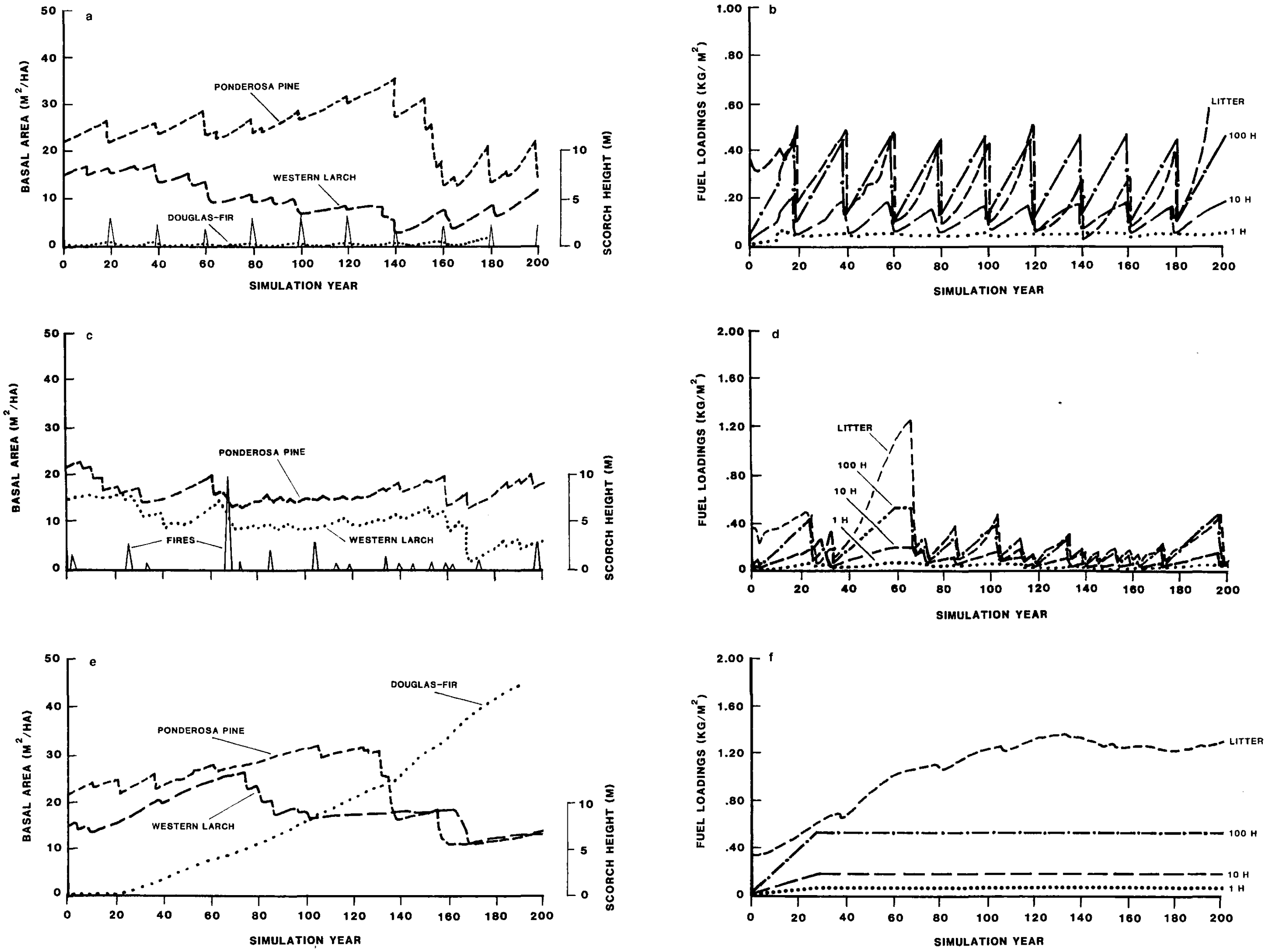


Figure 10 a-f—An example of FIRESUM outputs representing three possible fire scenarios. Graphs 10a and 10b show predicted basal areas and fuel loading for a 20-year fixed fire interval, respectively. Graphs 10c and 10d represent a stochastic fire interval having a mean of 8 years and graphs 10e and 10f predict basal areas and fuel loading in the absence of fires. All graphs are for the same simulation stand and simulate 200 years of ponderosa pine/Douglas-fir succession. Only four fuel components are graphed in 10b, 10d, and 10f: litter, 1-hour, 10-hour, and 100-hour timelage fuel classes. The scorch height of the fires in each scenario is illustrated by the spikes in graphs 10a, 10c, 10e with the corresponding scale located at the far right of these graphs.

MODEL TESTING AND ANALYSIS

Validation and Verification

Testing succession simulation models requires extensive stand data collected at one or more widely separated intervals during successional development. Acquiring these data can be difficult. To test or verify FIRESUM we employed a combination of two methods. The first method was to search the literature for long-term data compatible with the inputs and outputs of FIRESUM. Verification data must have density, age, and diameter (d.b.h.) measurements on trees by species by unit area. These data must have another set of measurements at least 25 to 30 years later, or age-diameter relationships so that regression equations can be developed and used to project a present stand forward or backward in time (Habeck 1985, Keane and others 1989a). The model is then used to simulate conditions measured by these historic data.

The second method of verification involved sampling two adjacent stands on one site. One stand is a mature forest while the other has resulted from a wildfire (disturbance stand). Tree densities, ages, diameters (d.b.h.), and enviromental variables (elevation, aspect, slope, soil depth, etc.) are recorded for each stand (example shown in table 10). The sampled values from the mature stand are used as inputs to FIRESUM. The model is then used to simulate effects of a wildfire on the input stand and grow a subsequent simulation stand of the same age as the

Table 10—Example site and environmental conditions for the ponderosa pine/ Douglas-fir stand used in a FIRESUM execution

Input parameter	Value
Site Description	
Elevation (m)	1,256.0
Slope (degrees)	8.0
Depth to bedrock (ft)	2.5
Water holding cap. (cm/m)	133.3
Fire group	6.0
Fire Weather	
Ambient temperature (°C)	20.0
Wind (km/hour)	3.2
Relative humidity (%)	40.0
Fuel Moisture Contents	
1-hour fuel moisture (%)	8.0
10-hour fuel moisture (%)	10.0
100-hour fuel moisture (%)	14.0
Litter moisture (%)	8.0
Duff moisture (%)	50.0
Dead shrub moisture (%)	10.0
Dead herbaceous moisture (%)	8.0
Live shrub moisture (%)	150.0
Live herbaceous moisture (%)	100.0

Table 11—Results of three tests on the fire succession model FIRESUM

Test	Ecosystem	Basal area	Fuel loading
		- - - *Percent*	*inaccurate* [1] - - -
Test 1	Ponderosa pine/Douglas-fir	12.2	14.6
Test 2	Whitebark pine/subalpine fir	16.2	10.5
Test 3	Whitebark pine	15.5	11.2
Average percentage inaccurate		14.6	12.1

[1] Variable basal area includes basal area for all species on simulation plot. Fuel loading is the total fuel loading (all six fuel components) for the plot. Percentage inaccurate indicates the difference in percentage of the observed from the predicted.

sampled disturbance stand. Results of the simulation are compared with the sampled values from the disturbance stand. The model can be refined or calibrated based on verification results.

Three verification tests have been administered to FIRESUM (table 11) (see Keane and others in press a, in press b). Test results indicated FIRESUM underpredicts basal areas and overpredicts fuel loadings. This is probably due to inaccurate quantification of the parameters involved in the algorithms. Also, site parameters measured for the sample stand could have been in error and model parameters might not have been adequate for these sample sites. These parameters were taken from the literature and may not be applicable to the area or to the site where the test plot was located.

Sensitivity Analysis

A sensitivity analysis of FIRESUM was performed by increasing a selected parameter by 10 percent of its original value and executing the model while holding all other parameters constant. Computer costs and time constraints limited basal area predictions to the average from 30 simulation runs Kelcher and Axelrod 1954). Standard deviations of basal area averaged from 30 runs were below 0.5 m^2/ha; small enough to discern the relative sensitivity of various parameters.

Results of the sensitivity analysis (table 12) agreed closely with those found by Kercher and Axelrod (1984). Maximum age for a tree species ($AGEMAX_i$) was clearly

Table 12—Results of the sensitivity analysis[1]

Parameter		Percent change in PIPO basal area	
Symbol	**Description**	**50 years**	**100 years**
AGEMAX	Maximum age of species	–17.50	–18.10
WOODMAX	Maximum woody fuel loading	–1.41	–.89
SPM	Maximum stocking of seedlings	–1.03	–2.33
DBULK	Bulk density of duff-litter	+1.61	+3.78
WSO	Minimum AET:PET of a species	–7.25	–5.11
DMIN	Minimum number degree-days	–11.34	–7.37
DMAX	Maximum number degree-days	–1.65	+.34
BARMAX	Maximum basal area	+10.01	+11.24
Dm	Maximum diameter	+9.98	+8.86
Hm	Maximum height	+5.01	+2.00
NYR	Years needles stay on tree	+.96	–1.04
DKL	Proportion of decay in litter	+.23	+.36
AINC	Minimum growth rate for mortality	–1.06	–3.11

[1] Values in table are percentage change in ponderosa pine basal area when parameter listed in first column is increased by 10 percent. Sensitivity is measured at the 50th and 100th year of simulation and is calculated from the average of thirty simulation runs.

the most sensitive parameter measured, due to its presence in both the growth and mortality algorithms. In general, parameters directly related to the theoretical growth equation seemed to be the most crucial in FIRESUM. Parameters involved in the calculation of tree regeneration were also important.

An additional, and more extensive, sensitivity analysis was performed for some parameters used in the fire module (FIREMOD). In this analysis, three sets of fuel moisture values for each of three size classes of woody fuel were entered into the model to evaluate overall effect on plot basal area. This process was repeated for three duff moisture values. Results show that dry fuels resulted in an increase in the basal area of ponderosa pine (table 13), presumably because fires ignited in dry fuels tend to be hotter than those ignited in moist fuels. These hotter fires apparently eliminated competing conifers and shrubs, thus allowing greater pine productivity. When duff moisture content was high, very little duff was removed by fire; this adversely affected regeneration of ponderosa pine, and to a lesser degree, Douglas-fir.

Table 13—Sensitivity analysis of fuel moisture values in FIRESUM

Moisture class	Duff moisture content	Ponderosa pine basal area (m^2/ha)[2]		
		25 yr	**50 yr**	**100 yr**
DRY	0.25	24.42	21.85	15.03
MOIST	.25	23.91	22.43	16.72
WET	.25	22.83	23.37	17.62
DRY	.75	19.68	17.34	13.08
MOIST	.75	21.75	18.68	15.30
WET	.75	22.44	22.44	17.50
DRY	1.50	22.91	20.32	15.34
MOIST	1.50	19.77	21.74	16.37
WET	1.50	19.34	22.88	17.24

[1] Fuel moisture contents are for woody fuel components only. The 1-, 10-, and 100-hour timelag moisture contents for the three moisture classes are: DRY (0.05, 0.05, 0.08), MOIST (0.12, 0.12, 0.16), and WET (0.19, 0.19, 0.24).

[2] Values are averages of ponderosa pine basal area from 30 simulation runs. Basal area was recorded for the 25th, 50th, and 100th year of simulation.

SUMMARY AND CONCLUSIONS

FIRESUM is similar to SILVA and many other JABOWA-type models in concept, but it is unique in construction. Related environmental components were integrated in FIRESUM so they depend on each other. Additional ecological processes such as woody fuel accumulation and duff depth-regeneration interaction were added to more completely simulate growing conditions in ponderosa pine/Douglas-fir ecosystems. The fuel and fire submodules were refined to more accurately predict fire behavior, and the regeneration algorithm was extensively modified to account for the role of site conditions in seedling mortality. Because site parameters in FIRESUM were stratified by habitat type groups (fire groups), many stands of differing species and site conditions may be modeled. Lastly, the FIRESUM program was modified by making fuel moisture and other site variables inputs to the program.

FIRESUM could be further modified to more accurately simulate ecological processes. The regeneration algorithm could be reworked to account for additional stochastic elements contributing to seedling establishment (Keane and others in press a, in press b). Cone crop size, seed dissemination, seed germination, and seed lost to birds and animals could be linked to weather and soil conditions. The fuel accumulation and decomposition algorithm could be improved. Currently, FIRESUM does not simulate accumulation and decomposition in woody, shrubby, and herbaceous fuels, as it does for litter and duff. Quantification of decomposition rates in all fuel components and linking the decomposition rates to climatic processes (for example, AET:PET ratio) would enhance the model's predictive value.

Another possible modification would be to develop a more intensive growth equation. The use of growth reduction factors may not allow sufficient resolution to accurately predict subtle changes in tree growth. The fire subroutines could also be modified to account for tree mortality due to crown fires or to root damage, duration of fire and its effect on tree mortality, vertical fire propagation, contribution of large fuels to fire intensity and tree mortality, and reduction of shrub and herbaceous fuels. Other changes might be to more intensively model the effect of soil fertility and water stress on tree growth, develop more accurate leaf area equations, link tree establishment and growth to understory shrub and herbaceous cover, and include off-site seed sources in the regeneration subroutine. Lastly, a wide range of stands with long-term measurements are needed to more accurately estimate the variablity of model predictions.

With the addition or modification of subroutines, FIRESUM could be applied to a broad range of ecological problems in the Inland Northwest. One possible application is to assess the effect of climatic change on tree growth and fire intensity and frequency. Climatic input parameters could be modified using current models that simulate changes in temperature and precipitation over time. Changes in photosynthetic activity due to the "greenhouse effect," increased carbon dioxide, increased temperature, and decreased water availablity, could be simulated by introducing another reduction factor in the growth algorithm. Climatic effects on fire frequency would have to be stochastically linked to the vegetation complex and site environment. FIRESUM might enable us to predict shifts in species composition and structure if the global climate is indeed changing as many scientists contend.

Another important application of FIRESUM is in increasing understanding of ecological processes. Such an understanding could ultimately aid natural resource management. For example, a land manager might wish to use FIRESUM to evaluate cumulative effects of different prescribed burning schedules on tree composition and structure. Other potential uses include assessment of insect- and disease-caused tree mortality related to fire frequency, prediction of stand productivity at varying fire frequencies, and evaluation of wildlife habitat potential under different fire regimes.

REFERENCES

Albini, Frank A. 1976a. Estimating wildfire behavior and effects. Gen. Tech. Rep. INT-30. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 92 p.

Albini, Frank A. 1976b. Computer-based models of wildland fire behavior: a user's manual. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 68 p.

Anderson, Hal E. 1969. Heat transfer and fire spread. Res. Pap. INT-69. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 20 p.

Alexander, Robert R.; Shearer, Raymond C.; Shepperd, W. D. 1984. Silvical characteristics of subalpine fir. Gen. Tech. Rep. RM-115. Fort Collins, CO: U.S. Department of Agriculture, Forest Service, Rocky Mountain Forest and Range Experiment Station. 29 p.

Allison, F. E.; Klein, C. J. 1961. Comparitive rates of decomposition in soil of wood and bark particles of several softwood species. Soil Society Science Proceedings. 25: 193-196.

Arno, Stephen F.; Simmerman, Dennis G.; Keane, Robert E. 1985. Forest succession on four habitat types in western Montana. Gen. Tech. Rep. INT-177. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 74 p.

Bevins, Collin D. 1977. Natural fuels accumulations in lodgepole pine. Final Report. Ogden, UT: Intermountain Forest and Range Experiment Station. 31 p.

Boe, K. N. 1954. Periodicity of cone crops for five Montana conifers. Proceedings of the Montana Academy of Sciences. 14: 5-9.

Botkin, D. B.; Janak, J. F.; Wallis, J. R. 1972. Some ecological consequences of a computer model of forest growth. Journal of Ecology. 60: 849-872.

Boyce, R. B. 1985. Conifer germination and seedling establishment on burned and unburned seedbeds. Moscow, ID: University of Idaho. 71 p. Thesis.

Brix, H. 1979. Effects of plant water stress on photosynthesis and survival of four conifers. Canadian Journal of Forest Research. 9: 160-165.

Brown, James K. 1970. Ratios of surface area to volume for common fire fuels. Forest Science. 16: 101-105.

Brown, James K. 1976. Predicting crown weights for eleven Rocky Mountain conifers. Oslo Biomass Studies IUFRO Congress: 103-115.

Brown, James K. 1978. Weight and density of crowns of Rocky Mountain conifers. Res. Pap. INT-197. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 56 p.

Brown, James K. 1981. Bulk densities of nonuniform surface fuels and their application to fire modeling. Forest Science. 27: 667-683.

Brown, James K.; Bevins, Collin D. 1986. Surface fuel loadings and predicted fire behavior for vegetation types in the northern Rocky Mountains. Res. Note INT-358. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 9 p.

Brown, James K.; Marsden, Michael A.; Ryan, Kevin C.; Reinhardt, Elizabeth D. 1985. Predicting duff and woody fuel consumed by prescribed fire in the northern Rocky Mountains. Res. Pap. INT-337. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 23 p.

Brown, James K.; See, Thomas E. 1981. Downed dead woody fuel and biomass in the northern Rocky Mountians. Gen. Tech. Rep. INT-117. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 48 p.

Dale, Virginia H.; Hemstrom, Miles. 1984. CLIMACS: a computer model of forest stand development for western Oregon and Washington. Res. Pap. PNW-327. Portland, OR: U.S. Department of Agriculture, Forest Service, Pacific Northwest Forest and Range Experiment Station. 60 p.

Davis, Kathleen M.; Clayton, Bruce D.; Fischer, William C. 1980. Fire ecology of Lolo National Forest habitat types. Gen. Tech. Rep. INT-79. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 77 p.

Edmonds, R. L. 1979. Decomposition and nutrient release in Douglas-fir needle litter in relation to stand development. Canadian Journal of Forest Research. 9: 132-140.

Eis, S.; Craigdallie, D. 1983. Reproduction of conifers. A handbook for cone crop assessment. For. Tech. Rep. 31. Ottawa: Canadian Forest Service: 12-27.

Fahey, T. 1983. Nutrient dynamics of aboveground detritus in lodgepole pine (*Pinus contorta* var. *latifolia*) ecosystems in southeastern Wyoming. Ecological Monographs. 53: 51-72.

Faurot, James L. 1977. Estimating merchantable volume and stem residue in four timber species. Res. Pap. INT-196. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 55 p.

Fogel, R.; Cromack, K., Jr. 1977. Effect of habitat and substrate quality on Douglas-fir litter decomposition in western Oregon. Canadian Journal of Botany. 55: 1632-1640.

Fosberg, M. A. 1970. Drying rates of heartwood below fiber saturation. Forest Science. 16: 57-63.

Fowells, H. A. 1965. Silvics of the forest trees of the United States. Agric. Handb. 271. Washington, DC: U.S. Department of Agriculture, Forest Service. 762 p.

Frandsen, William H., Andrews, Patricia L. 1979. Fire behavior in nonuniform fuels. Res. Pap. INT-232. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 34 p.

Grime, J. P.; Jeffery, D. W. 1965. Seedling establishment in verticle gradients of sunlight. Journal of Ecology. 53: 621-642.

Gottfried, Gerald J.; Ffolliot, P. F. 1983. Annual needle and leaf fall in an Arizona mixed conifer stand. Res. Note RM-428. Fort Collins, CO: U.S. Department of Agriculture, Forest Service, Rocky Mountain Forest and Range Experiment Station. 4 p.

Habeck, James R. 1985. Changes in stand structure related to fire history in ponderosa pine/Douglas-fir forests in Pattee Canyon, Missoula County, Western Montana. Final Report 22-C-3-INT-132. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 44 p.

Harlow, W. M.; Harrar, E. S. 1969. Textbook of dendrology. 5th ed. New York: McGraw-Hill. 527 p.

Hellmers, H.; Genthe, M. K.; Ronco, F. 1970. Temperature affects growth and development of Engelmann spruce. Forest Science. 16: 447-452.

Hunt, F. A. 1986. National register of big trees. American Forests. 94(4): 21-52.

Irwin, L. L.; Peek, James M. 1979. Shrub production and biomass trends following five logging treatments within the cedar-hemlock zone of northern Idaho. Forest Science. 25: 415-426.

Jenny, H.; Gessel, S. P.; Bingham, F. T. 1949. Comparative study of decomposition rates of organic matter in temperate and tropical regions. Soil Science. 68: 419-432.

Jeske, B. W.; Bevins, Collin D. 1976. Spatial and temporal distribution of natural fuels in Glacier National Park. In: First conference on sciencitific research in the National Parks; 1976 November 9-13; New Orleans, LA: 1-21.

Kaufmann, Merrill R.; Edminster, Carleton B.; Troendle, Charles A. 1982. Leaf area determinations for subalpine tree species in the central Rocky Mountains. Res. Pap. RM-238. Fort Collins, CO: U.S. Department of Agriculture, Forest Service, Rocky Mountain Forest and Range Experiment Station. 7 p.

Keane, Robert E.; Arno, Stephen F.; Brown, James K. [In press a.] Modeling stand dynamics in whitebark pine forests. Accepted by Ecology.

Keane, Robert E.; Arno, Stephen F.; Brown, James K.; Tombeck, Diana F. [In press b.] Modeling stand dynamics in whitebark pine forests. Accepted by Ecological Modeling.

Keen, F. P. 1940. Longevity of ponderosa pine. Journal of Forestry. 38: 597-598.

Kercher, James R.; Axelrod, M. C. 1981. SILVA: a model for forecasting the effects of SO_2 pollution on growth and succession in a western coniferous forest. Ecological Modelling. 23: 165-184.

Kercher, James R.; Axelrod, M. C. 1982. Modeling the individual and community effects of air pollution stress in a western USA coniferous forest. UCRL-87747. Livermore, CA: Lawrence Livermore National Laboratory. 42 p.

Kercher, James R.; Axelrod, M. C. 1984. A process model of fire ecology and succession in a mixed-conifer forest. Ecology. 65(6): 1725-1742.

Kessell, S. R.; Potter, M. W. 1980. A quantitative succession model for nine Montana forest communities. Environmental Management. 4: 112-115.

