

AD-A091 812

FOREST PRODUCTS LAB MADISON WI  
PROGRAMS FOR COMPUTER SIMULATION OF HARDWOOD LOG SAWING.(U)

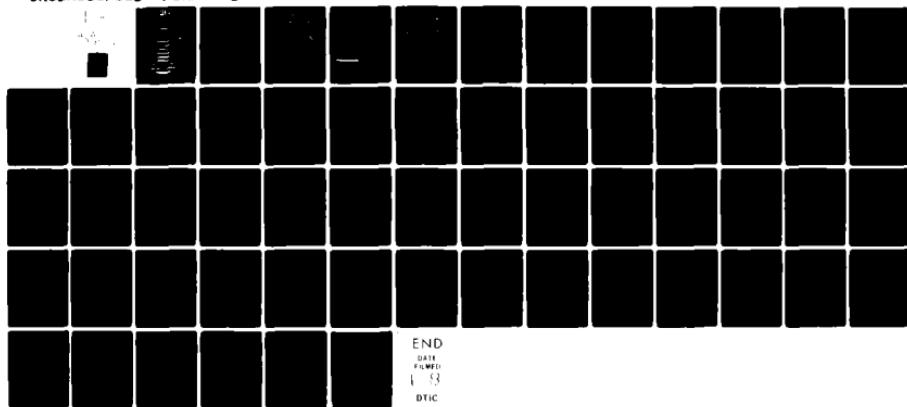
F/G 13/8

SEP 80 W K ADKINS, D B RICHARDS, D W LEWIS

UNCLASSIFIED

FSRP-FPL-357

NL



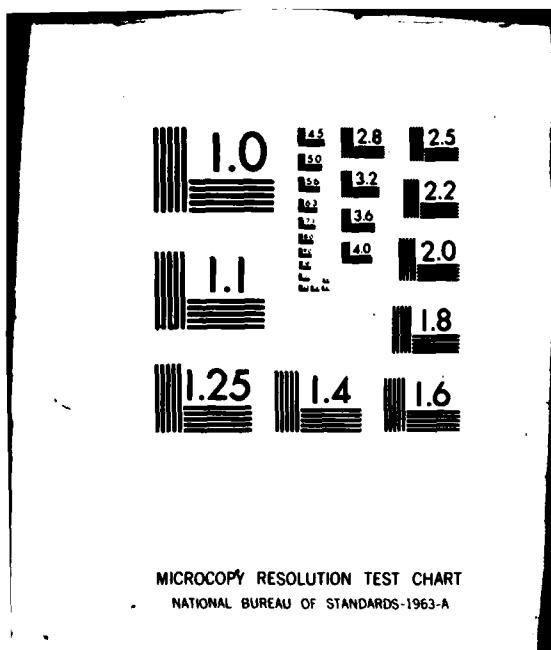
END

DATE

FILED

1-13

DTIC



United States  
Department of  
Agriculture  
Forest Service  
Forest  
Products  
Laboratory  
Research  
Paper  
FPL 357  
September 1980

LEVEL  
Programs for Computer  
Simulation of  
Hardwood Log Sawing

12

AD A 091 812

DTIC  
ELECTED  
NOV 18 1980

DISTRIBUTION STATEMENT A  
Approved for public release;  
Distribution Unlimited

DOC FILE COPY

8011 10 143

## Preface

This Research Paper is one in a series of three which describe the computer simulation of hardwood log sawing. Mathematically modeled logs with a selection of tapers, diameters, core defect diameters, and knot patterns were sawn by four sawing methods, and the resultant values were recorded.

The first paper, USDA Forest Service Research Paper FPL 355, "Simulation of hardwood log sawing," describes the sawing methods, and the background and development of these programs.

The second paper, FPL 356, "Lumber values from computerized simulation of hardwood log sawing," presents the results of the sawing in terms of volume yield and lumber value, and compares them for the four sawing methods.