Klemmendson, J. O.; Meier, C. E.; Campbell, R. E. 1985. Needle decomposition and nutrient release in ponderosa pine ecosystems. Forest Science. 31(3): 647-660.

Knapp, A. K.; Smith, W. K. 1982. Factors influencing understory seedling establishment of Engelmann spruce (*Picea engelmannii*) and subalpine fir (*Abies lasiocarpa*) in southwest Wyoming. Canadian Journal of Botany. 60: 2753-2761.

Lange, Robert W. 1971. Bark thickness, *k*, factors for four Montana coniferous tree species. Res. Note 9. Missoula, MT: Montana Forest and Conservation Experiment Station, School of Forestry, University of Montana. 2 p.

Little, Elias L. 1971. Atlas of United States trees. Vol. 1. Conifers and important hardwoods. Misc. Publ. 1146. Washington, DC: U.S. Department of Agriculture, Forest Service. 204 p.

Lopushinsky, William. 1970. Relationship of needle surface area to needle volume in ponderosa pine and lodgepole pine. Res. Note PNW-126. Portland, OR: U.S. Department of Agriculture, Forest Service, Pacific Northwest Forest and Range Experiment Station. 4 p.

Lopushinsky, W.; Klock, G. O. 1974. Transpiration of conifer seedlings in relation to soil water potential. Forest Science. 20: 181-186.

Lotan, James E.; Perry, David A. 1983. Ecology and regeneration of lodgepole pine. Agric. Handb. 606. Washington, DC: U.S. Department of Agriculture, Forest Service. 51 p.

Lynch, Donald W. 1959. Effects of wildfire on mortality and growth of young ponderosa pine trees. Res. Note INT-66. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 8 p.

Maclean, D. A.; Wien, R. W. 1978. Weight loss and nutrient changes in decomposing litter and forest floor material in New Brunswick forest stands. Canadian Journal of Botany. 56: 2730-2749.

Martin, Robert E. 1982. Shrub control by burning before timber harvest. In: Baumgartner, David M., ed. Site preparation and fuels management on steep terrain: proceedings of a symposium; 1982 February 15-17; Spokane, WA. Pullman, WA: Washington State University, Cooperative Extension: 35-40.

Mathews, Edward E. 1972. The accretion of fuel in lodgepole pine forests of southwest Montana. Missoula, MT: University of Montana. 39 p. Thesis.

McCaughey, Ward W.; Schmidt, Wyman C.; Shearer, Raymond C. 1986. Seed dispersal characteristics of conifers of the Inland Northwest. In: Shearer, Raymond C., compiler. Proceedings—conifer tree seed in the Inland Mountain West; 1985 August 5-6; Missoula, MT. Gen. Tech. Rep. INT-203. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Research Station: 50-63.

Means, J. E., ed. 1981. Proceedings of a symposium on forest succession and stand development research in the Northwest. Corvallis, OR: Oregon State University, Forest Research Laboratory. 170 p.

Means, J. E.; Cromack, K.; Macmillan, P. C. 1985. Comparison of decomposition models using wood density of Douglas-fir logs. Canadian Journal of Forest Research. 15: 1092-1098.

Meetenmeyer, V. 1978. Macroclimate and lignin control of decomposition rates. Ecology. 59: 465-472.

Minore, D. 1979. Comparative autecological characteristics of northwestern tree species: a literature review. Gen. Tech. Rep. PNW-87. Portland, OR: U.S. Department of Agriculture, Forest Service, Pacific Northwest Forest and Range Experiment Station. 28 p.

Moeur, Melinda. 1981. Crown width and foliage weight of northern Rocky Mountain conifers. Res. Pap. INT-283. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 14 p.

Myers, Clifford A.; Alexander, Robert A. 1972. Bark thickness and past diameters of Engelmann spruce in Colorado and Wyoming. Res. Note RM-217. Fort Collins, CO: U.S. Department of Agriculture, Forest Service, Rocky Mountain Forest and Range Experiment Station. 2 p.

Norum, Rodney A. 1974. Fire intensity-fuel reduction relationships associated with understory burning in larch/Douglas-fir stands. In: Proceedings of a symposium: Tall Timbers Fire Ecology Conference Number 14 and Intermountain Fire Council and Land Management; 1974 October 8-10, Missoula, MT. Tallahassee, FL: Tall Timbers Research Station: 559-572.

Pando, R. 1973. A.F.A.'s social register of big trees. American Forests. 79: 21-47.

Peterson, D. L. 1985. Crown scorch volume and scorch height: estimates of post-fire tree condition. Canadian Journal of Forest Research. 15: 596-598.

Pfister, R. D.; Shearer, R. C. 1977. Potential vegetation as an indicator of reforestation opportunity. In: National Silviculture Workshop: Proceedings; 1977 September 26-30; Flagstaff, AZ: 1-21.

Pfister, Robert D.; Kovalchik, Bernard L.; Arno, Stephen F.; Presby, Richard C. 1977. Forest habitat types of Montana. Gen. Tech. Rep. INT-34. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 174 p.

Piene, H.; Van Cleve, K. 1978. Weight loss of litter in cellulose bags in a thinned white spruce forest in interior Alaska. Canadian Journal of Forest Research. 8: 42-46.

Reed, K. L. 1980. An ecological approach to modeling growth of forest trees. Forest Science. 26: 33-50.

Reed, K. L.; Clark, S. G. 1979. SUCcession SIMulator: a coniferous forest simulator: model documentation. Bull. 11. Seattle: University of Washington, Coniferous Biome Ecosystem Analysis. 96 p.

Rothermel, Richard C. 1972. A mathematical model for predicting fire spread in wildland fuels. Res. Pap. INT-115. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 40 p.

Ryan, Kevin C.; Reinhardt, Elizabeth D. 1988. Predicting postfire mortality of seven western conifers. Canadian Journal of Forest Research. 18: 1291-1297.

Ryan, Kevin C.; Peterson, David L.; Reinhardt, Elizabeth D. 1987. Modeling long-term fire-caused mortality of Douglas-fir. Forest Science. 34(1): 190-199.

Sampson, A. W. 1944. Plant succession on burned chaparral lands in northern California. Bull. 685. Berkeley, CA: University of California, Agricultural Experiment Station. 144 p.

Sandberg, David V. 1980. Duff reduction by prescribed underburning in Douglas-fir. Res. Pap. PNW-272. Portland, OR: U.S. Department of Agriculture, Forest Service, Pacific Northwest Forest and Range Experiment Station. 18 p.

Schmidt, Wyman C.; Shearer, Raymond C.; Rowe, Arthur L. 1976. The ecology and silviculture of western larch forests. Tech. Bull. 1520. Washington, DC: U.S. Department of Agriculture, Forest Service. 96 p.

Seidel, K. W. 1975. Response of western larch to changes in stand density and structure. Res. Note PNW-258. Portland, OR: U.S. Department of Agriculture, Forest Service, Pacific Northwest Forest and Range Experiment Station. 4 p.

Shearer, Raymond C. 1974. Early establishment of conifers following prescribed broadcast burning in western larch/Douglas-fir forests. In: Proceedings of a symposium: Tall Timbers Fire Ecology Conference Number 14 and Intermountain Fire Council and Land Management; 1974 October 8-10, Missoula, MT. Tallahassee, FL: Tall Timbers Research Station: 481-500.

Shearer, Raymond C. 1975. Seedbed characteristics in western larch forests after prescribed burning. Res. Pap. INT-167. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 26 p.

Shearer, Raymond C. 1985. Cone production on Douglas-fir and western larch in Montana. In: Proceedings—conifer tree seed in the Inland Mountain West symposium; 1985 August 5-6; Missoula, MT. Gen. Tech. Rep. INT-203. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Research Station: 63-68.

Shearer, Raymond C.; Schmidt, Wyman C. 1970. Natural regeneration in ponderosa pine forests in western Montana. Res. Pap. INT-86. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 20 p.

Shugart, H. H. 1984. A theory of forest dynamics. New York: Springer-Verlag. 278 p.

Shugart, H. H.; Nobel, 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.; West, D. C. 1980. Forest succession models. BioScience. 30(5): 308-313.

Smith, J. H. G. 1972. Persistence, size, and weight of needles on Douglas-fir and western hemlock branches. Canadian Journal of Forest Research. 2: 173-178.

Steele, R.; Geier-Hayes, K. 1989. The grand fir/blue huckleberry habitat type in central Idaho: succession and management. Gen.Tech. Rep. INT-228. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 66 p.

Stickney, Peter F. 1980. Data base for post-fire succession, first 6 to 9 years, in Montana larch-fir forests. Gen. Tech. Rep. INT-62. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 133 p.

Thornthwaite, C. W.; Mather, J. R. 1957. Instructions and tables for computing potential evapotranspiration and the water balance. Drexel Institute of Technology Publications in Climatology. 10: 180-311.

Turner, J.; Long, J. N. 1975. Accumulation of organic matter in a series of Douglas-fir stands. Canadian Journal of Forest Research. 5: 681-690.

Van Wagner, C. E. 1973. Height of crown scorch in forest fires. Canadian Journal of Forest Research. 3: 373-378.

Van Wagtendonk, Jan W. 1972. Fire and fuel relationships in mixed conifer ecosystems of Yosemite National Park. Berkeley, CA: University of California Wildland Resource Science. 210 p. Dissertation.

Weinstein, D. A.; Shugart, H. H.; West, D. C. 1982. The long-term nutrient retention properties of forest ecosystems: a simulation investigation. ORNL/TM-8472. Oak Ridge, TN: Oak Ridge National Laboratory, Environmental Sciences Divison. 145 p.

Yoneda, T. 1975. Studies on the rate of decay of wood litter on the forest floor. I. Some physical properties of decaying wood. Japanese Journal of Ecology. 25: 40-46.

APPENDIX A: LISTING OF THE FORTRAN77 COMPUTER CODE FOR THE MODEL/PROGRAM FIRESUM

```
      PROGRAM FIRESUM
C ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C           Model for simulation of western conifeous forests.         *
C This version was coded by Bob Keane, Quantitative Ecologist.         *
C Modifications and alterations to original SILVA were accomplished by *
C                          Bob Keane                                   *
C  please contact Bob Keane for information on its use                 *
C                Keane:                                                *
C                 PO Box 8089                                          *
C                 USFS Fed. Bldg.                                      *
C                 Missoula, MT  59806                                  *
C                 406-329-3390                                         *
C                 fts: 585-3390 OR 584-4867                            *
C                 com: 406-329-4837 or 406-329-3390                    *
C ******************************************************************** *
C ***********      FIRESUM: Keane version June 6, 1989         ******* *
C *********** Fire effects in Ponderosa pine-Douglas-fir stands******* *
C *********** Fire effects in Whitebark pine forests           ******* *
C ******************************************************************** *
C         FIRESUM coding is an extensive modification of jabowa        *
C               botkin et. al. j. ecol. 60:849-872 (1972).             *
C          ****************************************************        *
C                ***********notice to users**************              *
C          SUBROUTINEs birth,dist,kill,cycles, and rings use the       *
C          random number generator rgen.  this is called as XRANDOM,   *
C          where x is a returned array of random numbers between       *
C          0 and 1 and n is the number of returned values of x.        *
C          rgen and the seed generator ranst are at the END of the     *
C          deck.  ranst is called once at the beginning of the job     *
C          to set the seed for rgen.  the user should replace these    *
C          SUBROUTINEs with his own random number generator.           *
C          Input files should have names:tredat,sitdat,and contrl.     *
C          Unit numbers are: 1-tredat.dat,2-sitdat.dat,3-contrl.dat.   *
C                      **************************                      *
C Modification of FIRESUM                                              *
C     Value of nj in birth rounded instead of trun                     *
C     Polar method used to generate gauss r.v. in birth                *
C Corrections to SUBROUTINE site and input of tree water response      *
C     in SUBROUTINE tree                                               *
C     SUBROUTINE grow and birth modified for water stress              *
C     still use botkins thornthwaite method                            *
C     this version has option of suppression effects of water stress   *
C            nwtstr = 1 water stress exists                            *
C            0 no water stress, optimum growth                         *
C This version prints out results in units of sq. m./ha. orsq.cm/sq.m. *
C This version uses three scratch arrays and uses them in shade also.  *
C This version sets limit of 3000 trees.                               *
C This version has been cleaned for sending to fws                     *
C This version has fortran random number generator XRANDM,only         *
C                   for export.                                        *
C This version is now ansi compatible for export.                      *
C This version replaces rnfl in dist with rgen                         *
C                                                                      *
C Dynamic Variables:                                                   *
C   AGE(j) - vector of tree ages (years)                               *
C   DBH(j) - vector of tree diameters (cm)                             *
C   OCCUR(j) - binary vector of fire years (0 or 1)                    *
C   DEGD - number of degree days for simulation plot.                  *
C   SLA(j) - vector of tree leaf areas (m2/m2)                         *
C   NTREES(i) - species vector for number of trees per species.        *
```

```
C   NTREES(i) - species vector for number of trees per species.        *
C   TABLE(j) - table of possible seed years for every species.         *
C   S1(j),S2(j),S3(j) - working arrays for program execution.          *
C   NDEAD(i) - number of dead trees per species.                       *
C   DDBH(j) - working vector for tree dbh.                             *
C   ABAR(j) - vector of basal areas for trees (m2/ha)                  *
C   PD(j) - probability of survival for each i tree.                   *
C Input Variables:                                                     *
C   ASIDE(i) - allsided to projected leaf area conversion factor.      *
C   C(7) - Coefficient in crown weight equations.                      *
C   ALPHA(7) - Coefficient in crown weight equations.                  *
C   B2(7) - calculated coefficient for height equation.                *
C   B3(7) - calculated coefficient for height and growth equation.     *
C   CEXT(8) - extinction factor for each fire group canopy.            *
C   CRAT(7) - live crown ratio for each species.                       *
C   SIGMA(7) - parameter for converting crown weight to leaf area.     *
C   AP(7) - alpha regression coefficient for                           *
C   BETAP(7) - beta regression coefficient for                         *
C   G(7) - calculated growth parameter for diameter increment.         *
C   AGEMX(7) - maximum attainable age by species.                      *
C   DM(7) - maximum attainable diameter by species.                    *
C   HM(7) - maximum attainable height by species.                      *
C   SPM(8) - maximum attainable seedling stocking by fire group.       *
C   XMBAR(8) - maximum attainable basal area by fire group.            *
C   PHI - maximum relativized available light (=1.0)                   *
C   DMIN(7) - minimum number of degree days by species.                *
C   DMAX(7) - maximum number of degree days by species.                *
C   DOPT(7) - optimal number of degree days by species.                *
C   BASET(12) - temperature by month (oC).                             *
C   BASEP(12) - precipitation by month (cm).                           *
C   BASEH - base elevation or elevational difference from w.s. to plot.*
C   ROCK - percent of exposed rock on plot.                            *
C   TILL - depth of soil in meters.                                    *
C   TEXT - soil water holding capacity in mm/m.                        *
C   EXCESS - prop of precip that is runoff.                            *
C   PLTSIZ - area of simulation plot (m2).                             *
C   IFG - fire group number.                                           *
C   SURA(7) - alpha coefficient for seedling survival equations.       *
C   SURB(7) - beta coefficient for seedling survival equations.        *
C   DBULK(8,2) - bulk density of litter and duff by fire group.        *
C   ISHADE(7) - shade tolerance by species.                            *
C   IMOIST(7) - moisture tolerance by species.                         *
C   MEXT(2) - moisture of extinction by live or dead fuel class.       *
C   RHOP(7) - fuel particle density for each fuel component.           *
C   BULK(2,8) - bulk densities for live and dead fuel by fire group.   *
C   MOIS(2,7) - moisture content of live and dead fuel components.     *
C   MPS(2,7) - surface area to volume ratio for live and dead comp.    *
C   LHV(2,7) - latent heat content of each fuel component.             *
C   ST(2,7) - fraction mineral content of dead fuel.                   *
C   SE(2,7) - fraction mineral content of fuel excluding silica.       *
C   DKD(7),DKL(7),DKF(7),LTD(7) - decomposition proportions for litter.*
C   ABM(7),FFL(7) - parameters quantifing litterfall loadings.         *
C   DMOIST - duff moisture content.                                    *
C   NS - number of species.                                            *
C   NSPAN - time span (in years) to simulate.                          *
C   NRUNS - number of times to repeat each simulation time span.       *
C   CLRCUT - flag indicating if stand originated as a clearcut.        *
C   NWRSTR - flag indicating if water stress factors are included.     *
C   IFIRE - number of years between fires (fire interval).             *
C   SBURN - proportion of burnable live shrubs.                        *
```

```
C   BC(7) - bark thickness conversion factor by species.               *
C   D1(7),D2(7),D3(7) - coefficients in fire mortality equation.       *
C   FWG(2,7) - working array for fuel loadings by component.           *
C   WOOD(3) - woody fuel loading by size class 1,10,100 hr.            *
C   NDYR(7) - number years needles stay on tree of species i.          *
C   CROP(7) - number of years between good cone crops.                 *
C   CBLOCK(7) - number of years before a good cone crop can occur.     *
C   GRWS(7) - growth reduction from water stress.                      *
C   GRF(7) - growth reduction from pollution.                          *
C This program calls 45 subroutines and 3 functions, consists of about *
C 3000 lines of code, and is written in FORTRAN 77.                    *
C **********************************************************************
      common/leaf/aside(7),c(7),alpha(7),b2(7),b3(7),cext(8),crat(7),
     &            sigma(7),ap(7),betap(7)
      common/trunk/g(7),agemx(7),dm(7),hm(7),spm(8),ysc(7)
      common/hdata/phi,xmbar(8),degd,ainc(7),binfest(8),ibcycle(8),brr
      common/climat/ dmin(7),dopt(7),dmax(7),baset(12),basep(12),baseh
      common/water/grws(7),ws0(7),wsm(7),nws(7),wr,grdd(7),grbar(7)
      common/plotq/elev,rock,till,soilm,text,excess,pltsiz,ifg
      common/birthk/sura(7),surb(7),dbulk(8,2),disequ(2,7),rdelay(8)
      common/types/ishade(7),imoist(7),spp(7)
      common/wbark/ cmax,agecon,dbhmin,birds,spc,spcac,cyr(4),fmax,cpt,
     &              pfind,ssc
      common/fuel1/ mext(2),rhop(2,7),bulk(2,8),mois(2,7)
      common/fuel2/ mps(2,7),lhv(2,7),st(2,7),se(2,7)
      common/fuel3/ dkl(7),dkd(7),dkf(7),ltd(7)
      common/fuel4/ abm(7),ffl(7),fyr(3,8),fload(3,8)
      common/fuel5/ amc(7),bmc(7),cmc(7),dmc(7),mmc(7),tmc,emc(7)
      common/cfire/cbd(7),vfl(7),cfmc(7),vfmc(7),cflm(7),csvr(7),
     &             vsvr(7),bl(7)
      common/sites/ occur(500),rh,wind,ttheta,t
      common/mort/ d1(7),d2(7),d3(7),bc(7)
      common/polut/ndyr(7),dmoist
      common/oper/ ns,nspan,nruns,clrcut,nwrstr,ifire,sburn,ibr,impb
      common/init/ ntrees0(7),dbh0(4000),ncount(20,7),nbins,width,
     &             age0(4000),agein(20,7)
      common/limits/ mxtrs,maxspc,mxdd,mxyrs,maxbin
      real mext,lhv,mmc,mps,ltd,mois,emc,cyr
      real dbh(4000),sla(4000),pd(4000),fwg(2,7),wood(3),ptree(7),
     &     dsw(7),dsize(7)
      real s1(4000),s2(4000),s3(4000),crop(7)
      real grf(7),area(4500),abar(4500),age(4000)
      integer table(4500),ntrees(7),ndead(7),cblock(7),clrcut,occur
      integer fyr,itop(4000)
      character*1 ishade,imoist,spp*4

      open(unit=5,file='OUTFILE.DAT',form='formatted',recl=150,
     &     pad='yes')

      nl = 2
      npine = 0
      lag = 0
      branch = 0.0
      iseed = 0
      icwf = 0

C ..... Call to initialize random number generator user should
C ..... introduces his own random number generator
      xran = rrand()
```

```
      call sitdta
      call tree(nl,crop,cblock)
      call calcnt
      call cntrl(dsize,dimax,dimin)
      call dist(sl,itop)
      nyears= nspan
      nsp= ns
      do 5 i = 1,ns
           dsw(i) = 0.0
           if(dsize(i) .lt. 0.0) dsw(i) = abs(dsize(i))
           if(dsize(i) .gt. 0.0) dsw(i) = sqrt(5000.0*dsize(i)*3.14159)
    5 continue

      call site

      close(5)

      open(unit=8,file='WOOD.DAT',pad='yes',recl=80)
      open(unit=9,file='BRUSH.DAT',pad='yes',recl=80)
      open(unit=11,file='FIRE.DAT',pad='yes',recl=80)
      open(unit=12,file='DUFF.DAT',pad='yes',recl=100)

      do 10 i = 1,nruns
           irun = i
           if(irun .eq. 2) then
                close(8)
                close(9)
                close(11)
                close(12)
           endif
           print *,'FIRESUM run number: ',i
           iseed = 0
           call cycles(table,nyears,nsp,sl,crop,cblock)
           call rings(nyears,sl)
           call starter(ntrees,dbh,age)
           do 20 k = 1,nspan
                kyr = k
                call birth(ntrees,dbh,age,sl,s2,table,nyears,kyr,duff,
     &                    irun,itop,dsw,ccrop,lag,iseed,ptree)
                call pollut(grf,kyr)
                call grow(dbh,pd,ntrees,sla,grf,sl,s2,s3,age,kyr,itop,
     &                   inend,npine)
                call fire(ntrees,dbh,fwg,nl,pd,kyr,duff,branch,wood,
     &                   irun,icwf,dimax,dimin)
                if(icwf .eq. 1) iseed = 1
                call kill(ntrees,ndead,dbh,pd,sl,age,branch,itop,
     &                   icwf)
                call basal(ntrees,dbh,kyr,nyears,area)
                if(icwf .eq. 1) then
                     lag = kyr + ifix(rdelay(ifg))
                endif
   20      continue
           nm= nspan*ns
           call avg(area,abar,nm)
   10 continue

C ..... printing annual basal area projections
      call output(abar,nyears)

      stop
```

```
      END
      BLOCKDATA
C ***********************************************************************
C  BLOCK DATA FOR SIMULATION RUN SPECIFICS
C     These numbers set operating limits on model:
C     mxtrs....... maximium number of trees allowed on the stand
C     mxyrs....... max number of years in one "run" of the model
C     maxbin...... max num of diam cohorts in initial dist of tree sizes
C     maxspc...... max number of tree species
C     mxyrs....... maximum number of years  the model may run
C ***********************************************************************
      common/limits/ mxtrs,maxspc,mxdd,mxyrs,maxbin

      data mxtrs/4000/
      data mxdd/4000/
      data mxyrs/500/
      data maxbin/20/
      data maxspc/7/

      END
      SUBROUTINE add(x,nx,xnew,new,k)
C ***********************************************************************
C This subroutine adds new trees to the DBH array (x(i,j)) in the
C appropriate species and DBH slots.  This subroutine is called from
C BIRTH and adds seedlings 137 cm tall and 1.0 cm diameter.
C ***********************************************************************
      common/oper/ ns,nspan,nruns,clrcut,nwrstr,ifire,sburn,ibr,impb
      integer clrcut
      dimension x(1),nx(1),xnew(1)

C ..... Add the elements of xnew to x after species k
C ..... nx array  is not updated
      if (new.eq.0) return
      n= isum(nx,ns)
      kk= isum(nx,k)
      nkk= n-kk
      if (nkk.eq.0) go to 15
      do 10 j= 1,nkk
            x(n+new-j+1)= x(n-j+1)
   10 continue
   15 continue
      do 20 j= 1,new
            x(kk+j)= xnew(j)
   20 continue

      return
      END
      SUBROUTINE avg(x,xbar,n)
C ***********************************************************************
C This subroutine averages the annual estimates of basal area for
C every run.  This is a running average and variance is not computed.
C ***********************************************************************
      dimension x(1),xbar(1)
      integer count
      data count/0/

      count= count+1
      w1= float(count-1)/float(count)
      w2= 1./float(count)
```

```
C ..... Estimate the average and include in array
      do 10 i= 1,n
            xbar(i)= w1*xbar(i)+w2*x(i)
   10 continue
      return
      END
      SUBROUTINE basal(ntrees,dbh,kyr,nyears,area)
C ********************************************************************
C * Subroutine basal keeps a continous account of species basal      *
C * area by simulation year.  These values are stored in working     *
C * array AREA(i,j) to be printed in subroutine OUTPUT.              *
C ********************************************************************
      common/plotq/elev,rock,till,soilm,text,excess,pltsiz,ifg
      common/oper/ ns,nspan,nruns,clrcut,nwrstr,ifire,sburn,ibr,impb
      integer clrcut
      dimension ntrees(1),dbh(1),area(nyears,1)
      data pi/3.141592654/

      do 10 k= 1,ns
            nk= ntrees(k)
            area(kyr,k)= 0.
            if (nk.eq.0) go to 10
            if(k .eq. 1) then
                 jj = 0
            else
                 jj= isum(ntrees,k-1)
            endif
            do 20 j= 1,nk
                 area(kyr,k)= area(kyr,k)+dbh(j+jj)**2.0
   20       continue

C ..... Compute the basal area for the plot in meters square
         area(kyr,k)=area(kyr,k)*pi/(4.*pltsiz)
   10 continue
      return
      END
      SUBROUTINE beetle(SPP,DIA,AGE,PROB)
C *********************************************************************
C This subroutine simulates tree mortality if tree is infested with  *
C bark beetles.  The current functions are from data collected from  *
C the gallatin by Region 1 personnel - contact Ammens Ogden          *
C *********************************************************************
      character spp*4

C ..... Compute the probability of mortality by species
      if(spp .eq. 'pial') then
            prob = ((0.7664 * dia) - 0.2222) / 100.0
            if(prob .lt. 0.0) prob = 0.0
            if(prob .gt. 1.0) prob = 1.0
      elseif(spp .eq. 'pico') then
            if(dia .lt. 46.0 .and. age .lt. 150.0) then
                 prob = (0.555 * dia) / 100.0
            else
                 prob = 0.10
            endif
      elseif(spp .eq. 'pipo' .or. spp .eq. 'pimo') then
            if(dia .lt. 46.0) then
                 prob = (0.555 * dia) / 100.0
            else
                 prob = 0.10
```

```
         endif
      else
         prob = 1.0
      endif
      return
      END
      SUBROUTINE birth(ntrees,dbh,age,ds,agnw,table,nyears,kyr,duff,
     &                 irun,itop,dsw,ccrop,lag,iseed,ptree)
C :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C  This subroutine adds new trees to plot based on climatic constraints
C  (degree-days,available water,cone crop) and site factors (shading and
C  duff depth).  Tree incursion is at 8 years and 1 cm dbh.
C
C     spm(j) ..... max number of new seedlings per meter for all species
C     fnj(j) ..... number of seedlings established onsite.
C     psur   ..... percent survival calculated from duff depth (Boyce 86)
C     dsw    ..... distance to seed wall in meters
C :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      real dbh(1),ds(1),agnw(1),age(1),ptree(7)
      real u(2),dsbar(7),sigds(7),sregen(7),dsw(7)
      integer clrcut,table(nyears,1),ntrees(1),itop(1)
      character*1 ishade,imoist,spp*4
      common/oper/ ns,nspan,nruns,clrcut,nwrstr,ifire,sburn,ibr,impb
      common/limits/ mxtrs,maxspc,mxdd,mxyrs,maxbin
      common/water/grws(7),ws0(7),wsm(7),nws(7),wr,grdd(7),grbar(7)
      common/leaf/aside(7),c(7),alpha(7),b2(7),b3(7),cext(8),crat(7),
     &            sigma(7),ap(7),betap(7)
      common/trunk/g(7),agemx(7),dm(7),hm(7),spm(8),ysc(7)
      common/climat/ dmin(7),dopt(7),dmax(7),baset(12),basep(12),baseh
      common/hdata/phi,xmbar(8),degd,ainc(7),binfest(8),ibcycle(8),brr
      common/birthk/sura(7),surb(7),dbulk(8,2),disequ(2,7),rdelay(8)
      common/plotq/elev,rock,till,soilm,text,excess,pltsiz,ifg
      common/types/ishade(7),imoist(7),spp(7)
      data dsbar/7*.5/,sigds/7*0.1/,no/0/

C ..... Initialize important variables
      if(kyr .eq. 1) duff = 1.5
      do 5 i = 1,ns
         sregen(i) = 0.0
    5 continue
      cones = 0.0
      seedlng = spm(ifg)
      fnjsum = 0.0
      dred = 1.0

C ..... Delay regeneration for interval based on fire group
      if(lag .gt. kyr) then
         go to 200
      endif

C ..... Start the regeneration process for each species
      do 100 j= 1,ns
         nj= 0
         sla= 0.

C ..... Climatic and cone crop tests
         if (table(kyr,j) .eq. no)  go to 100
         if (degd .lt. dmin(j))     go to 100
         if (degd .gt. dmax(j))     go to 100
         if (grws(j) .le. 0.0)      go to 100
```

```
          if (wr .lt. ws0(j))        go to 100

C .....Seedlings to be established.  Reduction factors now calculated
          psur = (sura(j) - surb(j) * duff) / sura(j)
          if(psur .lt. 0.0) psur = 0.0
          totsla = 0.0
          totree = 0.0
          do 10 ii = 1,ns
               if(lag+ysc(ii) .lt. kyr) iseed = 0
               if(iseed .eq. 0) ptree(ii) = 0.0
   10     continue
          do 20 kk= 1, ns
               nkk= ntrees(kk)
               if (nkk.eq.0) go to 20
               kkkk = kk - 1
               kkk = kk
               if(kk .eq. 1) then
                    ikk = 0
               else
                    ikk= isum(ntrees,kkkk)
               endif
               call needle(sla,ikk,nkk,dbh,kkk,wgt,pltsiz)
               totsla = totsla + sla

C ..... Calculation of proportion of seed trees
               do 15 ii = 1,nkk
                    if((dbh(ii+ikk) .gt. 10.0 .or. age(ii+ikk) .ge.
     &              ysc(kk)) .and. iseed .eq. 0) then
                         ptree(kk) = ptree(kk) + 1.0
                         totree = totree + 1.0
                    endif
   15          continue
   20     continue

C ..... Adjustment for off-site seeding, and then the seedling equation
          if(iseed .eq. 0) then
               if(totree .gt. 0.0)    ptree(j) = ptree(j) / totree
               if(totree .le. 0.0)    ptree(j) = 0.05
               if(ptree(j) .lt. 0.05) ptree(j) = 0.05
          endif

C ..... Calculation of whitebark pine seedlings if species present
          if(spp(j) .eq. 'pial') then
               call pinalb(fnj,totsla,dbh,age,ntrees,itop,ccrop,cones)
               seedlng = seedlng - (fnj / pltsiz)
               if(seedlng .le. 0.0) seedlng = 0.1
          endif

C ..... Reduction of seedling due to distance from seed source
   30     dred = 1.0
          if(dsw(j) .gt. 20) then
               if(ptree(j) .le. 0.01) then
                    if(spp(j) .eq. 'pial') then
                         xdist = dsw(j) - 20.0
                         if(xdist .le. 0.0) xdist = 0.0
                         dred = 10.0**(-0.8062-(0.000454*xdist)) /
     &                          0.1563
                         if(dred .lt. 0.0001) dred = 0.0001
                    else
                         xdist = (dsw(j) - 20.0) * 3.2808
```

```
                          xmax = abs(disequ(1,j) / disequ(2,j)) / 3.28
                          if(xdist .le. 0.0) xdist = 0.0
                          if(xmax .gt. xdist) then
                               dred = exp(disequ(1,j) + disequ(2,j)
     &                                * xdist) / exp(disequ(1,j))
                               if(dred .lt. 0.00001) dred = 0.00001
                          else
                               dred = 0.00001
                          endif
                     endif
                endif
           endif

C ..... Calculation of number of seedlings for species j
           if(spp(j) .ne. 'pial') then
                fnj = seedlng * pltsiz * psur * ptree(j) * dred
           else
                fnj = seedlng * dred
           endif

C ..... Reduction for shading effects by tolerance class
         if(ishade(j) .eq. 'I') fnj=fnj * exp(-0.8*totsla)
         if(ishade(j) .eq. 'M') fnj=fnj * exp(-0.25*(totsla+1.0))
         if(ishade(j) .eq. 'T') fnj=fnj * (1.0-exp(-0.25*(totsla+0.2)))

C ..... Final calculation of number established seedlings
           ntot= isum(ntrees,ns)
           xred = (1.0 - (float(ntot) / float(mxtrs)))
           if(fnj .gt. 100.0) fnj = 100.0
           fnj = fnj * xred
           sregen(j) = fnj
           fnjsum = fnjsum + fnj
           nj= int(fnj+.5)
           if(nj.eq.0) go to 100

C ..... Check to see if number of trees has not exceeded maximum
           if (ntot+nj.gt.mxtrs) call error(9)

C ..... Calculate a random diameter for each of the nj seedlings
          do 40 i= 1,nj
            hsbar= b2(j)*dsbar(j)-b3(j)*dsbar(j)**2-b3(j)*sigds(j)**2
            sighs= (b2(j)-2.*b3(j)*dsbar(j))*sigds(j)
   45       call rgen(u,2)
            u(1)= 2.*u(1)-1.
            u(2)= 2.*u(2)-1.
            s= u(1)**2+u(2)**2
            if (s.ge.1) go to 45
            z= u(1)*sqrt(-2.*alog(s)/s)
            hs= sighs*z+hsbar
            if (hs.lt.0.) go to 45
            ds(i)= (b2(j)/(2.*b3(j)))*
     &             (1.-sqrt(1.-4.*(b3(j)/b2(j)**2)*hs))
   40     continue

C ..... Place the seedling in appropriate cell in DBH and AGE array.
          ijj = j
            call add(dbh,ntrees,ds,nj,ijj)
            do 50 k= 1,nj
                 agnw(k)= rdelay(ifg)
   50       continue
```

```
           call add(age,ntrees,agnw,nj,ijj)

C ..... Put zero into the blister rust array ITOP
           n= isum(ntrees,ns)
           kk= isum(ntrees,ijj)
           nkk= n-kk
           if (nkk.eq.0) go to 65
           do 60 i= 1,nkk
                itop(n+nj-i+1)= itop(n-i+1)
   60      continue
   65      do 70 i= 1,nj
                itop(kk+i)= 0
                if(spp(j) .eq. 'pial' .or. spp(j) .eq. 'pimo') then
                     rnum = rnd()
                     if(rnum .lt. brr) itop(kk+i) = 2
                endif
   70      continue
           ntrees(j)= ntrees(j)+nj
  100 continue

C ..... Writing important regeneration variable values to external file
  200  if(irun .eq. 1) then
          write(12,1000) duff,totsla,fnjsum,(sregen(i),i=1,ns),cones
 1000     format(20f8.2)
      endif
      return
      END
      SUBROUTINE brnoff(ln,dn,wood)
C ..............................................................
C     compute litter and duff and woody fuel burnoff on stand       .
C     based on equations in Brown and others (1985).                .
C     All litter is burned off and then fractions of duff, and the  .
C     three fuel types are also burned off.                         .
C     all loadings in units of kilograms per square meters          .
C     wood(3)...... kg/m2 for each fuel type 1,10,100 hr.           .
C     dn(k) ....... biomass loading of k'th duff component          .
C     ln(k) ....... biomass loading of k'th litter component        .
C     pduff........ fract by which amount of duff decreases         .
C ..............................................................
      common/oper/ ns,nspan,nruns,clrcut,nwrstr,ifire,sburn,ibr,impb
      common/polut/ndyr(7),dmoist
      real ln(1),dn(1),pduff,wood(3)
      integer clrcut
      data da/83.70/,db/-0.426/

C...................
C     compute total loading and average moisture content
C...................
      pduff = (da+db*dmoist*100.0)/100.0
      if(pduff .lt. 0.0) pduff = 0.0
      do 10 i = 1,ns
          ln(i) = 0.0
          dn(i) = dn(i)*pduff
   10 continue
C.....................
C  calculation of woody fuel reductions from brown et al equations
C.....................
      do 20 i = 1,3
          if(i .le. 2) then
               conwood = 0.890 * wood(i)
```

```
           else
               conwood = 0.845 * wood(i)
           endif
           if(conwood .lt. 0.0) conwood = 0.0
           wood(i) = wood(i) - conwood
   20 continue
      return
      END
      SUBROUTINE brush(dbh,ntrees,bbm1,bbm2,init)
C :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C This subroutine computes shrub and herbaceous fuel loadings     :
C for the simulation plot.  The carrying capacity formula uses:   :
C x01,x1,x02,x2 indicate live brushy fuel both tol and intol      :
C g01,g1,g02,g2 indicate live grass and forb fuel both tol-intol  :
C b,dd,a,cc are coefficients to the biomass equation.
C :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      common/leaf/aside(7),c(7),alpha(7),b2(7),b3(7),cext(8),crat(7),
     &            sigma(7),ap(7),betap(7)
      common/plotq/elev,rock,till,soilm,text,excess,pltsiz,ifg
      common/trunk/g(7),agemx(7),dm(7),hm(7),spm(8),ysc(7)
      common/hdata/phi,xmbar(8),degd,ainc(7),binfest(8),ibcycle(8),brr
      common/oper/ ns,nspan,nruns,clrcut,nwrstr,ifire,sburn,ibr,impb
      integer clrcut
      dimension ntrees(1),dbh(1),b(8),dd(8)
      integer yes,no
      data yes/1/,no/0/
      data b /0.02700,0.04600,0.08605,0.07600,0.06900,0.07000,
     &        0.01598,0.05437/
      data dd /0.02934,0.1190,0.04816,0.04300,0.10125,0.10143,
     &         0.14224,0.19690/
      data x01,x02,g01,g02 /0.0137,0.0137,0.0010,0.0010/
      data a/1.1398/,cc/10.8644/

C ..... Inline function statements for the carrying capacity formula
      r1(y)= 1.-exp(-2.32*(y-.05+abs(y-.05)))
      r2(y)= 2.24*(1.-exp(-.568*(y-.08+abs(y-.08))))
      delta(y)= y * a * (1.0 - (y/b(ifg)))
      gdelta(y) = y * cc * (1.0 - (y/dd(ifg)))

C ..... Setting initial conditions after a fire of any intensity
      if (init.eq.yes) then
           x1= x1 * (1.0 - sburn)
           x2= x2 * (1.0 - sburn)
           if(x1 .eq. 0.0) x1 = x01
           if(x2 .eq. 0.0) x2 = x02
           g1 = g01
           g2 = g02
      endif
      if (ns.eq.0) return

C ..... Summing up all trees and calculating leaf area index for stand
      sla= 0.
      do 10 k= 1,ns
        kk = k
        nk= ntrees(k)
        if (nk.eq.0) go to 10
        if(kk .eq. 1) then
             jj = 0
        else
             jj= isum(ntrees,kk-1)
```

```
        endif
             call needle(tla,jj,nk,dbh,kk,wgt,pltsiz)
             sla = sla + tla
   10 continue

C ..... Calculating all biomass reduction factors for shading effects
      al= phi*exp(-cext(ifg)*sla)

C ..... Calculating current biomass on the simulation plot
      x1 = x1 + r1(al) * delta(x1)
      x2 = x2 + r2(al) * delta(x2)
      g1 = g1 + r1(al) * gdelta(g1)
      g2 = g2 + r2(al) * gdelta(g2)
      if(g1 .gt. dd(ifg)) g1 = dd(ifg)
      if(g2 .gt. dd(ifg)) g2 = dd(ifg)
      bbm1= (x1+x2)/2.
      bbm2= (g1+g2)/2.
      init= no
      return
      END
      SUBROUTINE calcnt
C ***********************************************************************
C * Subroutine calcnt calculates all parameters for growth equation  *
C * from data in external file TREE.DAT.  Intermediate values are     *
C * first calculated based on maximum height, age, and diameter.      *
C * Calculated parameters are then printed in external file OUTPUT.  *
C ***********************************************************************

      common/leaf/aside(7),c(7),alpha(7),b2(7),b3(7),cext(8),crat(7),
     &            sigma(7),ap(7),betap(7)
      common/trunk/g(7),agemx(7),dm(7),hm(7),spm(8),ysc(7)
      common/plotq/elev,rock,till,soilm,text,excess,pltsiz,ifg
      common/oper/ ns,nspan,nruns,clrcut,nwrstr,ifire,sburn,ibr,impb
      integer clrcut
      character name*10

C ..... Calculation of intermediate terms in growth equation
      do 10 j=1,ns
           a= 1.-137./hm(j)
           term1= alog(2.*(2.*dm(j)-1.))
           term2= (a/2.)*alog((9./4.+a/2.)/(4.*dm(j)**2+2.*a*dm(j)-a))
           term3= (a+a**2/2.)/sqrt(a**2+4.*a)
           term4= 3.+a-sqrt(a**2+4.*a)
           term5= 4.*dm(j)+a+sqrt(a**2+4.*a)
           term6= 4.*dm(j)+a-sqrt(a**2+4.*a)
           term7= 4.*hm(j)/agemx(j)
           term8= 3.+a+sqrt(a**2+4.*a)
           g(j)= term7*(term1+term2-term3*alog((term4*term5)/
     1          (term6*term8)))
           b2(j)= 2.*(hm(j)-137.)/dm(j)