This third paper, FPL 357, "Programs for computer simulation of hardwood log sawing," lists the programs, model assumptions, and program organization and variables.

## Abstract

Four computer programs were developed at the University of Kentucky as simulation models for investigating factors affecting sawn log value over four hardwood sawing methods: quadrant sawing, cant sawing, decision sawing, and live sawing with rmpf for grade. The programs are listed along with information on the sawing methods, model assumptions and program organization.

## Keywords

Computer simulation  
Mathematical modeling  
Hardwood sawing  
Computer programs  
Quadrant sawing  
Cant sawing  
Live sawing  
Decision sawing  
Grade sawing  
Grade yield

6  
United States  
Department of  
Agriculture  
Forest Service  
Forest  
Products  
Laboratory<sup>1</sup>  
Research  
Paper  
FPL 357

# Programs for Computer Simulation of Hardwood Log Sawing.

10 By

W. KADKINS  
D. B. RICHARDS  
D. W. LEWIS

Forest Products Technologist

E. H. BULGRIN Supervisory Forest Products Technologist

11

Sep 18

12

3/1

14 FSRP-FPI-357

DTIC  
ELECTED  
NOV 1 8 1980  
S C

## Introduction

Four computer programs were developed at the University of Kentucky as simulation models for investigating factors affecting sawn log value over four hardwood sawing methods: quadrant sawing, cant sawing, decision sawing, and live sawing with riper for grade.

Information in this paper details the sawing methods, model assumptions, program organization, variables used, common storage areas, and program listings.

## Sawing Methods

Cant and quadrant sawing are 4-sided sawing methods. Decision sawing is a possibly unbalanced 4-sided sawing method which responds to uncovered hidden defects by attempting to pick the log face which will yield the best lumber when the log face currently being cut drops in grade. Live sawing with riper for grade is a 2-sided sawing method which attempts to increase board value by ripping out core defects.

## Quadrant Sawing (QUAD)

The center cant in QUAD was arbitrarily picked to be a square cant that will yield four boards. Progressing outward from this central square cant, the boards increase in width in each quadrant in a stepwise fashion until the bark (i.e., log surface) is reached and then decrease appropriately to fit in the slab (fig. 1). In the program each quadrant is cut completely before progressing to another, but the board widths and the way they fit together at the corners are the same as would result if the log were turned about its axis after each board were cut, 180° turns alternating with 90° turns until the central square cant remains and is sawn into four boards.

## Cant Sawing (CANT)

By cutting slabs and boards from faces 1 and 3 in CANT, a central cant is produced that has a selected thickness. This central cant is then turned 90° and sawn into boards kerf-centered from the log axis out (fig. 2). While in current studies the central

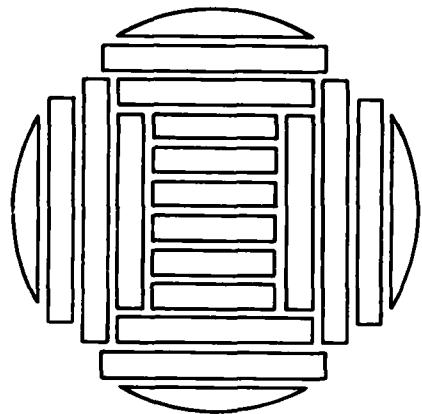
cant was arbitrarily given a thickness of 2 inches less than half the log diameter [(D/2) - 2], it can, of course, be assigned any reasonable thickness.