           b3(j)= (hm(j)-137.)/dm(j)**2
   10 continue

C ..... Writing intermediate results to external files
      write(5,1000)
      name= 'g         '
      write(5,2000) name, (g(j),j=1,ns)
      name= 'b2        '
      write(5,2000) name, (b2(j),j=1,ns)
      name= 'b3        '
```

```
      write(5,2000) name, (b3(j),j=1,ns)
      name= 'c         '
      write(5,3000) name, (c(j),j=1,ns)
      return

C ##################### FORMATS #####################
 1000 format(/1h ,32x,'*derived constants*',/)
 2000 format(1h ,a10,7f10.3)
 3000 format(1h ,a10,7f10.7)
      END
      SUBROUTINE cntrl(dsize,dimax,dimin)
C ****************************************************************************
C  This subroutine:                                                          *
C     reads operating parameters and initial distribution of tree diameters. *
C  Variables are:                                                            *
C     nsum......... total number of trees initially on the stand             *
C     nspan........ number of years per run of the model                     *
C     nruns........ number of runs of the model                              *
C     nbins........ number of diameter cohorts for inital state              *
C     width........ width in cm for each diameter cohort                     *
C     dbh0(j)...... vector of initial tree diameters in cm                   *
C     ntrees0(j)... number of trees initially in the j'th species            *
C     ncount(i,j).. number trees init in i'th diam cohort for j'th species   *
C ****************************************************************************
      common/oper/ ns,nspan,nruns,clrcut,nwrstr,ifire,sburn,ibr,impb
      common/init/ ntrees0(7),dbh0(4000),ncount(20,7),nbins,width
     1  ,age0(4000),agein(20,7)
      common/limits/ mxtrs,maxspc,mxdd,mxyrs,maxbin
      real dsize(7)
      integer clrcut,yes,agein
      character*10 name
      data yes/1/

      open(unit=4,file='CONTRL.DAT',form='formatted',recl=150,
     &     pad='yes')

C ..... Reading in simulation specifics, then writing the input to file
      read(4,1000)  name,nspan
      write(5,1000) name,nspan
      read(4,1000)  name,nruns
      write(5,1000) name,nruns
      read(4,1000)  name,clrcut,dimax,dimin
      write(5,1000) name,clrcut,dimax,dimin
      read(4,1000)  name,ifire
      write(5,1000) name,ifire
      read(4,1000)  name,ibr
      write(5,1000) name,ibr
      read(4,1000)  name,impb
      write(5,1000) name,impb
      read (4,4000) name,(dsize(j),j=1,ns)
      write(5,4000) name,(dsize(j),j=1,ns)
      read(4,1000)  name,nwrstr
      write(5,1000) name,nwrstr
      read(4,1000)  name,nbins
      write(5,1000) name,nbins
      read(4,2000)  name,width
      write(5,2000) name,width
C     if (nbins.gt.maxbin) call error(4)
C     if (nspan.gt.mxyrs)  call error(5)
      do 10 i= 1,nbins
```

```
           read(4,3000)  name,(ncount(i,j),j=1,ns)
           write(5,3000) name,(ncount(i,j),j=1,ns)
   10 continue
        do 20 i=1,nbins
             read(4,3000) name, (agein(i,j),j=1,ns)
             write(5,3000)name, (agein(i,j),j=1,ns)
   20   continue
      nsum= 0
      do 30 i= 1,nbins
           do 40 j= 1,ns
                nsum= nsum+ncount(i,j)
   40      continue
   30 continue
      if (nsum.gt.mxtrs) call error(6)
      rewind 4
      close(4)
      return

C #################### Formats ####################
 1000 format(a10,i6,f10.2,f10.2)
 2000 format(a10,f6.1)
 3000 format(a10,7i6)
 4000 format(a10,7f7.1)
      END
      SUBROUTINE crown(ntrees,dbh,ros,byram,flame,fzone,icwf)
      dimension dbh(1),ntrees(1)
        icwf = 0
        return
      END
      SUBROUTINE cycles(x,n,m,u,p,r)
C ****************************************************************************
C This subroutine assigns cone crop years from species-specific prob-        *
C  abilities for having a good cone crop.  A uniform random number gen-      *
C  erator is used (XRANDOM)  and is called from subroutine RGEN.             *
C Variables used:                                                            *
C     X(i,j): binary array storing appropriate classes of cone crops         *
C     P(i): probability of a good cone crop for species i.                   *
C     R(i): number of years to block after a good cone crop for spp i.       *
C     U(i): temporary storage array.                                         *
C ****************************************************************************
      integer x(n,m),r(7)
      real u(1),u1(1),p(7)
      if (n.eq.0.or.m.eq.0) return

C ..... Initializing cone crop array
      do 10 i= 1,n
        do 20 j= 1,m
          x(i,j)= 0
   20   continue
   10 continue

C ..... Calculating probabilities for blocked and unblocked states
      do 50 j= 1,m
           i= 0
           if (r(j).eq.1) go to 30

C ..... Calculate pnb, prob of an ublocked state
           pnb= 1./(p(j)*float(r(j)-1))
           call rgen(u1,1)
           if (u1(1).le.pnb) go to 30
```

```
C ..... Select an integer b at random from 1,2,3,... r-1
          call rgen(u1,1)
          i= int(float(r(j)-1)*u1(1))+1
   30     call rgen(u,n)
   40     i= i+1
          if (i.gt.n) go to 50
          if (u(i).gt.p(j)) go to 40
          x(i,j)= 1
          i= i+r(j)-1
          go to 40
   50 continue
      return
      END
      SUBROUTINE dist(u,itop)
C ****************************************************************************
C  This subroutine calculates initial distribution of tree diameters.        *
C   - if clearcut option is specified set inital diameter vector to zero.    *
C     Trees are distributed randomly (ie. uniform pdf) within a diam cohort *
C     Variables are:                                                         *
C      nbins........ number of diameter cohorts for inital state             *
C      width........ width in cm for each diameter cohort                    *
C      dbh0(j)...... vector of initial tree diameters in cm                  *
C      ntrees0(j)... number of trees initially in the j'th species           *
C      ncount(i,j).. number trees init in i'th diam cohort, j'th species     *
C      clrcut....... flag to specify clear cut option                        *
C ****************************************************************************
      common/oper/ ns,nspan,nruns,clrcut,nwrstr,ifire,sburn,ibr,impb
      common/init/ ntrees0(7),dbh0(4000),ncount(20,7),nbins,width,
     &             age0(4000),agein(20,7)
      common/limits/ mxtrs,maxspc,mxdd,mxyrs,maxbin
      integer clrcut,yes,agein
      dimension u(1),itop(1)
      data yes/1/

C ..... Initialize appropriate arrays
      do 5 i= 1,ns
          ntrees0(i)= 0
    5 continue

      do 8 i= 1,mxtrs
          dbh0(i)= 0.0
          age0(i)= 0.0
          itop(i) = 0
    8 continue

C ..... Assign each tree diameter and age to appropriate cell in each
C .....  array (DBH and AGE)
      kk= 0
      do 10 j= 1,ns
        do 20 i= 1,nbins
             n= ncount(i,j)
             ntrees0(j)= ntrees0(j)+n
             if (n.eq.0) go to 20
             call rgen(u,n)
             do 30 k= 1,n
                  dbh0(kk+k)= width*(u(k)+i-1)
                  age0(kk+k)= float(agein(i,j))
   30        continue
             kk= kk+n
```

```
   20   continue
   10 continue
      return
      END
      SUBROUTINE error(fmt)
C ***************************************************************************
C  This subroutine terminates program and send message to terminal.        *
C ***************************************************************************
      integer fmt
      character*50 msg(12)
      data msg /' N1 is greater than N2 .                           ',
     1          ' N1 is greater than MID.                           ',
     2          ' MID is greater than N2.                           ',
     3          ' Too many diameter cohorts.                        ',
     4          ' Time span is too large, redo control file.        ',
     5          ' Initial distribution has too many trees.          ',
     6          ' Too many species in TREDAT, redo file.            ',
     7          ' No end-of-species marker, fix TREDAT file.        ',
     8          ' Too many trees in BIRTH.                          ',
     9          ' Too many dead trees in KILL.                      ',
     1          ' DINC is greater than 5.0 cm, abnormal.            ',
     2          '                                                   '/

C ..... Print appropriate error message
      write(5,1000) msg(fmt)
 1000 format(/1H ,a50)
      stop
      END
      SUBROUTINE fire(ntrees,dbh,fwg,nl,p,yr,duff,branch,wood,
     &                irun,icwf,dimax,dimin)
C *****************************************************************************
C This subroutine is a sub-driver for all components used to calculate fire *
C   intensity.  Subroutine logic is as follows:                               *
C     1.  Update fuel loadings: call subroutine FUEL                          *
C     2.  Compute if current simulation year is a fire year, if not RETURN   *
C     3.  Assign fuel loadings into appropriate array TFWG(i,j).              *
C     4.  Compute fire intensity: call FIREMOD.                               *
C     5.  Compute scorch height and resultant tree mortality: call INJURY    *
C     6.  Compute fuel comsumption: call BRNOFF                               *
C     7.  Compute duff and litter depth.                                      *
C   Important variables are:                                                  *
C     TFWG(i,j)- fuel loadings for live and dead fuel components,             *
C     DUFF- duff and litter depth in cms,                                     *
C   Computed intensity and scorch height are written to external file.        *
C *****************************************************************************
      common/oper/ ns,nspan,nruns,clrcut,nwrstr,ifire,sburn,ibr,impb
      common/leaf/aside(7),c(7),alpha(7),b2(7),b3(7),cext(8),crat(7),
     &           sigma(7),ap(7),betap(7)
      common/plotq/elev,rock,till,soilm,text,excess,pltsiz,ifg
      common/polut/ndyr(7),dmoist
      common/birthk/sura(7),surb(7),dbulk(8,2),disequ(2,7),rdelay(8)
      common/sites/ occur(500),rh,wind,ttheta,t
      common/fuel1/ mext(2),rhop(2,7),tbulk(2,8),mois(2,7)
      common/fuel2/ mps(2,7),lhv(2,7),st(2,7),se(2,7)
      common/fuel5/ amc(7),bmc(7),cmc(7),dmc(7),mmc(7),tmc,emc(7)
      common/mort/ d1(7),d2(7),d3(7),bc(7)
      dimension ntrees(1),dbh(1),p(1),fwg(2,7),tfwg(2,7)
      real ln(7),ln1(7),dn(7),dn1(7),wood(3),lw,mois,hs
      integer flag(3),yr,yes,no,occur,clrcut
      data yes/1/,no/0/
```

```
      data lnl/7*0./,dnl/7*0./
      data init/1/

C ..... Initialization of parameters
      icwf = 0
      duff1 = 0.0
      duff2 = 0.0
      lw = 0.0
      n= isum(ntrees,ns)
      if (n.eq.0) return

C ..... Decide if current year is a clearcut year
      if(clrcut .eq. yr) then
           do 10 i = 1,n
                if(dbh(i) .ge. dimin .and. dbh(i) .le.. dimax) then
                     p(i) = 0.99999
                endif
   10      continue
           go to 30
      endif

C ..... Update fuel components, including litter and duff.
      call fuel(ntrees,dbh,fwg,ln,lnl,dn,dnl,wood,yr,init,irun,
     &          branch,icwf)

C ..... Decide if current year is a fire year.
      if (occur(yr).eq.no) go to 30

C ..... Putting the five dead fuel types into temporary array
C .....  tfwg(i,j) types are litter, 1 hour, 10, and 100 hour woody,
C .....  and cured grass and last dead shrub.
      do 15 i = 1,ns
           lw = lw + ln(i)
           tfwg(1,i) = 0.0
   15 continue
      tfwg(1,1) = lw
      do 16 i = 1,3
           tfwg(1,i+1) = wood(i)
   16 continue
      tfwg(1,5) = fwg(1,5)
      tfwg(1,6) = fwg(1,6)
      tfwg(1,7) = 0.0
      do 17 i = 1,2
           tfwg(2,i) = fwg(2,i)
   17 continue

C ..... Simulating a fire by calling FIREMOD
      call firemd(nl,tfwg,byram,flag,ifg,rate,flame,fzone)

C ..... Computing crown fire initiation
      call crown(ntrees,dbh,rate,byram,flame,fzone,icwf)

C ..... Calculating scorch height and tree mortality
      call injury(ntrees,dbh,byram,p,hs,icwf)

C ..... Computing fuel reduction or consumption
      call brnoff(ln,dn,wood)
      init= yes

C ..... Writing fire intensity and scorch height to file
```

```
      if(irun .eq. 1) then
           write(11,1000) yr,byram,hs,flame,rate
 1000      format(I4,7f10.4)
      endif

C ..... Calculating the depth of the duff layer from duff and litter
C ..... components LN and DN.

   30 do 40 i=1,ns
           duff1 = duff1 + ln(i)
           duff2 = duff2 + dn(i)
   40 continue
C ..... Computation of duff depth from duff bulk density
      duff1 =  (duff1/dbulk(ifg,1))*100.0
      duff2 =  (duff2/dbulk(ifg,2))*100.0
      duff = duff1 + duff2
      return
      END
      SUBROUTINE firemd(nl,fwg,byram,flag,ifg,rate,flame,fzone)
C     ***************************************************************
C     *                                                              *
C     *   metric version of original (nov. 1973) SUBROUTINE firemd     *
C     *   -- units are converted on input and reconverted on output    *
C     *   but internal computation is expressed in british units --    *
C     *                                                              *
C     *   conversion factors are stored in array named  * cio *        *
C     *   factor      value      converts      from        to          *
C     *   ......     .......     ........      ....       ....         *
C     *   cio(1)     .032808      sigma       1/ft       1/cm          *
C     *   cio(2)     .18915    xir,ir,xio  btu/sqft/min kw/sqm         *
C     *   cio(3)     37.259      rhobqig     btu/cuft  kj/cu m         *
C     *   cio(4)     1.60934     wind....      mi/h      km/h          *
C     *   cio(5)     .3048     ratex,rate     ft/min     m/min         *
C     *   cio(6)     3.4592    byramx,byram  btu/ft/s    kw/m          *
C     *   cio(7)     4.8824         fwg      lb/sq ft   kg/sq m        *
C     *   cio(8)     2.3244         lhv       btu/lb     kj/kg         *
C     *   cio(9)     .016018        rhop     lb/cu ft    g/cc          *
C     *                                                              *
C     *                                                              *
C     *   variables used in this SUBROUTINE (written in fortran - iv)  *
C     *   are identified below. the rate-of-spread model employed is   *
C     *   documented in usda forest service research paper int-115,    *
C     *   a mathematical model for predicting fire spread in wildland  *
C     *   fuels, r. rothermel (northern forest fire lab., missoula),   *
C     *   but excluding the **effective heating number** revision of   *
C     *   w.h. frandsen suggested in usda forest service general tech- *
C     *   nical report int-10, 1973. the calculation of byrams inten-  *
C     *   sity (btu/min/ft of fireline length) is based on the crude   *
C     *   approximation that the burning zone produces a uniform rate  *
C     *   of heat output from front to back and that the depth of the  *
C     *   flame is determined by the burning time of the gross descrip-*
C     *   tive mean particle diameter 4/sigma.                         *
C     *   significant revisions include.....                           *
C     *       a new way of computing the moisture of extinction of     *
C     *       live fuels, including 1) exponential weighting of size   *
C     *       classes to get fine dead/live ratio and 2) using mext(1) *
C     *       in place of the constant 0.3 in the equation for mext(2) *
C     *       use of a new weighting factor, g(i,j), in place of       *
C     *       f(i,j) in computing net effective loading by size class  *
```

```
C     *       use of a power law formula for the reaction velocity     *
C     *       correlation parameter *a*                                *
C     *       elimination of weighting factors on reaction intensity   *
C     *       of categories  (eff. heating no. or f(i) )               *
C     *   programmed nov. 1973 by f. albini, nffl, missoula.           *
C     *                                                                *
C     *   input variables...first the physical variables               *
C     *                                                                *
C     *   symbol    pg.no./eq.no.           definition                 *
C     *             in int-115                                         *
C     *   mext(1)...31/65          moisture of extinction of dead fuel *
C     *   ttheta....33/80          tangent of local slope              *
C     *   mois(i,j).31/66          moisture content of fuel type (i,j) *
C     *   mps(i,j)..30/53,32/72    mean surf/vol, 1/ft, of fuel (i,j)  *
C     *   fwg(i,j)..31/60,32/73,74 surface loading, lb/sqft fuel (i,j) *
C     *   lhv(i,j)..31/61          low heat value, btu/lb, fuel (i,j)  *
C     *   rhop(i,j).30/53,32/73    ovendry particle density, lb/cuft   *
C     *   st(i,j)...31/60          mineral content of fuel type (i,j)  *
C     *   se(i,j)...31/63          mineral content excluding silica    *
C     *   wind....                 wind speed at mid flame height (mph)*
C     *                                                                *
C     *   input variables...program control and specification variables*
C     *                                                                *
C     *   symbol    size range              description                *
C     *                                                                *
C     *   nd........ 0 - 7    number of dead fuel size classes to be   *
C     *                       considered (specifies largest class if  *
C     *                       there are more classes than nd)          *
C     *   nl........ 0 - 7    same as nd, but for live fuels           *
C     *   ifines(1). 1 - nd   ordinal number of smallest-size dead fuel*
C     *                       to be used in computation                *
C     *   ifines(2). 1 - nl   same as ifines(1) but for live fuels     *
C     *  large1    largest dead fuel size class to be included         *
C     *  large2    largest live fuel size class to be included         *
C     *                                                                *
C     *                                                                *
C     *                  .....output variables.....                    *
C     *                                                                *
C     *                                                                *
C     *   symbol                        definition                     *
C     *                                                                *
C     *   flag( ) . array of error flags,set to 1 for error            *
C     *             flag(1) dead fuel too moist too spread flame       *
C     *             flag(2) wind speed exceeds reliable extrapolation  *
C     *             flag(3) gross surf/vol too small (sigma.lt.175)    *
C     *   betal.....mean packing ratio (pg 32/eq 73)                   *
C     *   sigma.....characteristic surface area to volume ratio of the *
C     *             fuel complex, 1/ft (pg 32/eq 71)                   *
C     *   gamma.....reaction velocity, 1/min (pg 31/eq 67)             *
C     *   xir.......reaction intensity, btu/min/sqft, calculated from  *
C     *             eq 58, pg 31, but with area-weighting factor, f-sub*
C     *             -i replaced by unity....no category weighting      *
C     *   rhobqig...heat sink term -product of bulk density, effective *
C     *             heating number, and heat of preignition- btu/cuft  *
C     *             (pg 32/eq 77)                                      *
C     *   phis......slope factor modifying spread rate (pg 33/eq 80)   *
C     *   windx.....wind speed which produces maximum spread rate, mph *
C     *             (pg 33/eq 86)                                      *
C     *   phiwx.....maximum value of wind factor (pg 33/eq 80,87)      *
C     *   ratex.....maximum wind-driven rate of spread, ft/min (pg 32/ *
```

```
C     *             eq 75, with phi-sub-w of eq 79 at u=0.9*i-sub-r)   *
C     *   byramx....byrams intensity, btu/min/ft of fireline length,   *
C     *             at the rate of spread = ratex (near statement 30)  *
C     *   ir(i).....reaction intensity, btu/min/sqft, for dead (i=1)   *
C     *             or live (i=2) fuel type -components of xir         *
C     *   mext(2)...moisture of extinction of live fuel (pg 35/eq 88)  *
C     *   **n.b.- mext(1), for dead fuel, is an input parameter        *
C     *   byram.....byrams intensity, btu/min/ft of fireline length,   *
C     *             for wind speed corresponding to index j (near 33)  *
C     *   rate......spread rate, ft/min, for wind speed (pg 32/eq 75)  *
C     *   flame ....flame lenght in meters p86, eq 17
C     *                                                                *
C     *   working variables.....internal to SUBROUTINE                 *
C     *      -index i refers to fuel category (1=dead, 2=live)         *
C     *      -index j refers to (size) class within category (j.le.100)*
C     *                                                                *
C     *   symbol                        definition                     *
C     *                                                                *
C     *   ai(i).....fuel surface area/sqft of ground (pg 30/eq 54)     *
C     *   a(i,j)....fuel surface area/sqft of ground (pg 30/eq 53)     *
C     *   wo(i,j)...net dry fuel loading, lb/sqft (pg 31/eq 60)        *
C     *   f(i,j)....weighting factor (pg 30/eq 56)                     *
C     *   g(i,j)....weighting factor for computing net effective load- *
C     *             ing for each category...replaces weighting factor  *
C     *             f(i,j) used for intrinsic properties (pg 30/eq 56) *
C     *             for loading calculation, size classes are grouped  *
C     *             and weighted uniformly according to contribution to*
C     *             total area by group as a whole...g = aa(n)/ai(i).. *
C     *   aa1.......area of size class 1 (mps.ge.1200)                 *
C     *   aa2.......area of size class 2 (1200.gt.mps.ge.192)          *
C     *   aa3.......area of size class 3 (192.gt.mps.ge.96)            *
C     *   aa4.......area of size class 4 (96.gt.mps.ge.48)             *
C     *   aa5.......area of size class 5 (48.gt.mps.ge.16)             *
C     *   ..........note - fuels with mps .lt. 16 are not used         *
C     *   gs(i,j)...shorthand for exp(-138./mps(i,j))                  *
C     *   at........total fuel surface area/sqft of ground (pg 30/eq55)*
C     *   fx(i).....weighting factor (pg 30/eq 57)                     *
C     *   noclas(i).noclas(1)=nd, noclas(2)=nl. see inputs             *
C     *   isize(i,j)=place no. of jth finest fuel, category i          *
C     *   fined.....dry loading of dead fines, lb/sqft                 *
C     *   finel.....dry loading of live fines, lb/sqft                 *
C     *   wdfmn.....total moisture loading of dead fines, lb/sqft      *
C     *   findm.....average moisture of dead fines (wdfmn/fined)       *
C     *   xmoisl....computed live moisture of extinction (pg 35/eq 88) *
C     *   ax........=f(i,j)                                            *
C     *   qig(i,j)..heat of preignition, btu/lb (pg 32/eq 78)          *
C     *   mcsa(i)...weighted average moisture content (pg 31/eq 66)    *
C     *   bse(i)....weighted average mineral content (pg 31/eq 63)     *
C     *   sigmal(i).characteristic surf/vol ratio (pg 32/eq 72)        *
C     *   lhvl(i)...weighted average low heat value, btu/lb (pg31/eq61)*
C     *   suml......total dry loading, lb/sqft -(see pg 32/eq 74)      *
C     *   sum2......total volumetric loading, ft (see pg 32/eq 73)     *
C     *   wol(i)....weighted average fuel loading, lb/sqft (pg 31/eq59)*
C     *   sum3......sum in heat sink equation, btu/cuft (pg 32/eq 77)  *
C     *   beta......moisture content/moisture of extinction...redefined*
C     *             for each category (pg 31/eq 65)                    *
C     *   mdcsa(i)..moisture damping coefficient (pg 31/eq 64)         *
C     *   barns(i)..mineral damping coefficient (pg 31/eq 64)          *
C     *   sigma.....gross characteristic surf/vol ratio (pg 32/eq 71)  *
C     *   rhopl.....bulk density of fuel complex, lb/cuft (pg 32/eq 74)*
```

```
C     *  best......computed optimum packing ratio (pg 32/eq 69)      *
C     *  rat.......ratio of packing ratio to best (used in eq 67/pg31)*
C     *  al........empirical fit parameter a of eq 70/pg 32          *
C     *            but nondivergent power law used, not eq 70/pg 32  *
C     *  v.........sigma**1.5 used in eq 68/pg 32                    *
C     *  b.........exponent in eqn for propagating flux/reaction in- *
C     *            tensity, xsi, (pg 32/eq 76)                       *
C     *  xml.......parameter b of eq 83/pg 33                        *
C     *  xnl.......parameter e of eq 84/pg 33                        *
C     *  cl........parameter c of eq 82/pg 33                        *
C     *  wmax......maximum effective wind speed, ft/min (pg 33/eq 86) *
C     *  r.........rate of spread, ft/min (pg 32/eq 75)              *
C     *  rmax......maximum wind-driven rate of spread, ft/min        *
C     *                                                              *
C     **********     firemd     ************     firemd     **********
      common/fuel1/ mext(2),rhop(2,7),tbulk(2,8),mois(2,7)
      common/fuel2/ mps(2,7),lhv(2,7),st(2,7),se(2,7)
      common/sites/ occur(500),rh,wind,ttheta,t
      common/oper/ ns,nspan,nruns,clrcut,nwrstr,ifire,sburn,ibr,impb
      real rhop,mext,mois,mps,lhv,st,se,wind,ttheta
      dimension ai(2),bse(2),sigmal(2),wol(2),
     &         a(2,7),f(2,7),fx(2),wo(2,7),qig(2,7),barns(2)
      real betal,sigma,gamma,xir,rhobqig,phis,windx,phiwx,ratex,
     &    lhvl(2)
      real byramx,byram,rate,xio,fwg(2,7),mcsa(2),mdcsa(2),ir(2)
      integer isize(2,7),ifines(2),large1,large2,nl,nd,flag(3),clrcut
      dimension g(2,7),gs(2,7),gn(2),noclas(2),cio(9),
     &         bulk(2,7)
      data cio/.032808,.18915,37.259,1.60934,.3048,3.4592,4.8824,
     &        2.3244,0.016018/

      nd = 6
      large1= nd
      large2= nl
      ifines(1)= 1
      ifines(2)= 1
      do 651 i=1,nl
           do 650 j=1,nd
                mps(i,j)=mps(i,j)/cio(1)
                fwg(i,j)=fwg(i,j)/cio(7)
                lhv(i,j)=lhv(i,j)/cio(8)
                rhop(i,j)=rhop(i,j)/cio(9)
                if(i .eq. 1) bulk(i,j)= tbulk(i,ifg)/cio(9)
                if(i .eq. 2) bulk(i,j) = tbulk(i,j)/cio(9)
 650       continue
 651  continue
      wind = wind/cio(4)
      noclas(1) = nd
      noclas(2) = nl
C.... zero all working arrays and initialize variables
      gamma=0.
      xir=0.
      windx=0.
      phiwx=0.
      ratex=0.
      byramx=0.
      xio=0.
      flag(1)= 0
      flag(2)= 0
      flag(3)= 0
```

```
      mext(2)= 0.
      do 1 i=1,2
           ai(i)=0.
           mcsa(i)=0.
           bse(i)=0.
           sigmal(i)=0.
           lhvl(i)=0.
           wol(i)=0.
           sum4= 0.
           sum1= 0.
           sum2= 0.
           sum3= 0.
           ir(i)= 0.
           barns(i)= 0.
           fx(i)= 0.
           sigma= 0.
           at= 0.
           gn(i) = 0.
           do 1 j=1,7
                isize(i,j)=j
                g(i,j) =0.
                gs(i,j) = 0.
                a(i,j)=0.
                f(i,j)=0.
                wo(i,j)=0.
                qig(i,j)=0.
                byram=0.
                rate =0.
    1 continue
C     sort fuel components by size, finest fuels first
C     isize(i,j) = place no. of jth finest fuel of category i
      do 4 i=1,2
           jmax = noclas(i)
           if(jmax.le.1) go to 4
           jmm = jmax -1
           do 3 j = 1,jmm
                km = jmax - j
                do 2 k=1,km
                     ida=isize(i,k)
                     idb=isize(i,k+1)
                     siza=mps(i,ida)
                     sizb=mps(i,idb)
                     if(siza.ge.sizb) go to 2
                     isize(i,k+1)=ida
                     isize(i,k)=idb
    2           continue
    3      continue
    4 continue
C     delete large logs from firespread considerations
      do 205 i = 1,2
           kmax = noclas(i)
           if(kmax.lt.1) go to 205
           do 202 k = 1,kmax
                j = isize(i,k)
                if((mps(i,j)).ge.16.) go to 202
                noclas(i) = k-1
                go to 205
  202      continue
  205 continue
C     calculate weighting factors
```

```
C     first, for dead fuels....
C     then for live fuels....
      n1 = noclas(1)
      n2 = noclas(2)
      noclas(1) = min0(large1,n1)
      noclas(2) = min0(large2,n2)
      do 7 i = 1,2
           kmin = ifines(i)
           kmax = noclas(i)
           if((kmax.eq.0).or.(kmin.gt.kmax))  go to 7
           do 5 k = kmin,kmax
                j = isize(i,k)
                gs(i,j) = mps(i,j)/rhop(i,j)
                a(i,j) = fwg(i,j)*gs(i,j)
                gs(i,j) = exp(-138./mps(i,j))
                ai(i) = ai(i) + a(i,j)
                wo(i,j) = fwg(i,j)*(1. - st(i,j))
    5      continue
           do 6 k = kmin,kmax
                j = isize(i,k)
                f(i,j) = a(i,j)/ai(i)
   6       continue
   7  continue
      at = ai(1) + ai(2)
      fx(1) = ai(1)/at
      fx(2) = 1. - fx(1)
C.... find weight loading of dead and live fines, moisture extinct. live
C.... note dead and live fuels wtd by exp(-c/sigma) -- c=138 or 500
      fined= 0.0
      finel= 0.0
      wdfmn= 0.0
      findm= 0.0
      do 18 i=1,2
            n=ifines(i)
            jm=noclas(i)
            if((jm.le.0).or.(n.gt.jm))  go to 18
            if(i.eq.2)  go to 15
            do 13 j=n,jm
                jj=isize(i,j)
                sa=mps(i,jj)
                ep =exp(-138./sa)
                wtfac= fwg(i,jj)*ep
                wmfac= wtfac*mois(i,jj)
                fined =fined + wtfac
                wdfmn = wdfmn + wmfac
  13       continue
           if(fined.eq.0.) go to 18
           findm = wdfmn/fined
  15       if(i.eq.1) go to 18
           do 16 j=n,jm
                jj = isize(i,j)
                sa = mps(i,jj)
                ep = exp(-500./sa)
  16            finel = finel + fwg(i,jj)*ep
  18  continue
      if(finel.eq.0.) go to 19
      factor = fined/finel
      xmois1=2.9*factor*(1.-findm/mext(1))-0.226
      if(xmois1.lt.mext(1))  xmois1=mext(1)
      go to 20
```

```
  19  xmoisl=100.
  20  mext(2)=xmoisl
C.... intermediate computations for each category of fuel (live + dead)
      do 22 i=1,2
      aa1 = 0.0
      aa2 = 0.0
      aa3 = 0.0
      aa4 = 0.0
      aa5 = 0.0
      jm=noclas(i)
        n=ifines(i)
      if((jm.eq.0).or.(n.gt.jm)) go to 22
      do 21 k=n,jm
      j=isize(i,k)
        ax=f(i,j)
      sigm = mps(i,j)
      if(sigm.ge.1200.) aa1 = aa1 + a(i,j)
      if((sigm.lt.1200.).and.(sigm.ge.192.)) aa2 = aa2 + a(i,j)
      if((sigm.lt.192.).and.(sigm.ge.96.)) aa3 = aa3 + a(i,j)
      if((sigm.lt.96.).and.(sigm.ge.48.)) aa4 = aa4 + a(i,j)
      if(sigm.lt.48.) aa5 = aa5 + a(i,j)
      qig(i,j)=250. + 1116.*mois(i,j)
      mcsa(i)=mcsa(i) + ax*mois(i,j)
      bse(i)=bse(i) + ax*se(i,j)
      sigmal(i)=sigmal(i) + ax*mps(i,j)
      lhvl(i)=lhvl(i) + ax*lhv(i,j)
      sum4= sum4+bulk(i,j)*fwg(i,j)
      sum1=sum1 + fwg(i,j)
      sum2=sum2 + fwg(i,j)/rhop(i,j)
  21  sum3  =  sum3 + fx(i)*f(i,j)*qig(i,j)*gs(i,j)
      do 221 k = n,jm
      j = isize(i,k)
        sigm = mps(i,j)
      if(sigm.ge.1200.) g(i,j) = aa1/ai(i)
      if((sigm.lt.1200.).and.(sigm.ge.192.)) g(i,j) = aa2/ai(i)
      if((sigm.lt.192.).and.(sigm.ge.96.)) g(i,j) = aa3/ai(i)
      if((sigm.lt.96.).and.(sigm.ge.48.)) g(i,j) = aa4/ai(i)
      if(sigm.lt.48.) g(i,j) = aa5/ai(i)
      wol(i) = wol(i) + g(i,j)*wo(i,j)
 221  continue
      beta = mcsa(i)/mext(i)
      mdcsa(i)=1. - beta*(2.59 - beta*(5.11 - beta*3.52))
      if(mext(i).lt.mcsa(i)) mdcsa(i)=0.
      barns(i)=0.174/(bse(i)**0.19)
      if(barns(i).gt.1.) barns(i)=1.
      sigma=sigma + fx(i)*sigmal(i)
      ir(i) = wol(i)*lhvl(i)*mdcsa(i)*barns(i)
  22  continue
      if (mdcsa(1).le.0) flag(1)= 1
      if (mdcsa(1).le.0.) go to 3777

C.... begin final computations
C.... bulk density....
      rhopl= sum4/sum1

C.... packing ratio
      betal= sum2*rhopl/sum1

C.... optimum packing ratio
      best=3.348/(sigma**.8189)
```

```
      rat=betal/best

C.... new exponent a equation used here
      al=133./(sigma**.7913)

C.... reaction intensity weighted by surface area fraction
      v=sigma**1.5
      gamma=(v*(rat**al)*exp(al*(1.-rat)))/(495. + .0594*v)
      ir(1)=gamma*ir(1)
      ir(2)=gamma*ir(2)
      xir=ir(1)+ir(2)
C.... heat sink terms
      rhobqig=rhopl*sum3
C.... propagating intensity
      b= (.792+.681*sqrt(sigma))*(.1+betal)
      xio =(xir*exp(b))/(192. + .2595*sigma)
C.... slope factor phis
      phis=5.275*ttheta*ttheta/(betal**0.3)
C.... parameters for determining wind factor phiw
      xml=0.02526*(sigma**.54)
      xnl=0.715*exp(-0.000359*sigma)
      cl =7.47*exp(-0.133*(sigma**.55))
      cl = cl/(rat**xnl)
      wmax=0.9*xir
        windx=wmax/88.
         phiwx=cl*(wmax**xml)
      rmax=xio*(1.0 + phis + phiwx)/rhobqig
        ratex=rmax
      byramx=xir*ratex*384./sigma
      w=wind*88.
        phiw=cl*(w**xml)
         r=xio*(1.+phis+phiw)/rhobqig
      rate= r
      byram= xir*r*384./sigma
      fzone = (byram/xir)
      if((w.ne.0.).and.(sigma.lt.175.))  flag(3)=1.0
      if (w.gt.wmax) flag(2)= 1

C     before return to calling program
C     must convert everything to metric here

 3777 continue
      sigma=sigma*cio(1)
       xir=xir*cio(2)
        rhobqig=rhobqig*cio(3)
      windx=windx*cio(4)
       ratex=ratex*cio(5)
        byramx=byramx*cio(6)/60.
      ir(1)=ir(1)*cio(2)
       ir(2)=ir(2)*cio(2)
      do 3778 i=1,nl
      do 3778 j=1,nd
           mps(i,j)=mps(i,j)*cio(1)
           fwg(i,j)=fwg(i,j)*cio(7)
           lhv(i,j)=lhv(i,j)*cio(8)
           rhop(i,j)=rhop(i,j)*cio(9)
           bulk(i,j)= bulk(i,j)*cio(9)
 3778 continue
      flame = 0.45 * (byram/60.0)**(0.46)
      wind= wind*cio(4)
```

```
      bcio=cio(6)/60.
       rcio=cio(5)
      byram=bcio * byram
      rate= rate * rcio
      flame = flame * cio(5)
      fzone = fzone * cio(5)
      xio=xio*cio(2)
      return
      END
      SUBROUTINE FLTEMP(flame,ftmp)
C ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C  This subroutine calculates the average flame temperature of a
C  fire with a specified intensity and rate of spread.  This temp
C  is used in the calculation of heat needed to ignite crown.
C :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

       trate = 1500.0 / flame
       if(trate .gt. 1500.0) trate = 1500.0
       ftmp = 2000.0 - (trate)
       return
      END
      SUBROUTINE foil(pfoil,dbh,kk)
C *********************************************************************
C * Subroutine foil calculates the proportion foliage in the live     *
C * crown using regression equations from Brown (1976).  Equations    *
C * are exponential form except for grand fir and lodgepole pine      *
C * crown portion regression equations.                               *
C *   pfoil - proportion of live foliage in crown.
C *********************************************************************
      common/types/ishade(7),imoist(7),spp(7)
      common/leaf/aside(7),c(7),alpha(7),b2(7),b3(7),cext(8),crat(7),
     &            sigma(7),ap(7),betap(7)
      character*1 ishade,imoist,spp*4

C ..... Calculate the pro. foilage for each individual species
          if(spp(kk) .eq. 'abgr') then
               pfoil = 1.0 / (ap(kk) + betap(kk)*dbh)
          elseif(spp(kk) .eq. 'pico') then
               pfoil = ap(kk) + betap(kk)*dbh
          else
               pfoil = ap(kk)*exp(betap(kk)*dbh)
          endif
      return
      END
      SUBROUTINE fuel(ntrees,dbh,fwg,ln,lnl,dn,dnl,wood,yr,init,irun,
     &                branch,icwf)
C ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C This subroutine:                                                         :
C     calculates moisture content and loading for each fuel component      :
C     mois(1,k) ... fraction moist content of fuel component k             :
C     fwg(1,k) .... biomass loading of fuel component k  kg/sq m           :
C     emc(k) ...... equilibrium moisture content in percent                :
C     bbm ......... brush biomass loading, kg/sq m                         :
C     rh .......... relative humidity in percent                           :
C     t ........... ambient temperature in deg c                           :
C ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      common/fuel1/ mext(2),rhop(2,7),bulk(2,8),mois(2,7)
      common/fuel5/ amc(7),bmc(7),cmc(7),dmc(7),mmc(7),tmc,emc(7)
      common/sites/ occur(500),rh,wind,ttheta,t
      common/plotq/elev,rock,till,soilm,text,excess,pltsiz,ifg
```

```
   common/oper/ ns,nspan,nruns,clrcut,nwrstr,ifire,sburn,ibr,impb
   common/polut/ndyr(7),dmoist
   integer clrcut,yr
   dimension ntrees(1),dbh(1),fwg(2,7),sfuel(8)
   real ln(1),lnl(1),dn(1),dnl(1),wood(3)
   real mois,mext,branch,amc,bmc,cmc,dmc,mmc,tmc,rh,emc
   data sfuel/1.000,1.000,0.717,0.668,0.768,0.768,0.985,0.852/

   flit = 0.0
   if (ns.le.0) return

C ..... Update fuel loadings
   call loader(ntrees,dbh,lnl,ln,dnl,dn,wood,yr,branch,icwf)

C ..... Calculation of moisture content of fuel - defined as EMC
    do 20 k= 1,ns
       if(emc(k) .eq. 0.0) then
          emc(k) = amc(k)*rh**bmc(k)+cmc(k)*exp((rh-100.)/
  &                    dmc(k))+mmc(k)*(tmc-t)
       endif
       flit = flit + ln(k)
       mois(1,k)= emc(k)
 20 continue

C ..... Update fuel loadings for shrubby and herbaceous fuels
   call brush(dbh,ntrees,bbml,bbm2,init)

C #################################################################
C  Putting shrub and grass fuel in appropriate element of fuel
C  array (fwg).  Proportions sfuel(i) for shrubs go into live
C  fuel, and 0.90 for herbaceous go into dead fuels and vice
C  versa.
C #################################################################

   fwg(1,1) = flit
   do 30 i = 1,3
      fwg(1,i+1) = wood(i)
 30 continue
   fwg(1,5) = (1.0 - sfuel(ifg))*bbml
   fwg(1,6) = (bbm2*0.80)
   fwg(2,1)= sfuel(ifg)*bbml
   fwg(2,2)= (bbm2*0.20)

C ..... Writing current fuel values to external files
   if(irun .eq. 1) then
      write(8,1000) (fwg(1,l),l=1,4)
      write(9,1000) (fwg(1,l),l=5,6),(fwg(2,m),m=1,2)
   endif
C ##### FORMATS #####
 1000 format(7f10.3)
   return
   END
   SUBROUTINE grow(dbh,pd,ntrees,sla,grf,s1,s2,s3,age,kyr,itop,
  &                 inend,npine)
C ***************************************************************************
C This subroutines calculates the annual growth increment for each species. *
C Program logic is:                                                         *
C   1.  Compute basal area of stand and subsequent reduction factor.        *
C   2.  Compute reduction factor for climatic effects - DEGD.               *
C   3.  Compute leaf area and subsequent reduction factor for shading.      *
```

```
C   4.  Calculate growth increment and reduce by each computed factor.         *
C   5.  Compute tree mortality from random and stress factors.                 *
C   6.  Remove tree if computed to be dead.                                    *
C Important variables include:                                                 *
C   BAR - Basal area of stand in cm**2                                          *
C   DEGD - number of degree days for simulation stand.                         *
C   T - growth reduction factor for climatic effects.                          *
C   S - reduction factor for soil fertility effects.                           *
C   AL - proportion of available light to a given tree.                        *
C   H - tree height in cm                                                      *
C   DINC - diameter growth increment for current simulation year in cm         *
C   ISHADE(i) - shade tolerance category for species i.                        *
C   GRF(i) - growth reduction factor for pollution for species i.              *
C   GRWS(i) - growth reduction factor for water stress for species i.          *
C   AGEMX(i) - maximum attainable age for species i.                           *
C   PP(i) - probability of random mortality.                                   *
C   MORT,B1,B2,B3,CEXT - equation coefficients.                                *
C   AINC(i) - minimum possible diameter growth for species i.                  *
C Subroutines called:                                                          *
C   SHADE - computes leaf area index by height class.                          *
C   ERROR - prints error messages if run bounds are violated.                  *
C ******************************************************************************
      dimension dbh(1),ntrees(1),sla(1),pd(1),grf(1),s1(1),s2(1),
     &          s3(1),age(1),itop(1)
      character*1 ishade,imoist,spp*4,spec*4
      common/leaf/aside(7),c(7),alpha(7),b2(7),b3(7),cext(8),crat(7),
     &            sigma(7),ap(7),betap(7)
      common/oper/ ns,nspan,nruns,clrcut,nwrstr,ifire,sburn,ibr,impb
      common/trunk/g(7),agemx(7),dm(7),hm(7),spm(8),ysc(7)
      common/water/grws(7),ws0(7),wsm(7),nws(7),wr,grdd(7),grbar(7)
      common/hdata/phi,xmbar(8),degd,ainc(7),binfest(8),ibcycle(8),brr
      common/climat/dmin(7),dopt(7),dmax(7),baset(12),basep(12),baseh
      common/types/ishade(7),imoist(7),spp(7)
      common/plotq/elev,rock,till,soilm,text,excess,pltsiz,ifg
      integer clrcut
      real mort(2)
      data pi/3.14159265/,mort/0.328,0.100/

      nlive = 0
      n= isum(ntrees,ns)
      if(n.eq.0)return
      if(kyr .lt. impb) then
           inend = 0
      elseif(kyr .eq. impb) then
           inend = kyr + ibcycle(ifg)
      endif

C ..... Compute total basal area of entire stand
      bar= 0.
      do 5 j= 1,n
        bar= bar+(pi/4.)*dbh(j)**2
    5 continue

C ..... Compute shading leaf area for each tree
      call shade(ntrees,dbh,sla,s1,s2,s3,pltsiz)
      do 10 i=1,ns

C ..... Calculate soil fertility reduction factor from basal area
         grbar(i) = 1.0 - bar / (xmbar(ifg) * 10000.0 * pltsiz)
```

```
            ni= ntrees(i)
            if (ni.eq.0) go to 10
            if(i .eq. 1) then
                 jj = 0
            else
                 jj= isum(ntrees,i-1)
            endif
            do 20 j= 1,ni

C ..... Compute standardized available light, then calculated growth
C ..... increment (maximum)
               al= phi*exp(-cext(ifg)*sla(j+jj))
               h= 137.+b2(i)*dbh(j+jj)-b3(i)*dbh(j+jj)**2.0
               dinc= g(i)*dbh(j+jj)*(1.-h*dbh(j+jj)/(hm(i)*dm(i)))
               dinc=dinc/(274.+3.*b2(i)*dbh(j+jj)-4.*b3(i)*dbh(j+jj)**2.0)

C ..... Reduce diameter increment for shading effects
               if(ishade(i).eq.'i' .or. ishade(i) .eq. 'I') then
                    dinc = 2.24*(1.-exp(-1.136*(al-.08)))*dinc
               elseif(ishade(i) .eq. 't' .or. ishade(i) .eq. 'T' .or.
     &                ishade(i) .eq. 'm' .or. ishade(i) .eq. 'M') then
                    dinc = (1.-exp(-4.64*(al-.05)))*dinc
               endif

C ..... Reduce diameter increment using environmental growth reduction factors

               dinc= dinc * grf(i) * grws(i) * grdd(i) * grbar(i)
               if (dinc .gt. 5.0) call error(11)

C ..... Calculate tree mortality for random and stress factors
               if(spp(i) .eq. 'pial') then
                    pd(j+jj) = 3.0 / agemx(i)
               else
                    pd(j+jj) = 4.0 / agemx(i)
               endif
               if (dinc .lt. ainc(i)) then
                    if(ishade(i) .eq. 'I' .or. ishade(i) .eq. 'i') then
                       pd(j+jj) = pd(j+jj) + mort(1) - (mort(1)*pd(j+jj))
                    else
                       pd(j+jj) = pd(j+jj) + mort(2) - (mort(2)*pd(j+jj))
                    endif
               endif

C ..... Calculate tree mortality if blister rust infection
               if(kyr .ge. ibr) then
                    if(spp(i) .eq. 'pial' .or. spp(i) .eq. 'pimo') then
                         dia = dbh(j+jj)
                         tage = age(j+jj)
                         infec = itop(j+jj)
                         if(infec .eq. 0) then
                              rnum = rnd()
                              if(rnum .lt. pinfec) itop(j+jj) = 1
                         endif
                         call rust(dia,tage,prob,pinfec,infec)
                         pd(j+jj) = pd(j+jj) + prob
                    endif
               endif

C ..... Calculate tree mortality if mountain pine beetle infestation
               if(kyr .eq. impb) then
```

```
                 if(spp(i) .eq. 'pial' .or. spp(i) .eq. 'pico'
     &           .or. spp(i) .eq. 'pipo' .or. spp(i) .eq.
     &           'pimo') then
                      if(dbh(j+jj) .gt. 10.0) then
                           npine = npine + 1
                           nlive = nlive + 1
                      endif
                 endif
            endif
            if(kyr .gt. impb .and. kyr .le. inend) then
                 if(spp(i) .eq. 'pial' .or. spp(i) .eq. 'pico' .or.
     &              spp(i) .eq. 'pipo' .or. spp(i) .eq. 'pimo') then
                      dia = dbh(j+jj)
                      tage = age(j+jj)
                      spec = spp(i)
                      call beetle(spec,dia,tage,prob)
                      pd(j+jj) = pd(j+jj) + prob
                      if(dbh(j+jj) .gt. 10.0) then
                           nlive = nlive + 1
                      endif
                 endif
            endif

C ..... Incrementing individual tree diameter
            dbh(j+jj) = dbh(j+jj) + dinc
   20    continue
   10 continue
      if(npine .gt. 0) then
           pinfest = 1.0 - float(nlive) / float(npine)
      else
           pinfest = 1.0
      endif
      if(kyr .gt. inend .or. pinfest .ge. binfest(ifg)) inend = 0
      return
      END
      SUBROUTINE injury(ntrees,dbh,byram,p,hs,icwf)
C ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C This subroutine calculates scorch height then estimates tree mort-   :
C  ality from scorch height using the function RISK.  Parameters for   :
C  RISK include percent crown scorched, DBH, and scorch height.        :
C Variables are:                                                       :
C     c1,c2,c3 ... coefficients for byrams equation                    :
C     byram ...... byrams fire intensity (kw/m)                        :
C     wind....... wind speed          (km/hr)                          :
C     tkill ...... lethal foliage temperature  (deg cent)              :
C     bc ......... ratio of bark thickness to diameter at breast height:
C     hs ......... crown scorch height in meters                       :
C     p .......... prob tree dies within one year                      :
C     t .......... ambient air temperature   (deg cent)                :
C ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      common/oper/ ns,nspan,nruns,clrcut,nwrstr,ifire,sburn,ibr,impb
      common/leaf/aside(7),c(7),alpha(7),b2(7),b3(7),cext(8),crat(7),
     &            sigma(7),ap(7),betap(7)
      common/sites/ occur(500),rh,wind,ttheta,t
      common/mort/ d1(7),d2(7),d3(7),bc(7)
      integer clrcut
      dimension ntrees(1),dbh(1),p(1)
      data c1/.7422/,c2/.02559/,c3/.2778/,tkill/60./,hsmin/.1/

      n= isum(ntrees,ns)
```

```
      if (n.eq.0) return

C ..... Byrams equation for crown scorch height
      hs= c1*byram**1.1667/(sqrt(c2*byram+(c3*wind)**3)*(tkill-t))
      if (hs.lt.hsmin) return
      do 10 k= 1,ns
           kkk = k
           nk= ntrees(k)
           if (nk.eq.0) go to 10
           if(kkk .eq. 1) then
                jj = 0
           else
                jj= isum(ntrees,kkk-1)
           endif
           do 20 j= 1,nk

C ..... Calculation of crown scorch volume (Ryan Rheinhardt)
                ht = (137.+b2(k)*dbh(j+jj)-b3(k)*dbh(j+jj)**2)/100.0
                hcr = crat(k)*ht
                b = hs - (ht - hcr)
                if(b .le. 0.0) b = 0.0
                if(b .ge. hcr) b = hcr
                ck = 100.0 * (b*(2*hcr-b)/(hcr**2.0))
                dia = dbh(j+jj)

C ..... Estimation of probability of tree mortality from fire
                if(icwf .eq. 1) then
                     p(j+jj) = 1.00
                else
                     p(j+jj)= risk(ck,dia,kkk)
                endif
   20      continue
   10 continue
      return
      END
      SUBROUTINE kill(nalive,ndead,dbh,pd,u,age,branch,itop,icwf)
C :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C : Subroutine kill eliminates trees from simulation plot by first    :
C : generating a random number (u(k)) and comparing it with current   :
C : probability of death for a given tree (p(i)).  If u(k) less than :
C : p(i) the tree is removed and the standing woody fuel is distrib-  :
C : uted on plot with subroutine SNAG.                                :
C :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      common/limits/ mxtrs,maxspc,mxdd,mxyrs,maxbin
      common/leaf/aside(7),c(7),alpha(7),b2(7),b3(7),cext(8),crat(7),
     &            sigma(7),ap(7),betap(7)
      common/oper/ ns,nspan,nruns,clrcut,nwrstr,ifire,sburn,ibr,impb
      integer clrcut
      dimension nalive(1),ndead(1),dbh(1),pd(1),u(1),age(1),
     &          itop(1)

      n= isum(nalive,ns)
      if (n.eq.0) return

C ..... Call the random number generator and initialize
      call rgen(u,n)
      indxl= 0
      ksp= 0
      ksum= 0
```

```
C ..... Calculate mortality by tree and species
      do 10 k= 1,n
    5      if (k.le.ksum) go to 6
           ksp= ksp+1
           ksum= ksum+nalive(ksp)
           go to 5

C ..... If a tree lives:
    6      if (u(k).gt.pd(k)) then
                indxl= indxl+1
                dbh(indxl)= dbh(k)
                age(indxl)=age(k)+1.0
                pd(indxl)= pd(k)
                itop(indxl) = itop(k)
           else

C ..... If a tree dies:
                dia = dbh(k)
                if(icwf .eq. 0) call snag(dia,branch,ksp)
                nalive(ksp)= nalive(ksp)-1
                ndead(ksp)= ndead(ksp)+1
           endif
   10 continue
      return
      END
      SUBROUTINE loader(ntrees,dbh,lnl,ln,dnl,dn,wood,yr,branch,icwf)
C :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C  This subroutine adds woody fuel, duff and litter to the forest floor. :
C  Woody fuel is collected in WOOD(i) while litter is stored in LN(i).   :
C  The duff weight is also calculated and stored in DN(i).               :
C  The output variables:                                                 :
C     fyr .... number of years to reach maximum fuel loadings            :
C     fload .. maximum fuel loading for woody fuel in a fire group       :
C     lnl .... previous year's litter loading for 100 sq meters stand    :
C     ln  .... current year litter loading for 100 sq meters             :
C              fuel properties                                           :
C     dkl .... litter decay constants                                    :
C     ltd .... litter to duff conversion constants                       :
C     dkf .... fresh litter fall decay constants                         :
C     dkd .... duff decay constants                                      :
C              leaf properties                                           :
C     ffl .... fraction of leaf biomass which  falls in one year         :
C :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      dimension ntrees(1),dbh(1)
      common/oper/ ns,nspan,nruns,clrcut,nwrstr,ifire,sburn,ibr,impb
      common/plotq/elev,rock,till,soilm,text,excess,pltsiz,ifg
      common/polut/ndyr(7),dmoist
      common/fuel3/ dkl(7),dkd(7),dkf(7),ltd(7)
      common/fuel4/ abm(7),ffl(7),fyr(3,8),fload(3,8)
      real ln(1),lnl(1),dn(1),dnl(1),wood(3)
      real dkl,dkd,dkf,ltd,abm,ffl,fnl(7),litduff
      integer clrcut,yr,fyr
      data litduff/0.100/

C ..... Initializing fuel loadings for start of simulation
      if(yr .eq. 1 .or. icwf .eq. 1) then
           do 5 i = 1,3

C ..... If the stand has been clearcut
                if(clrcut .eq. 1) then
```

```
                     wood(i) = fload(i,ifg)/float(fyr(i,ifg))
                     do 1 j = 1,ns
                          ln(j) = litduff * 0.25
                          dn(j) = litduff * 0.75
    1                continue
                elseif(icwf .eq. 1) then

C ..... if stand has had a crown fire
                     wood(i) = (fload(i,ifg)/float(fyr(i,ifg)))*0.1
                     do 2 j = 1,ns
                          ln(j) = litduff * 0.25
                          dn(j) = litduff * 0.75
    2                continue
                else

C ..... If the stand is mature
                     wood(i) = fload(i,ifg)/(float(fyr(i,ifg))/2.0)
                     do 3 j = 1,ns
                          ln(j) = litduff * 0.25
                          dn(j) = litduff * 0.75
    3                continue
                endif
    5      continue
           branch = 0.0
           return
      endif

C ..... Calculating needlefall then litter accumulation
      do 10 k= 1,ns
        kkk = k
        dnl(k)= dn(k)
        lnl(k)= ln(k)
        nk= ntrees(k)
        if (nk.eq.0) go to 10
        if(kkk .eq. 1) then
             jj = 0
        else
             jj= isum(ntrees,kkk-1)
        endif
        call needle(sla,jj,nk,dbh,kkk,wgt,pltsiz)
        if(clrcut .eq. yr) then
             fnl(k)= wgt/(1000.0*pltsiz)
        else
             fnl(k)= wgt/(1000.0*float(ndyr(k))*pltsiz)
        endif

C ..... The dynamic loading equations for litter and duff components
        ln(k)= lnl(k)*(1.-dkl(k)-ltd(k))+fnl(k)*(1.-dkf(k))
        dn(k)= dnl(k)*(1.-dkd(k))+lnl(k)*ltd(k)
   10 continue

C ..... Calculation of woody fuel components - 1,10,100 hour fuels
      do 30 i = 1,3
           if(wood(i) .lt. fload(i,ifg)) then
                wood(i) = wood(i) + fload(i,ifg)/float(fyr(i,ifg))
     &                    + ((branch * 0.333) / pltsiz)
           else
                wood(i) = fload(i,ifg)
           endif
   30 continue
```

```
      branch = 0.0
      return
      END
      SUBROUTINE needle(sla,ikk,nkk,dbh,kk,wgt,pltsiz)
C ************************************************************************
C This subroutine calculates the crown weight from equations in   Brown *
C (1984) then gets the percentage of the weight that is foiliage from   *
C subroutine PFOIL.  Using these and other species-specific parameters  *
C the leaf area index SLA of the stand is estimated.                    *
C ************************************************************************
      character*1 imoist,ishade,spp*4
      common/leaf/aside(7),c(7),alpha(7),b2(7),b3(7),cext(8),crat(7),
     &            sigma(7),ap(7),betap(7)
      common/types/ishade(7),imoist(7),spp(7)
      dimension dbh(1)

      sla = 0.0
      wgt = 0.0

C ..... Calculate the weight of live crown by species
      do 10 ii = 1,nkk
           dia = dbh(ii+ikk) * 0.3937
           call foil(pfoil,dia,kk)
           if(spp(kk) .eq. 'abla') then
                wt = (alpha(kk) + c(kk)*dia**(2.0))*453.59
           else
                wt = 453.59 * exp(alpha(kk) + c(kk)*alog(dia))
           endif
           wgt = wgt + wt * pfoil

C ..... Calculate the leaf area for this species
           sla = sla + ((wt / 0.5)*sigma(kk)/aside(kk))/(100000.0*
     &                 pltsiz)
   10 continue
      return
      END
      SUBROUTINE output(x,nyears)
C ******************************************************************
C This subroutine writes the average basal area of each tree for  *
C each year of simulation.  X(i,j) contains species' basal area.  *
C ******************************************************************
      common/oper/ ns,nspan,nruns,clrcut,nwrstr,ifire,sburn,ibr,impb
      integer clrcut
      dimension x(nyears,1)

      open(unit=7,file='BASAL.DAT',pad='yes',recl=100)

C ..... Print the average basal area over nspan years to unit 7
      do 10 j= 1,nspan
           write(7,2000) (x(j,k),k=1,ns)
   10 continue
      close(7)
      return
 1000 format(i3,1x,i3)
 2000 format(7f10.3)
      END
      SUBROUTINE pinalb(fnj,sla,dbh,age,ntrees,itop,ccrop,cones)
C **********************************************************************
C *                    - subroutine pinalb -
C * This subroutine calculates the number of whitebark pine seedlings *
```

```
C * to establish on the simulation plot.  The algorithm is based on   *
C * a cone:bird ratio which indicates availability  of cones to the    *
C * Clark's nutcracker.  Excess cones are then available for bears     *
C * and squirrels.  In addition, the number of seedlings (or caches)  *
C * depends on density of foilage modeled as a function of leaf area  *
C * index.                                                             *
C **********************************************************************
      dimension dbh(1),age(1),itop(1)
      common/leaf/aside(7),c(7),alpha(7),b2(7),b3(7),cext(8),crat(7),
     &            sigma(7),ap(7),betap(7)
      common/hdata/phi,xmbar(8),degd,ainc(7),binfest(8),ibcycle(8),brr
      common/wbark/ cmax,agecon,dbhmin,birds,spc,spcac,cyr(4),fmax,cpt,
     &              pfind,ssc
      common/plotq/elev,rock,till,soilm,text,excess,pltsiz,ifg
      common/types/ishade(7),imoist(7),spp(7)
      common/oper/ ns,nspan,nruns,clrcut,nwrstr,ifire,sburn,ibr,impb
      integer ntrees(1),clrcut
      character*1 ishade,imoist,spp*4
      data pw1,pw2,pw3,a1,a2,a3/5.0,5.0,5.5,0.6,0.6,0.8/
      data amin,aopt,amax/40.0,250.0,850.0/

C ..... Line functions for cacheability etc..
      pref(y) = 1.00 - ((exp(-(((y / fmax) - 1.0) / (a1 - 1.0))**pw1)
     &          - exp(-(-1.0 / (a1 - 1.0))**pw1)) /
     &          (1.0 - exp(-(-1.0 / (a1 - 1.0))**pw1)))
      frac(y) = 1.0 - ((exp(-(((y / cmax) - 1.0) / (a2 - 1.0))**pw2)
     &          - exp(-(-1.0 / (a2 - 1.0))**pw2)) /
     &          (1.0 - exp(-(-1.0 / (a2 - 1.0))**pw2)))
      cac(y) = exp((y / cmax)**(pw3) - (1.0 + 0.5 * ((cmax-y)/cmax)))

C ..... Initialize appropriate variables
      v = (amax - aopt) / (aopt - amin)
      cones = 0.0

C ..... Search to find if whitebark species is present
      do 20 i = 1,ns
           if(spp(i) .eq. 'pial') then
                ntrs = ntrees(i)

C ..... Calculation of cone bearing trees on plot
                ictree = 0
                if(i .eq. 1) then
                     ii = 0
                else
                     ii = isum(ntrees,i-1)
                endif
                do 1 j = 1,ntrs
                     if(dbh(j+ii) .gt. dbhmin .and. age(j+ii) .gt.
     &                  amin .and. age(j+ii) .le. amax) then
                           ictree = ictree + 1
                     endif
    1           continue

C ..... Calculation of relative size of cone crop
                rnum = rnd()
                do 5 j = 1,4
                     if(rnum .le. cyr(j)) then
                          confac = float(j-1) / 3.0
                          go to 6
                     endif
```

```
    5          continue

    6          if(ictree .le. 1) then
                    cones = ccrop * confac
                    go to 30
               endif

C ..... Calculation of number of cones per tree and then the summation
               if(i .eq. 1) then
                    ii = 0
               else
                    ii = isum(ntrees,i-1)
               endif
               do 10 j = 1,ntrs
                    if(dbh(j+ii) .gt. dbhmin .and. age(j+ii) .gt.
     &                 amin .and. age(j+ii) .le. amax) then
                          t = ((age(j+ii) - amin) *
     &                        (amax-age(j+ii))**v) /
     &                        (((amax-aopt)**v) *
     &                        (aopt - amin))
                          cones = (cpt * t) + cones
                          if(itop(j+ii) .eq. 1) cones = cones * 0.1
                    endif
   10          continue
               ccrop = cones
               if(cones .gt. 0.0) then
                    cones = cones * confac
               else
                    cones = cmax * 0.1 * confac
               endif
               go to 30
          endif
   20 continue
      return

C ..... Calculation of the number of caches on the plot
   30 caches = ((cones * spc) / spcac) * (1.0 - (pfind + ssc))
      if(caches .le. 0.0) caches = 0.0

C ..... Calculation of the cones per bird ratio
      cpb = ((cones / birds) / pltsiz) * 4046.849
      if(cpb .gt. cmax) cpb = cmax

C ..... Calculation of the fraction of cones available to griz
      fcone = frac(cpb)
      if(fcone .le. 0.2) fcone = 0.2
      if(fcone .gt. 0.9) fcone = 0.9

C ..... Calculation of the reduction factor for cacheability
      cabil = cac(cpb)

C ..... Calculation of the reduction factor for preferability (LAI)
      if(sla .gt. fmax) sla = fmax
      pleaf = pref(sla)
      if(pleaf .le. 0.1) pleaf = 0.1
      if(pleaf .gt. 1.0) pleaf = 1.0

C ..... Final calculation of seedlings started in current year FNJ
      fnj = caches * cabil * pleaf
```

```
      return
      END
      SUBROUTINE pollut(grf,kyr)
C ************************************************************************
C This subroutine calculated growth reduction effects from air pollutants *
C  However, since air pollution effects are minimal in the Inland North-  *
C  west, the growth reduction factor for pollution was set equal to 1.0.  *
C Important Variables:                                                    *
C     grf(i).... growth reduction factor for species i                    *
C     cr(i)..... threshold of pollution damage for species i              *
C     sen(i).... sensitivity coefficient for species i                    *
C     ndyr(i)... number of years needles are retained for species i       *
C     cbar...... seasonal average so2 concentration ppm                   *
C     kyr....... current year                                             *
C     ns.......  number of tree species                                   *
C ************************************************************************
      common/polut/ndyr(7),dmoist
      common/oper/ ns,nspan,nruns,clrcut,nwrstr,ifire,sburn,ibr,impb
      integer clrcut
      dimension grf(1)
      if (ns.le.0) return
      if (kyr.eq.0) return

C ..... Set pollution growth reduction factor to 1.0 for Montana
      do 10 i= 1,ns
           grf(i)= 1.000
   10 continue
      return
      END
      SUBROUTINE rgen(x,i)
C ***********************************************************************
C  Subroutines RGEN and RANST and function RAN are random number         *
C  generators for the model. Users should use their own random number    *
C  generators which return n random numbers u between 0 and 1 with       *