## Decision Sawing (DECID)

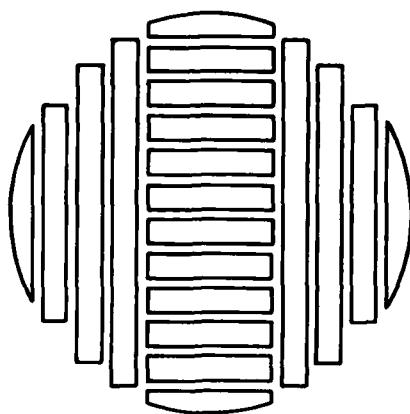
DECID simulates the decisions of a human sawyer in grade sawing. Faces 1, 2, 3, and 4 of the log are sawn in sequence until the log is square and wane-free at midlength. Each exposed face of the log is then graded by the Forest Products Laboratory (FPL) computerized grading program, the highest grade face is selected for sawing (surface area is used to decide ties), and the selected face is sawed until a grade drop occurs. The program again grades every affected face and selects the highest grade face for sawing (surface area decides ties) and continues sawing until a grade drop occurs. Log turning and sawing

<sup>1</sup> Maintained at Madison, Wis., in cooperation with the University of Wisconsin.

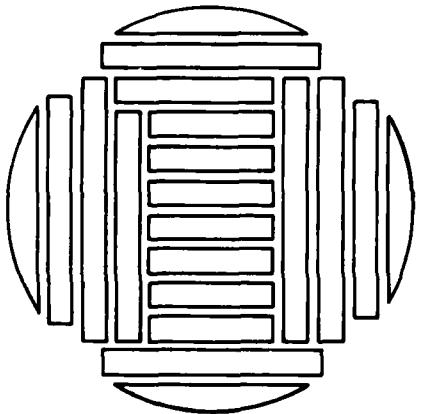
<sup>2</sup> Former graduate student, Dept. of Computer Science, and Professor of Forestry, Dept. of Forestry, University of Kentucky, Lexington, Ky.



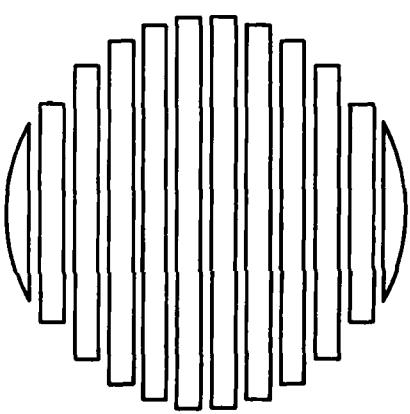
**Figure 1.**—End view of a quadrant sawn (QUAD) log.  
(M 148 124)



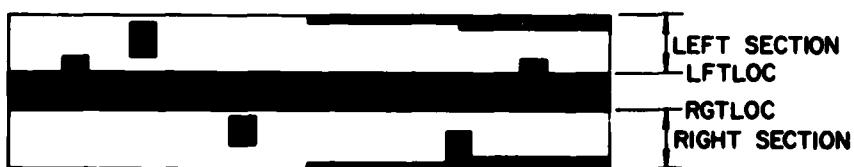
**Figure 2.**—End view of a cant sawn (CANT) log.  
(M 148 326)



**Figure 3.**—End view of a decision sawn (DECID) log.  
(M 148 325)



**Figure 4.**—End view of a live sawn (LIVE) log.  
(M 148 327)



**Figure 5.**—Outer surface of a board that has been reripped for grade showing the coordinate system of the resulting three board sections as well as the truncation of any knots at the edge of a section.  
(M 148 126)

continues in like manner until a central square cant remains that will yield exactly four equal boards when parallel sawed. As noted earlier, some of the boards in DECID may be unsymmetrical with respect to the log axis (fig. 3).

### **Live Sawing with Reripping for Grade (LIVE)**

A saw kerf bisects the log along the central axis in LIVE and the plane of each subsequent saw cut (and hence each board face) is parallel to this central cut (fig. 4).