C  uniform distribution.  XRANDOM is a Perkin-Elmer generator.           *
C ***********************************************************************
      dimension x(i)

C ..... Fill array x(i) with random numbers from XRANDOM
      do 10 j=1,i
           xx = rnd()
           x(j) = xx
   10 continue
      return
      END
      SUBROUTINE rings(n,u)
C ************************************************************************
C Subroutine RINGS produces an array containing the simulation years that *
C are fire years.  This is a stochastic function where a random number is *
C generated (U(i)) and if less than p (set in the data statement) then    *
C a fire is to be simulated for that year.  The calculation is abandoned  *
C if IFYR is greater than zero (user specified fire years).               *
C Variables are:                                                          *
C    X(k) - fire year array containing 0 (no fire) or 1 (fire)            *
C    U(i) - random number array                                           *
C    R - number of years to block fires after a fire has been generated   *
C    PNB - probability of fire in a blocked year.                         *
C    IFYR - user specified fire interval                                  *
C ************************************************************************
      common/oper/ ns,nspan,nruns,clrcut,nwrstr,ifire,sburn,ibr,impb
```

```
      common/sites/ x(500),rh,wind,ttheta,t
      integer x,r,yes,no,circut
      real u(1),ul(1)
      data r/3/,p/.0125/,yes/1/,no/0/
      if (n.eq.0) return

C ..... Initializing fire array
      do 10 i= 1,n
           x(i)= no
   10 continue

C ..... Assign fire years if user specified
      if(ifire .gt. 0) then
           ifyr = ifire
           do 20 k = 1,n
                if(k .eq. ifyr) then
                     x(k) = yes
                     ifyr = ifyr + ifire
                endif
   20      continue

C ..... Assign only one fire year if number is negative
      elseif(ifire .lt. 0) then
           ifyr = iabs(ifire)
           x(ifyr) = yes
           return

C ..... Calculate fire years using stochastic function
      else
           i= 0
           if (r.eq.1) go to 35

C ..... Calculate pnb, prob of an ublocked state
           pnb= 1./(p*float(r-1))
           call rgen(ul,1)
           if (ul(1).le.pnb) go to 35

C ..... Select an integer b at random from 1,2,3,... r-1
           call rgen(ul,1)
           i= int(float(r-1)*ul(1))+1
   35      call rgen(u,n)
   40      i= i+1
           if (i.gt.n) return
           if (u(i).gt.p) go to 40

C ..... Assign fire years
           x(i)= yes
           i= i+r-1
           go to 40
      endif

      return
      END
      SUBROUTINE rust(dia,age,prob,pinfec,infec)
C **********************************************************************
C This subroutine simulates individual tree mortality in the event     *
C of a blister rust infection.  Mortality functions are from           *
C                                                                      *
C **********************************************************************
```

```
      prob = 0.0
      if(infec .eq. 0) then
           pinfec = 0.50
      elseif(infec .eq. 1) then

C ..... Calculate prob mortality for 5 needle pine from equation
           prob = exp(-0.10*dia)
           if(age .gt. 850.0) prob = 0.99
      endif

      return
      END
      SUBROUTINE shade(ntrees,dbh,sla,h,temp,indx,pltsiz)
C **********************************************************************
C This subroutine calculates the effective leaf area index by tree     *
C height to estimate shading effects for individual trees.  Logic is:  *
C   1.  Calculated leaf areas for every tree.                          *
C   2.  Sort leaf areas according to height.                           *
C   3.  Sum leaf areas by height.                                      *
C   4.  Reorder the cumulative leaf areas by DBH.                      *
C Variables are:                                                       *
C   TEMP(i) - temporary array containing leaf areas                    *
C   SLA(i) - working array for leaf areas                              *
C   DBH(i) - array containing dbh for each tree on plot                *
C   ALPHA,SIGMA,ASIDE,PLTSIZ - conversion factors for crown weight to  *
C                             leaf area                                *
C Subroutines called:                                                  *
C   SORTP - sorts leaf area according to height                        *
C **********************************************************************
      common/oper/ ns,nspan,nruns,clrcut,nwrstr,ifire,sburn,ibr,impb
      integer clrcut
      character*1 imoist,ishade,spp*4
      common/types/ishade(7),imoist(7),spp(7)
      common/leaf/aside(7),c(7),alpha(7),b2(7),b3(7),cext(8),crat(7),
     &            sigma(7),ap(7),betap(7)
      common/trunk/g(7),agemx(7),dm(7),hm(7),spm(8),ysc(7)
      common/hdata/phi,xmbar(8),degd,ainc(7),binfest(8),ibcycle(8),brr
      dimension ntrees(1),dbh(1),sla(1),indx(1),temp(1),h(1)

C ..... Calculation of leaf area for each tree
      n= isum(ntrees,ns)
      if (n.eq.0) return
      do 10 k= 1,ns
           nk= ntrees(k)
           if (nk.eq.0) go to 10
           if(k .eq. 1) then
                kk = 0
           else
                kk= isum(ntrees,k-1)
           endif
           do 20 i= 1,nk
                h(i+kk)= 137.+b2(k)*dbh(i+kk)-b3(k)*dbh(i+kk)**2.0
                if(spp(k) .ne. 'abla') then
                     temp(i+kk) = ((exp(alpha(k)+c(k)*alog(dbh(i+kk)
     &                            /2.54))*453.59)/0.5)*
     &                            sigma(k)/aside(k)
                else
                     temp(i+kk) = (((alpha(k) + c(k)*(dbh(i+kk)/2.54)
     &                            **(2.0))*453.59)/0.5)*
     &                            sigma(k)/aside(k)
```

```
                endif
                temp(i+kk) = temp(i+kk)/(100000.0*pltsiz)
                indx(i+kk)= i+kk
   20       continue
   10 continue

C ..... Sort sla according to h
      call sortp(h,n,indx)
      do 40 j= 1,n
           k= indx(j)
           sla(j)= temp(k)
   40 continue

C ..... Compute final values of sla
      nml= n-1
      do 50 j= 1,nml
           temp(j)= sum(sla(j+1),n-j)
   50 continue
      temp(n)= 0.

C ..... Reorder elements of sla to correspond to dbh
      do 60 j= 1,n
           k= indx(j)
           sla(k)= temp(j)
   60 continue
      return
      END
      SUBROUTINE sitdta
C :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C This program reads in site specific data from an external file on  :
C device 3.  Values are then passed back to main driver.             :
C :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      common/plotq/elev,rock,till,soilm,text,excess,pltsiz,ifg
      common/climat/dmin(7),dopt(7),dmax(7),baset(12),basep(12),baseh
      common/hdata/phi,xmbar(8),degd,ainc(7),binfest(8),ibcycle(8),brr
      common/sites/ occur(500),rh,wind,ttheta,t
      common/polut/ndyr(7),dmoist
      common/fuel5/ amc(7),bmc(7),cmc(7),dmc(7),mmc(7),tmc,emc(7)
      character*10 name

      open(unit=3,file='SITE.DAT',form='formatted',
     &     recl=150,pad='yes')

C ..... Read in site specific data for simulation plot
      read (3,1000) name, (baset(j),j=1,12)
      write(5,1000) name, (baset(j),j=1,12)
      read (3,1000) name, (basep(j),j=1,12)
      write(5,1000) name, (basep(j),j=1,12)
      read (3,2000) name, baseh
      write(5,2000) name, baseh
      read (3,2000) name, excess
      write(5,2000) name, excess
      read (3,2000) name, phi
      write(5,2000) name, phi
      read (3,2000) name, text
      write(5,2000) name, text
      read (3,2000) name, rock
      write(5,2000) name, rock
      read (3,2000) name, elev
      write(5,2000) name, elev
```

```
      read (3,3000) name, ifg
      write(5,3000) name, ifg
      read (3,2000) name, till
      write(5,2000) name, till
      read (3,2000) name, rh
      write(5,2000) name, rh
      read (3,2000) name, wind
      write(5,2000) name, wind
      read (3,2000) name, ttheta
      write(5,2000) name, ttheta
      read (3,2000) name, t
      write(5,2000) name, t
      read (3,2000) name,pltsiz
      write(5,2000) name,pltsiz
      read (3,4000) name,(emc(j),j=1,7)
      write(5,4000) name,(emc(j),j=1,7)
      read (3,2000) name,dmoist
      write(5,2000) name,dmoist
      read (3,2000) name,brr
      write(5,2000) name,brr
      rewind 3
      close(3)
      return
 1000 format(a10,12f5.2)
 2000 format(a10,f10.3)
 3000 format(a10,i10)
 4000 format(a10,7f10.3)
      END
      SUBROUTINE site
C :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C This subroutine calculates all site parameters that are used in the :
C various algorithms throughout the program.  Actual and potential    :
C evapotranspiration are calculated along with water stress growth    :
C reduction factors.  New calculations are passed to main program.    :
C :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      dimension sitet(12),pp(12)
      dimension pei(12),actei(12),stori(12)
      common/water/grws(7),ws0(7),wsm(7),nws(7),wr,grdd(7),grbar(7)
      common/oper/ ns,nspan,nruns,clrcut,nwrstr,ifire,sburn,ibr,impb
      common/plotq/elev,rock,till,soilm,text,excess,pltsiz,ifg
      common/climat/ dmin(7),dopt(7),dmax(7),baset(12),basep(12),baseh
      common/hdata/phi,xmbar(8),degd,ainc(7),binfest(8),ibcycle(8),brr
      character*6 nsoilq,nheati,nsoilm,nape*4,nspe*4,nwra*4
      character*5 ndiff,ndegd,ngrws,na*2,npe*3,nacte,nstor
      character nwr*3,nsat*4,nwrs*4
      integer clrcut
      data nsoilq/' soilq'/,ndiff/' diff'/,ndegd/' degd'/
      data nheati/' heati'/,na/' a'/,npe/' pe'/,nacte/' acte'/
      data nsoilm/' soilm'/,nstor/' stor'/
      data nwr/' wr'/,ngrws/' grws'/,nsat/' sat'/
      data nwra/' wra'/,nape/' ape'/,nwrs/' wrs'/,nspe/' spe'/

      rocky= (100.-  rock)/100.

C ..... Rock is percent of surface area in rock outcrop
C ..... Till is depth of watering or root zone in feet.
C ..... Text is amount of available water for storage in mm/m

      till=till/3.2808
      xmbar(ifg) = xmbar(ifg)*rocky
```

```
      diff=baseh-elev
      tmin=baset(1)+(2.2*diff/1000.)
      tmax=baset(7)+(3.6*diff/1000.)
      tave=(tmax+tmin)/2.
      t=40.
      if(tmin.gt.t) write(5,98)

   98 format(1h ,' --------- you cant use minimum january temperature',
     1's greater than 40',1h ,'without modifying SUBROUTINE ',
     2'site -----------')

      if(tmax.lt.tmin) write(5,99)

   99 format(1h ,'---------- to work in the southern hemisphere one ',
     1'must modify SUBROUTINE site --------')

      degd=(365./(2.*3.14159))*(tmax-tmin)-(365./2.)*(t-tave) + (
     1(365./3.14159)*(t-tave)**2 )/(tmax-tmin)

C ..... Calculation of actual and potential evapotranspiration.
      heati=0.
      soilm=0.
      do 10 i=1,12
           sitet(i)=baset(i) + 3.6*diff/1000.
           sitet(i)=(5./9.)*(sitet(i)-32.)
           pp(i)=basep(i)*25.4
           if(sitet(i).le.0.0) go to 10
C ..... Calculation of intermediate heat index
           heati=heati+(sitet(i)/5.0)**1.514
   10 continue

C ..... Calculation of intermediate exponent in thornwaithes equation
      a=(9.675*(heati**3.)-77.1*heati**2+17920.*heati+492390.)*.000001
      m=1

C ..... Computation of storage capacity of soil
      strmax=amin1(till,10.)*text*rocky

C ..... Calculation of the water balance equation
      do 250 i=1,12
           if(sitet(i).le.0.0) go to 250
           pe=16.*(((10.*sitet(i))/heati)**a)
           if(m.gt.1) go to 220
           stor=strmax
           m=2
  220      if(pe.ge.stor + excess*pp(i)) go to 230
           acte=pe
           go to 240
  230      acte=stor+excess*(amin1(pp(i),strmax))
  240      stor=amin1(strmax,stor-acte+pp(i))
           soilm=soilm + acte
           pei(i)=pe
           actei(i)=acte
           stori(i)=stor
  250 continue
      ape=0.0
      do 300 i=1,12
           ape=ape+pei(i)
  300 continue
```

```
C ..... Calculation of the water stress reduction factor parameters
C ..... Ape=annual potential evapotranspiration
C ..... Soilm= annual actual evapotranspiration
C ..... Spe=seasonal potential evapotranspiration
C ..... Sat= seasonal actual evapotranspiration
C ..... Wra=annual actual et/annual potentail et
C ..... Wrs=seasonal actaul et/seasonal potentail et
      spe=0.0
      sat=0.0
      do 301 i=4,10
           spe=spe+pei(i)
  301      sat=sat+actei(i)
      wra=soilm/ape
      wrs=sat/spe
      wr=wra
C ..... Call wrstrs to figure reduction factor then write results to file
      call wrstrs
      write(5,3000)
      write(5,1000) nsoilq,xmbar(ifg)
      write(5,1000) ndiff,diff
      write(5,1000) ndegd,degd
      write(5,1000) nheati,heati
      write(5,1000) na,a
      write(5,1000) nsoilm,soilm
      write(5,2000) npe,(pei(k),k=1,12)
      write(5,2000) nacte,(actei(k),k=1,12)
      write(5,2000) nstor,(stori(k),k=1,12)
      write(5,1000) nwr,wr
      write(5,4000) ngrws,(grws(k),k=1,ns)
      write(5,4000) ndegd,(grdd(k),k=1,ns)
      write(5,1000) nwra,wra
      write(5,1000) nape,ape
      write(5,1000) nwrs,wrs
      write(5,1000) nspe,spe
      write(5,1000) nsat,sat
 1000 format(1x,a8,f10.3)
 2000 format(a8,12f5.1)
 3000 format(10x,'calculated parameters in site')
 4000 format(a8,7f10.4)
      return
      END
      SUBROUTINE snag(dbh,branch,kk)
C :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C  This subroutine adds the branchwood material of a dead tree to   :
C  the woody fuel components.  BRANCH variable holds the total      :
C  biomass of the dead woody branchwood until subroutine FIRE then :
C  equal values of BRANCH go into the three woody fuel types WOOD. :
C :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      character*1 imoist,ishade,spp*4
      common/types/ishade(7),imoist(7),spp(7)
      common/leaf/aside(7),c(7),alpha(7),b2(7),b3(7),cext(8),crat(7),
     &            sigma(7),ap(7),betap(7)

C ..... Calculate the downed woody fuel from a dead snage
      dbh = dbh*0.3937
      if(spp(kk) .eq. 'abla') then
           wt = (alpha(kk) + c(kk)*(dbh)**(2.0))*0.045359
      else
           wt = exp(alpha(kk)+c(kk)*alog(dbh))*0.045359
      endif
```

```
C ..... Calculate the weight of needlefall
      call foil(pfoil,dbh,kk)
      branch = branch + wt*(1.0 - pfoil)

      return
      END
      SUBROUTINE sortp(a,n,b)
C ***************************************************************************
C This subroutine sorts leaf area by height of individual trees, then      *
C passes the manipulated array back to subroutine GROW.                    *
C ***************************************************************************
      dimension a(n)
      integer b(n)
      dimension iu(16),il(16)
      integer p

      i=1
      j=n
      m=1
   5  if(i.ge.j) go to 70

C first order a(i),a(j),a((i+j)/2), and use median to split the data
   10 k=i
      ij=(i+j)/2
      t=a(ij)
      it=b(ij)
      if(a(i).le.t) go to 20
      a(ij)=a(i)
      b(ij)=b(i)
      a(i)=t
      b(i)=it
      t=a(ij)
      it=b(ij)
   20 l=j
      if(a(j).ge.t) go to 40
      a(ij)=a(j)
      b(ij)=b(j)
      a(j)=t
      b(j)=it
      t=a(ij)
      it=b(ij)
      if(a(i).le.t) go to 40
      a(ij)=a(i)
      b(ij)=b(i)
      a(i)=t
      b(i)=it
      t=a(ij)
      it=b(ij)
      go to 40
   30 a(l)=a(k)
      b(l)=b(k)
      a(k)=tt
      b(k)=itt
   40 l=l-1
      if(a(l).gt.t) go to 40
      tt=a(l)
      itt=b(l)
C split the data into a(i to l).lt.t, a(k to j).gt.t
   50 k=k+1
```

```
      if(a(k).lt.t) go to 50
      if(k.le.l) go to 30
      p=m
      m=m+1
C split the larger of the segments
      if(l-i.le.j-k) go to 60
      il(p)=i
      iu(p)=l
      i=k
      go to 80
   60 il(p)=k
      iu(p)=j
      j=l
      go to 80
   70 m=m-1
      if(m.eq.0) return
      i=il(m)
      j=iu(m)
C short sections are sorted by bubble sort
   80 if(j-i.gt.10) go to 10
      if(i.eq.1) go to 5
      i=i-1
   90 i=i+1
      if(i.eq.j) go to 70
      t=a(i+1)
      it=b(i+1)
      if(a(i).le.t) go to 90
      k=i
  100 a(k+1)=a(k)
      b(k+1)=b(k)
      k=k-1
      if(t.lt.a(k)) go to 100
      a(k+1)=t
      b(k+1)=it
      go to 90
      END
      SUBROUTINE starter(ntrees,dbh,age)
C ***********************************************************************
C This subroutine exchanges dbh and age information from temporary    *
C arrays to the working arrays.  This initially places the trees in   *
C the simulation plot.                                                 *
C ***********************************************************************
      common/init/ ntrees0(7),dbh0(4000),ncount(20,7),nbins,width
     1  ,age0(4000),agein(20,7)
      common/limits/ mxtrs,maxspc,mxdd,mxyrs,maxbin
      common/oper/ ns,nspan,nruns,clrcut,nwrstr,ifire,sburn,ibr,impb
      integer clrcut,agein
      dimension ntrees(1),dbh(1),age(1)

      do 10 j= 1,ns
        ntrees(j)= ntrees0(j)
   10 continue

      do 20 j= 1,mxtrs
        dbh(j)= dbh0(j)
        age(j)=age0(j)
   20 continue
      return
      END
      SUBROUTINE tree(nl,crop,cblock)
```

```
C :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C  This subroutine reads in species and fuel specific data for model sim- :
C  lation area (NRM).  Each input value is stored in appropriate COMMON   :
C  block or brought back to main driver.  Each value is also printed in   :
C  a file on device 5 for proof of correct entry.  Values are stratified  :
C  by species (dimensioned to seven) or fire group (dimensioned to eight).:
C :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      common/plotq/elev,rock,till,soilm,text,excess,pltsiz,ifg
      common/limits/ mxtrs,maxspc,mxdd,mxyrs,maxbin
      common/water/grws(7),ws0(7),wsm(7),nws(7),wr,grdd(7),grbar(7)
      common/leaf/aside(7),c(7),alpha(7),b2(7),b3(7),cext(8),crat(7),
     &            sigma(7),ap(7),betap(7)
      common/trunk/g(7),agemx(7),dm(7),hm(7),spm(8),ysc(7)
      common/hdata/phi,xmbar(8),degd,ainc(7),binfest(8),ibcycle(8),brr
      common/climat/ dmin(7),dopt(7),dmax(7),baset(12),basep(12),baseh
      common/birthk/sura(7),surb(7),dbulk(8,2),disequ(2,7),rdelay(8)
      common/wbark/ cmax,agecon,dbhmin,birds,spc,spcac,cyr(4),fmax,cpt,
     &              pfind,ssc
      common/types/ishade(7),imoist(7),spp(7)
      common/fuel1/ mext(2),rhop(2,7),bulk(2,8),mois(2,7)
      common/fuel2/ mps(2,7),lhv(2,7),st(2,7),se(2,7)
      common/fuel3/ dkl(7),dkd(7),dkf(7),ltd(7)
      common/fuel4/ abm(7),ffl(7),fyr(3,8),fload(3,8)
      common/fuel5/ amc(7),bmc(7),cmc(7),dmc(7),mmc(7),tmc,emc(7)
      common/mort/ d1(7),d2(7),d3(7),bc(7)
      common/polut/ndyr(7),dmoist
      common/cfire/cbd(7),vfl(7),cfmc(7),vfmc(7),cflm(7),csvr(7),
     &             vsvr(7),bl(7)
      common/oper/ ns,nspan,nruns,clrcut,nwrstr,ifire,sburn,ibr,impb
      integer clrcut,count,cblock(7),fyr
      real mext,lhv,mmc,mps,ltd,mois,crop(7),cyr
      character*10 mark,chr,name,spp*4
      character*1 imoist,ishade
      data mark/'$$$$$$$$$$'/

      open(unit=2,file='TREE1.DAT',form='formatted',
     &     recl=150,pad='YES')

      n1= 2
C ..... Find number of species
      count= 0
   10 count= count+1
      read(2,1000,end=100) chr

      if (chr.ne.mark) go to 10
      rewind 2
      ns= count-1
      if (ns.gt.maxspc) call error(7)
C ..... Write header information
      do 20 i= 1,ns
           read(2,2000) spp(i)
           write(5,2000) spp(i)
   20 continue
      write(5,1000) mark
      read (2,1000) mark
      read (2,3000) name, (hm(j),j=1,ns)
      write(5,3000) name, (hm(j),j=1,ns)
      read (2,3000) name, (dm(j),j=1,ns)
      write(5,3000) name, (dm(j),j=1,ns)
      read (2,3000) name, (agemx(j),j=1,ns)
```

```
      write(5,3000) name, (agemx(j),j=1,ns)
      read (2,3000) name, (dmin(j),j=1,ns)
      write(5,3000) name, (dmin(j),j=1,ns)
      read (2,3000) name, (dopt(j),j=1,ns)
      write(5,3000) name, (dopt(j),j=1,ns)
      read (2,3000) name, (dmax(j),j=1,ns)
      write(5,3000) name, (dmax(j),j=1,ns)
      read (2,8000) name, (spm(j),j=1,8)
      write(5,8000) name, (spm(j),j=1,8)
      read (2,6000) name, (aside(j),j=1,ns)
      write(5,6000) name, (aside(j),j=1,ns)
      read (2,3000) name, (c(j),j=1,ns)
      write(5,3000) name, (c(j),j=1,ns)
      read (2,6000) name, (alpha(j),j=1,ns)
      write(5,6000) name, (alpha(j),j=1,ns)
      read (2,3000)  name, (sigma(j),j=1,ns)
      write(5,3000) name, (sigma(j),j=1,ns)
      read (2,6000) name, (ap(j),j=1,ns)
      write(5,6000) name, (ap(j),j=1,ns)
      read (2,6000) name, (betap(j),j=1,ns)
      write(5,6000) name, (betap(j),j=1,ns)
      read (2,8000) name, (cext(j),j=1,8)
      write(5,8000) name, (cext(j),j=1,8)
      read (2,5000) name, (ishade(j),j=1,ns)
      write(5,5000) name, (ishade(j),j=1,ns)
      read (2,5000) name, (imoist(j),j=1,ns)
      write(5,5000) name, (imoist(j),j=1,ns)
      read (2,8000) name, (xmbar(j),j=1,8)
      write(5,8000) name, (xmbar(j),j=1,8)
      read (2,6000) name, (crat(j),j=1,ns)
      write(5,6000) name, (crat(j),j=1,ns)
      read(2,3000)  name, mext(1)
      write(5,3000) name, mext(1)
      read(2,3000)  name, (amc(k),k=1,ns)
      write(5,3000) name, (amc(k),k=1,ns)
      read(2,3000)  name, (bmc(k),k=1,ns)
      write(5,3000) name, (bmc(k),k=1,ns)
      read(2,3000)  name, (cmc(k),k=1,ns)
      write(5,3000) name, (cmc(k),k=1,ns)
      read(2,3000)  name, (dmc(k),k=1,ns)
      write(5,3000) name, (dmc(k),k=1,ns)
      read(2,3000)  name, (mmc(k),k=1,ns)
      write(5,3000) name, (mmc(k),k=1,ns)
      read(2,3000)  name, tmc
      write(5,3000) name, tmc
      read(2,3000)  name, (rhop(1,k),k=1,6)
      write(5,3000) name, (rhop(1,k),k=1,6)
      read(2,8000)  name, (bulk(1,k),k=1,8)
      write(5,8000) name, (bulk(1,k),k=1,8)
      read(2,3000)  name, (lhv(1,k),k=1,6)
      write(5,3000) name, (lhv(1,k),k=1,6)
      do 50 i = 1,ifg
           read(2,3000)  name, (mps(1,k),k=1,6)
           if(i .eq. ifg) write(5,3000) name, (mps(1,k),k=1,6)
   50 continue
      do 60 i = 1,8-ifg
           read(2,1000) mark
   60 continue
      read(2,3000)  name, (st(1,k),k=1,6)
      write(5,3000) name, (st(1,k),k=1,6)
```

```
read(2,3000)  name, (se(1,k),k=1,6)
write(5,3000) name, (se(1,k),k=1,6)
read(2,4000)  name, (dkl(k),k=1,ns)
write(5,4000) name, (dkl(k),k=1,ns)
read(2,4000)  name, (ltd(k),k=1,ns)
write(5,4000) name, (ltd(k),k=1,ns)
read(2,4000)  name, (dkf(k),k=1,ns)
write(5,4000) name, (dkf(k),k=1,ns)
read(2,4000)  name, (dkd(k),k=1,ns)
write(5,4000) name, (dkd(k),k=1,ns)
read(2,4000)  name, (ffl(k),k=1,ns)
write(5,4000) name, (ffl(k),k=1,ns)
read(2,4000)  name, (d1(k),k=1,ns)
write(5,4000) name, (d1(k),k=1,ns)
read(2,3000)  name, (d2(k),k=1,ns)
write(5,3000) name, (d2(k),k=1,ns)
read(2,3000)  name, (d3(k),k=1,ns)
write(5,3000) name, (d3(k),k=1,ns)
read(2,3000)  name, (bc(k),k=1,ns)
write(5,3000) name, (bc(k),k=1,ns)
read(2,3000)  name, (rhop(2,k),k=1,nl)
write(5,3000) name, (rhop(2,k),k=1,nl)
read(2,3000)  name, (bulk(2,k),k=1,nl)
write(5,3000) name, (bulk(2,k),k=1,nl)
read(2,3000)  name, (lhv(2,k),k=1,nl)
write(5,3000) name, (lhv(2,k),k=1,nl)
read(2,3000)  name, (mps(2,k),k=1,nl)
write(5,3000) name, (mps(2,k),k=1,nl)
read(2,3000)  name, (st(2,k),k=1,nl)
write(5,3000) name, (st(2,k),k=1,nl)
read(2,3000)  name, (se(2,k),k=1,nl)
write(5,3000) name, (se(2,k),k=1,nl)
read(2,3000)  name, (mois(2,k),k=1,nl)
write(5,3000) name, (mois(2,k),k=1,nl)
read(2,7000)  name,(ndyr(j),j=1,ns)
write(5,7000) name,(ndyr(j),j=1,ns)
read (2,4000) name,(ainc(j),j=1,ns)
write(5,4000) name,(ainc(j),j=1,ns)
read(2,3000)  name,(ws0(k),k=1,ns)
write(5,3000) name,(ws0(k),k=1,ns)
read(2,3000)  name,(wsm(k),k=1,ns)
write(5,3000) name,(wsm(k),k=1,ns)
read(2,7000)  name,(nws(k),k=1,ns)
write(5,7000) name,(nws(k),k=1,ns)
read (2,4000) name,(sura(j),j=1,ns)
write(5,4000) name,(sura(j),j=1,ns)
read (2,4000) name,(surb(j),j=1,ns)
write(5,4000) name,(surb(j),j=1,ns)
read (2,8000) name,(dbulk(j,1),j=1,8)
write(5,8000) name,(dbulk(j,1),j=1,8)
read (2,8000) name,(dbulk(j,2),j=1,8)
write(5,8000) name,(dbulk(j,2),j=1,8)
read (2,8000) name,sburn
write(5,8000) name,sburn
read (2,4000) name,(crop(j),j=1,ns)
write(5,4000) name,(crop(j),j=1,ns)
read (2,7000) name,(cblock(j),j=1,ns)
write(5,7000) name,(cblock(j),j=1,ns)
read (2,4000) name,(ysc(j),j=1,ns)
write(5,4000) name,(ysc(j),j=1,ns)
```

```
      read (2,4000) name,(disequ(1,j),j=1,ns)
      write(5,4000) name,(disequ(1,j),j=1,ns)
      read (2,4000) name,(disequ(2,j),j=1,ns)
      write(5,4000) name,(disequ(2,j),j=1,ns)
      do 70 i = 1,3
           read (2,9000) name,(fyr(i,j),j=1,8)
           write(5,9000) name,(fyr(i,j),j=1,8)
   70 continue
      do 80 i = 1,3
           read (2,8000) name,(fload(i,j),j=1,8)
           write(5,8000) name,(fload(i,j),j=1,8)
   80 continue
      read (2,3000) name,(cbd(i),i=1,ns)
      write(5,3000) name,(cbd(i),i=1,ns)
      read (2,3000) name,(vfl(i),i=1,ns)
      write(5,3000) name,(vfl(i),i=1,ns)
      read (2,3000) name,(cfmc(i),i=1,ns)
      write(5,3000) name,(cfmc(i),i=1,ns)
      read (2,3000) name,(vfmc(i),i=1,ns)
      write(5,3000) name,(vfmc(i),i=1,ns)
      read (2,3000) name,(cflm(i),i=1,ns)
      write(5,3000) name,(cflm(i),i=1,ns)
      read (2,3000) name,(csvr(i),i=1,ns)
      write(5,3000) name,(csvr(i),i=1,ns)
      read (2,3000) name,(vsvr(i),i=1,ns)
      write(5,3000) name,(vsvr(i),i=1,ns)
      read (2,3000) name,(bl(i),i=1,ns)
      write(5,3000) name,(bl(i),i=1,ns)
      read (2,8000) name,(binfest(i),i=1,8)
      write(5,8000) name,(binfest(i),i=1,8)
      read (2,9000) name,(ibcycle(i),i=1,8)
      write(5,9000) name,(ibcycle(i),i=1,8)
      read (2,8000) name,(rdelay(i),i=1,8)
      write(5,8000) name,(rdelay(i),i=1,8)

      do 90 i = 1,ns
           if(spp(i) .eq. 'pial') then
                read(2,9100) name,cmax,agecon,dbhmin,birds,spc,spcac,
     &                       pfind,(cyr(j),j=1,4),fmax,cpt,ssc
                write(5,9100) name,cmax,agecon,dbhmin,birds,spc,spcac,
     &                        pfind,(cyr(j),j=1,4),fmax,cpt,ssc
                go to 99
           endif
   90 continue
   99 close(2)
      return

  100 call error(8)
      close(2)
      return

C ######################## FORMATS ######################

 1000 format(a10)
 2000 format(a4)
 3000 format(a10,7f10.3)
 4000 format(a10,7f10.4)
 5000 format(a10,7(9x,a1))
 6000 format(a10,7f10.7)
 7000 format(a10,7i10)
```

```
 8000 format(a10,7f10.4,/,10x,f10.4)
 9000 format(a10,7i10,/,10x,i10)
 9100 format(a10,7f10.1,/,10x,7f10.4)
      END
      SUBROUTINE wrstrs
C :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C This subroutine computes the growth reduction factor due to         :
C water stress.  This is a value between 0 and 1 and is stored        :
C in the array GRWS(i).                                               :
C :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      common/oper/ ns,nspan,nruns,clrcut,nwrstr,ifire,sburn,ibr,impb
      common/water/grws(7),ws0(7),wsm(7),nws(7),wr,grdd(7),grbar(7)
      common/hdata/phi,xmbar(8),degd,ainc(7),binfest(8),ibcycle(8),brr
      common/climat/ dmin(7),dopt(7),dmax(7),baset(12),basep(12),baseh
      integer clrcut

      do 10 i=1,ns

C ..... Calculate growth reduction factor for water stress
           if(nwrstr .ne. 0) then
                grws(i)=1.- ( (wsm(i)-wr)/(wsm(i)-ws0(i)) )**nws(i)
                if(grws(i).lt.0.0) grws(i)=0.0
           else
                grws(i) = 1.0
           endif

C ..... Calculate climatic reduction factor using degree-days
         if(degd .gt. dmin(i) .and. degd .lt. dmax(i)) then
              v = (dmax(i) - dopt(i)) / (dopt(i) - dmin(i))
              grdd(i) = ((degd - dmin(i)) * (dmax(i)-degd)**v) /
     &            (((dmax(i)-dopt(i))**v) * (dopt(i) - dmin(i)))
         else
              grdd(i) = 0.0
         endif

   10 continue
      return
      END
      FUNCTION isum(vect,n)
C *******************************************************************
C This function sums all items in vector VECT from 1 to n and       *
C returns the summed number stored in variable ISUM.                *
C *******************************************************************
      integer vect(n)
      isum= 0
      if (n.le.0) return
      do 10 j= 1,n
           isum= isum+vect(j)
   10 continue
      return
      END
      FUNCTION risk(ck,dbh,j)
C ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C : Function RISK computes the probability of death from fire for    :
C : tree under consideration.  Equation is from Ryan and Rheinhardt  :
C : (1986).  Also presented is Bevins (1978) equation for small re-  :
C : generation.  Major variables are:                                :
C :   d1,d2,d3 ... coefficients for one year mortality equation      :
C :   bc(j) ...... thickness of bark in cm                           :
C :   c1,c2,c3,c4 ... coefficients for exponential equation.         :
```

```
C :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

      common/mort/ d1(7),d2(7),d3(7),bc(7)
      data d0/12.7/
      data r/10./,c1/1.466/,c2/-1.914/,c3/0.1792/,c4/0.000535/

C ..... Calculate the constants in the mortality equation
      a0 = d1(j)
      a1 = d2(j)*bc(j)
      a2 = d3(j)
      b0 = alog(r)+a0
      b1 = a1+2.*alog(r)/d0
      b2 = -alog(r)/d0**2

C ..... Mortality equation from Ryan and Rhienhardt 1986
        risk= 1./(1.+exp(-(c1+c2*bc(j)*dbh+c3*
     &        (bc(j)*dbh)**(2.0)+c4*ck**(2.0))))

C ***************************************************************
C ..... Previous mortality equation for trees under 5 in DBH ***
C     if (dbh.lt.d0) risk= 1.-1./(1.+exp(b0-b1*dbh-b2*dbh**2.0 *
C    &                        +a2*hs))                         *
C ***************************************************************
      return
      END
      FUNCTION sum(vect,n)
C :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C : Function SUM adds real elements 1 to n of an array.                :
C :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      real vect(n)
      sum= 0.
      if (n.le.0) return
      do 10 j= 1,n
           sum= sum+vect(j)
   10 continue
      return
      END
      FUNCTION itable(t)
C :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C This function computes the various properties of air at a        :
C specified temperature level.                                     :
C :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

      if(t .ge. 0.0 .and. t .lt. 250.0)     itable =  1
      if(t .ge. 250.0 .and. t .lt. 300.0)   itable =  2
      if(t .ge. 300.0 .and. t .lt. 350.0)   itable =  3
      if(t .ge. 350.0 .and. t .lt. 400.0)   itable =  4
      if(t .ge. 400.0 .and. t .lt. 450.0)   itable =  5
      if(t .ge. 450.0 .and. t .lt. 500.0)   itable =  6
      if(t .ge. 500.0 .and. t .lt. 550.0)   itable =  7
      if(t .ge. 550.0 .and. t .lt. 600.0)   itable =  8
      if(t .ge. 600.0 .and. t .lt. 650.0)   itable =  9
      if(t .ge. 650.0 .and. t .lt. 700.0)   itable = 10
      if(t .ge. 700.0 .and. t .lt. 750.0)   itable = 11
      if(t .ge. 750.0 .and. t .lt. 800.0)   itable = 12
      if(t .ge. 800.0 .and. t .lt. 850.0)   itable = 13
      if(t .ge. 850.0 .and. t .lt. 900.0)   itable = 14
      if(t .ge. 900.0 .and. t .lt. 950.0)   itable = 15
      if(t .ge. 950.0 .and. t .lt. 1000.0)  itable = 16
      if(t .ge. 1000.0 .and. t .lt. 1100.0) itable = 17
```

```
if(t .ge. 1100.0 .and. t .lt. 1200.0) itable = 18
if(t .ge. 1200.0 .and. t .lt. 1300.0) itable = 19
if(t .ge. 1300.0 .and. t .lt. 1400.0) itable = 20
if(t .ge. 1400.0)                     itable = 20
return
END
```