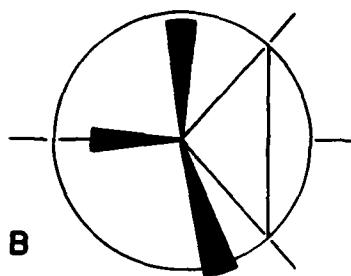
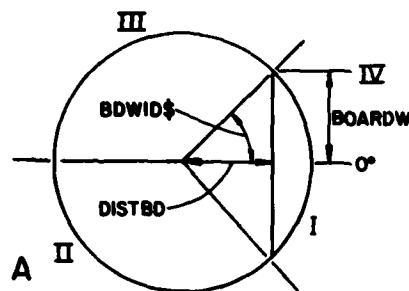
In the live rip method the log is sawed as in LIVE but the outer face of each board is then evaluated for defect type. If the central cylindrical core defect shows up on the outer face of the board, this defect is automatically ripped out and the boards produced are regraded and revalued (fig. 5). If the sum of the values of the boards so produced is higher than the value of the original wide board, the new rerip sum is substituted for the original wide board value. If the rerip value is less than the original wide board value, then the original value is retained and it is assumed that the board would not have been reripped. The rerip subroutine is applied in sequence to each board that has the central core defect appearing on the outer board face.

### **Model Assumptions**

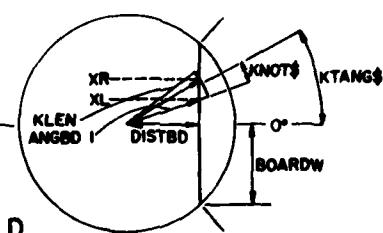
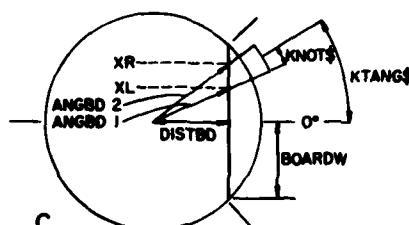
**Log:** A truncated cone with arbitrary length, small-end diameter, and taper. The length is measured in inches, the diameter in inches, and the taper in degrees.

**Knot:** A solid conic section of a sphere (i.e., a cone capped with the spherical surface) with arbitrary length, height in the log, and interior angle, emanating perpendicularly from the log axis. Length is measured in inches from the log axis, height in inches from the log base, and angle in degrees clockwise with zero degrees being the line from the log axis perpendicular to the initial saw face (figs. 6a-f).

**Core Defect:** A solid cylinder extending the full log length, which may or may not be centered on the log axis. Core diameter and linear offset from the log axis are measured in inches; angular offset



M 148 317

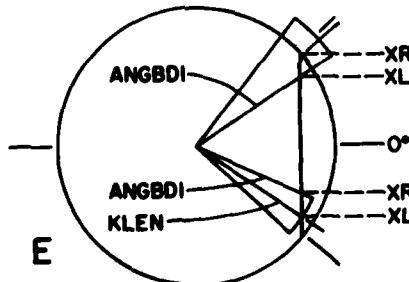


M 148 318

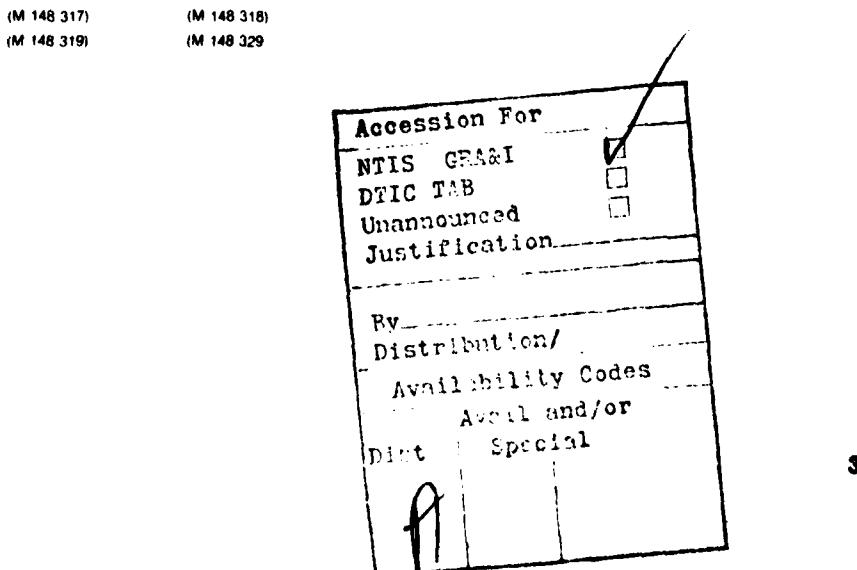
**Figure 6.—Method of evaluating knots which partially intersect a board face:**  
 (a) the sectors into which the log section is divided for evaluation of knot defect location on the board face; (b) knots that fall wholly in sectors II and III and hence do not intersect board face; (c) the modeling of a knot wholly in sector IV that fully penetrates the board face; (d) a knot that partially penetrates the board face due to the short length of the knot; (e) one knot that only partially penetrates the board face due to the fact that it is partially in sector III and another knot with partial penetration due to combination of short knot and angular position; (f) the pattern on the board of several knots that show partial penetration of various types.

(M 148 317)  
 (M 148 318)

(M 148 319)  
 (M 148 329)



M 148 319



Accession For	
NTIS GRA&I	
DTIC TAB	
Unannounced	
Justification	
By _____	
Distribution/	
Availability Codes	
Avail and/or	
Dist Special	
<i>[Signature]</i>	

in degrees from the zero degree line defined above for knots (figs. 7, 8).

**Edging Method:** All boards are edged so that they are wane-free for at least one-half their length by assigning the width at midlength as the maximum board width (figs. 9, 10).

**Defect Representation:** All defects are measured in  $\frac{1}{4}$ -inch increments and placed on a board face as rectangles with dimensions of the maximum length and width of the actual defect (fig. 9).

**Board:** Rectangular with some width  $\geq$  3 inches at mid-length; some width  $\geq$  2.5 inches at the top; some length  $\geq$  4 feet; and arbitrary thickness. In addition, the width of wane at any point along the board is limited to less than 4 inches. The board is cut back in length by 1-foot decrements until wane and width requirements are met. Length, width, and thickness are measured in inches. The length and width are converted to  $\frac{1}{4}$ -inch increments for board face grading (figs. 9, 10).

**Wane:** The first wane defect on a board edge begins at or near board midlength and extends to the minimum of either the board length or that point where the board halfwidth has decreased by  $\frac{1}{4}$  inch. The next wane defect begins at the point the last one ended and extends to the next point of  $\frac{1}{4}$ -inch decrease; wane defects continue to be inserted in like manner until the end of the board is reached (figs. 9, 10).

**Sawing:** The log is sawed parallel to the log axis. Headsaw kerf and rerip saw kerf (live sawing with rerip only) are independently arbitrary and are measured in inches. The log is completely sawn for each of twelve  $15^\circ$  rotational increments of initial placement on the log carriage. The total log value and surface measure yield, as well as the surface measure in each of the permissible grades, is calculated for each rotational increment. High, low, and average values and yields are calculated for the 12 rotational positions.

**Grading:** Each board face with defects is graded by the FPL grading program. The final board grade is assigned based on the combination of grades of both faces. Possible grades are First