APPENDIX B: PRINTOUT OF THE EXTERNAL INPUT FILE TREE.DAT, WHICH CONTAINS VARIOUS SPECIES AND SITE PARAMETERS FOR EQUATIONS IN FIRESUM

Variables not defined in text are described in Keane and others (1989a) and Keane and others (1989b).

```
pipo
abgr
psme
pico
laoc
abla
pien
$$$$$$$$$$
hm          6562.500  5333.700  5715.000  4115.000  6857.500  4175.700  5456.700
dm           250.500   139.400   208.840   110.000   250.000   126.700   234.400
agemx        450.000   275.000   350.000   220.000   450.000   180.000   320.000
dmin        2249.900  2496.600  1810.400  1215.300  1817.400   801.800   801.400
dopt        4010.000  4200.000  4200.000  4200.000  4200.000  3800.000  4200.000
dmax        8608.000  7194.000  7194.000  6500.000  7194.000  6200.000  6200.000
spm            1.000     3.000     6.000     2.000     4.000     3.000     5.000
spm-cont       5.000
aside      3.5400000  2.040000  2.850000 3.5400000 3.5400000  2.040000 2.0400000
c              2.074     1.608     1.582     1.882     1.679     1.255     1.710
alpha      0.2680000 1.3090000 1.1370000 0.1220000 0.4370000 7.3450000 1.0400000
sigma         57.600    72.900    69.100    64.700   184.000    70.000    54.200
ap         0.5580000 1.5920000 0.4840000 0.4930000 0.3470000 0.5970000 0.5780000
betap     -0.0475000 0.0590000-0.0210000 0.0117000-0.0434000-0.0425000-0.0325000
cext          0.4260    0.5250    0.5250    0.4260    0.4260    0.4260    0.4260
cext cont     0.525
ishade             I         T         M         I         I         T         M
imoist             T         I         T         T         I         I         I
xmbar         0.0071    0.0089    0.0149    0.0074    0.0091    0.0107    0.0083
xmbar cont    0.0111
crat       0.4000000  0.800000  0.800000 0.4000000 0.4000000  0.800000 0.8000000
mext            .250
amc            1.651     1.651     1.651     1.651     1.651     1.651     1.651
bmc            0.493     0.493     0.493     0.493     0.493     0.493     0.493
cmc           19.350    19.350    19.350    19.350    19.350    19.350    19.350
dmc           10.880    10.880    10.880    10.880    10.880    10.880    10.880
mmc             .320      .320      .320      .320      .320      .320      .320
tmc           24.000
rhop            .510      .390      .390      .390      .510      .510      .510
bulk          0.0158    0.0088    0.0068    0.0080    0.0115    0.0071    0.0126
bulk cont     0.0080
lhv        18586.700 18586.700 18586.700 18586.700 18586.700 18586.700 18586.700
mps-fg1       57.410     8.890     3.480     0.950     3.156   91.8560    3.0000
mps-fg2       57.410    11.760     2.880     0.980     3.156   91.8560    3.0000
mps-fg3       57.410    16.000     3.077     0.980     3.156   91.8560    3.0000
mps-fg4       57.410    16.000     3.077     0.980     3.156   91.8560    3.0000
mps-fg5       57.410    16.000     3.077     0.980     3.156   91.8560    3.0000
mps-fg6       57.410    16.000     3.077     0.980     3.156   91.8560    3.0000
mps-fg7       57.410    11.760     2.880     0.980     3.156   91.8560    3.0000
mps-fg8       57.410    11.760     2.880     0.980     3.156   91.8560    3.0000
st              .055      .055      .055      .055      .055      .055      .055
se              .010      .010      .010      .010      .010      .010      .010
dkl            .1116     .0667     .1167     .1116     .2000     .0667     .0667
ltd            .5500     .6500     .6550     .6600     .8500     .6500     .6500
dkf            .0575     .0339     .0339     .0440     .1310     .0339     .0339
dkd            .2210     .2210     .2210     .2210     .3210     .2210     .2210
```

```
ffl               .1200      .1200      .1200      .1200      .1200     1.0000      .1200
dl                .1688      .1688      .1688      .1688      .1688      .1688      .1688
d2                1.969      1.969      1.969      1.969      1.969      1.969      1.969
d3                 .306       .306       .306       .306       .306       .306       .306
bc                 .070       .033       .065       .014       .071       .015       .022
rhop              0.513      0.513
bulk              0.001      0.001
lhv           18595.000  18595.000
mps              49.200     91.860
st                 .055      0.055
se                 .010      0.010
mois              1.000      1.500
ndyr                  4          7          5          3          1          7          6
ainc             0.0120     0.0050     0.0070     0.0150     0.0160     0.0080     0.0080
ws0                 .25        .47        .32        .38        .38        .65        .65
wsm                  1.         1.         1.         1.         1.         1.         1.
nws                   2          2          2          2          2          2          2
sura            10.5900    40.0100    38.6900    14.1200    20.1700    40.0100    40.0100
surb             2.7400     5.1150     4.2400     2.2800     5.5900     6.1150     6.1150
dbulk(1,i)      15.8000    36.2000    41.6000    21.9000    25.3000    35.0000    43.3000
  cont          38.1000
dbulk(2,i)      76.9000    76.9000   145.7900    76.9000   110.6300   110.6300   139.4800
 cont          142.7000
sburn            0.7500
crop             0.3950     0.3330     0.4460     0.3180     0.3680     0.3330     0.1670
cblock                2          2          1          2          2          2          3
ysc             20.0000    25.0000    20.0000    15.0000    25.0000    25.0000    25.0000
disequ(1j)      13.1251    13.4099    14.1251    12.6760    14.3257    13.4099    12.7470
disequ(2j)       0.0255     0.0183     0.0222     0.0376     0.0148     0.0183     0.0251
fyr-1hr              40         40         40         40         30         30         40
fyr-1hr              50
fyr-10hr             40         40         40         40         30         30         40
fyr-10hr             50
fyr-100hr            40         40         40         40         30         30         40
fyr-100hr            50
fload-1hr        0.0210     0.1350     0.2710     0.0638     0.0520     0.0520     0.1776
fload-1hr        0.0748
fload-10hr       0.0833     0.2650     0.1548     0.2619     0.1879     0.1879     0.4294
fload-10hr       0.1960
fload100hr       0.1546     1.6475     0.1055     0.5484     0.5635     0.5635     4.7022
fload100hr       0.5459
cbd               1.106      0.577      0.304      1.202      1.042       .000       .000
vfl               1.058      0.593      0.561      0.721      0.529       .000       .000
cfmc              1.050      1.050      1.050      1.050      1.050       .000       .000
vfmc               .100       .100       .100       .100       .100       .000       .000
cflm              1.000      1.000      1.000      1.000      1.000       .000       .000
csvr             60.700     64.700    184.000     72.900     54.200       .000       .000
vsvr             10.000     20.000     10.000     30.000     30.000       .000       .000
bl                0.1        0.10       0.1        0.100      0.100      0.100      0.100
binfest           0.660      0.800      0.800      0.450      0.640      0.660      0.440
bin(cont)         0.800
ibcycle              10         10         10         10         10         10         10
ibcycle c            10
rdelay           10.000     25.000      8.000     11.000     12.000      8.000     15.000
rdelay           20.000
cmax              600.0       60.0       20.0        3.0       58.8        3.7      0.800
icyr             0.3521     0.4673     0.7825     1.0000        7.0       60.0      0.120
end
```

APPENDIX C: PRINTOUT OF THE EXTERNAL INPUT FILE SITE.DAT, WHICH CONTAINS VARIOUS SITE PARAMETERS FOR EQUATIONS IN FIRESUM

Variables not defined in text are described in Keane and others (1989a) and Keane and others (1989b).

```
baset      17.8 21.1 23.5 31.7 40.1 51.0 64.0 63.0 55.0 38.9 26.5 21.4
basep      6.12 4.69 3.90 4.10 3.65 2.81 1.92 2.56 2.00 2.94 4.38 5.62
baseh       8000.000
excess         0.250
phi            1.000
text         133.300        site = Sabe Mountain
rock           0.100
elev        7200.000
ifg                2
till           3.000
rh             40.00
wind           3.200
ttheta         0.26
t             19.000
pltsiz       400.000
emc            0.080     0.080     0.080     0.080     0.100     0.080     0.100
dmoist         0.7500
brr            0.01
END
```

APPENDIX D: PRINTOUT OF THE EXTERNAL INPUT FILE CONTRL.DAT, WHICH CONTAINS VARIOUS INITIAL STAND PARMETERS THAT ARE USED TO CREATE THE SIMULATION STAND IN FIRESUM

Variables not defined in text are described in Keane and others (1989a) and Keane and others (1989b).

```
nspan        500
nruns          5
clear cut      0                 SITE: ONE HORSE RIDGE CLIMAX STAND
ifire        600
iblist       600
ibeetle      600
dsize        20.0   20.0   20.0   20.0   20.0   20.0   20.0
nwrstr         1
nbins         10
width        5.0
count 1       18     0     0     4     0
count 2        3     0     0     1     0
count 3        3     0     0     0     0
count 4        3     0     0     0     0
count 5        6     0     0     0     0
count 6        7     0     0     0     0
count 7        3     0     0     0     0
count 8        3     0     0     0     0
count 9        0     0     0     0     0
count10        2     0     0     0     0
age  1        38     0     0    65     0
age  2        96     0     0    53     0
age  3        60     0     0     0     0
age  4        70     0     0     0     0
age  5       300     0     0     0     0
age  6       250     0     0     0     0
age  7       350     0     0     0     0
age  8       370     0     0     0     0
age  9         0     0     0     0     0
age  10      450     0     0     0     0
END
```

*U.S . GOVERNMENT PRINTING OFFICE: 1989-0-673-039/01006

Keane, Robert E.; Arno, Stephen F.; Brown, James K. 1989. FIRESUM—an ecological process model for fire succession in western conifer forests. Gen. Tech. Rep. INT-266. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Research Station. 76 p.

Describes an ecological process model of succession that simulates long-term stand dynamics in forests of the Northern Rocky Mountains. This model is used to evaluate the effects of various fire regimes, including prescribed burning and fire suppression, on the vegetation and fuel complex of a simulation stand. This report documents the model FIRESUM (a **FIRE SU**ccession Model), examples of model output, and sensitivity analysis and validation results.

KEYWORDS: fire effects, fire regime, succession, documentation, computer program, wildland fire, fire management, fire ecology, forest succession, fire effects, fire regime

INTERMOUNTAIN RESEARCH STATION

The Intermountain Research Station provides scientific knowledge and technology to improve management, protection, and use of the forests and rangelands of the Intermountain West. Research is designed to meet the needs of National Forest managers, Federal and State agencies, industry, academic institutions, public and private organizations, and individuals. Results of research are made available through publications, symposia, workshops, training sessions, and personal contacts.

The Intermountain Research Station territory includes Montana, Idaho, Utah, Nevada, and western Wyoming. Eighty-five percent of the lands in the Station area, about 231 million acres, are classified as forest or rangeland. They include grasslands, deserts, shrublands, alpine areas, and forests. They provide fiber for forest industries, minerals and fossil fuels for energy and industrial development, water for domestic and industrial consumption, forage for livestock and wildlife, and recreation opportunities for millions of visitors.

Several Station units conduct research in additional western States, or have missions that are national or international in scope.

Station laboratories are located in:

Boise, Idaho

Bozeman, Montana (in cooperation with Montana State University)

Logan, Utah (in cooperation with Utah State University)

Missoula, Montana (in cooperation with the University of Montana)

Moscow, Idaho (in cooperation with the University of Idaho)

Ogden, Utah

Provo, Utah (in cooperation with Brigham Young University)

Reno, Nevada (in cooperation with the University of Nevada)