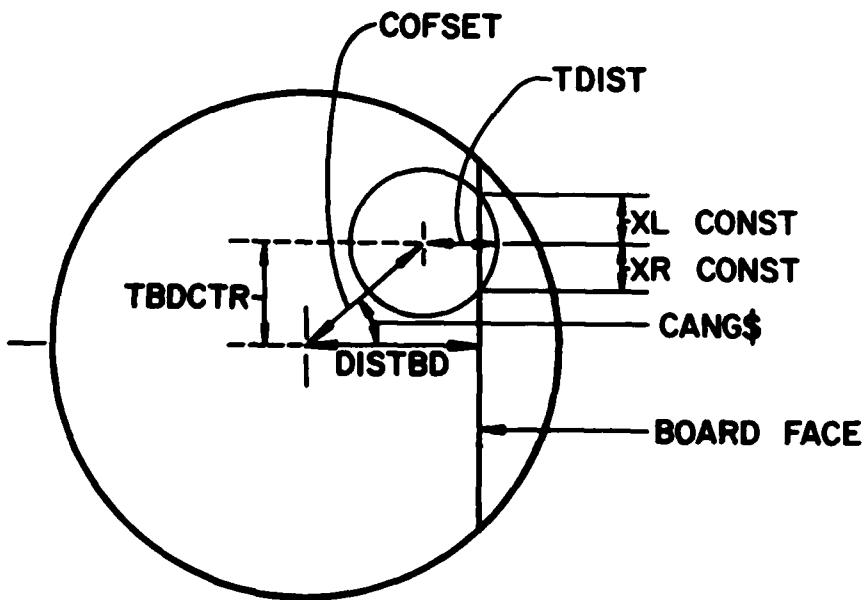


Figure 7.—A log section showing the core defect displaced from the central axis of the log.

(M 148 320)

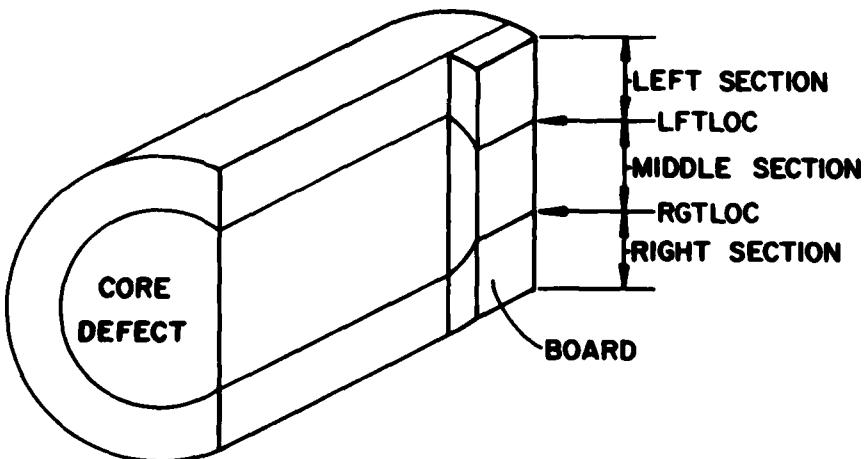


Figure 8.—Cross section of a log showing how the core defect is ripped out of a board if it appears on the outer face of a board. Although the ripping kerf is not shown, it is always taken out of the defective middle section.

(M 148 321)

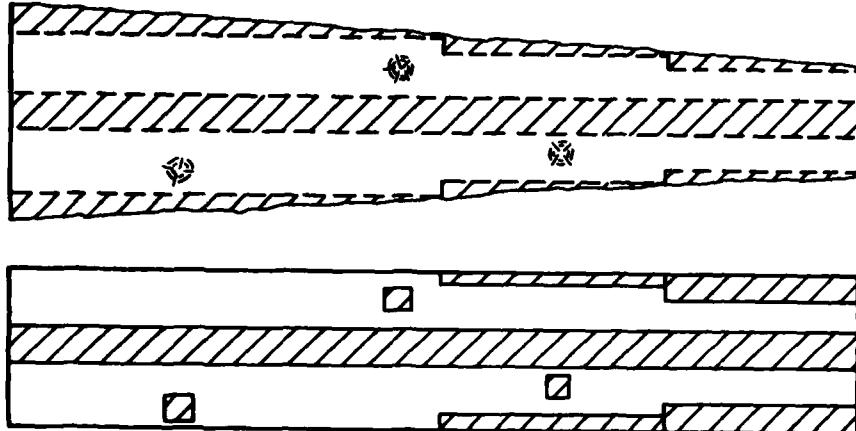


Figure 9.—The manner in which the computer would take the unedged board with knots (top view) and model it as an edged board with rectangular wane, knot, and core defects (bottom view).

(M 148 382)

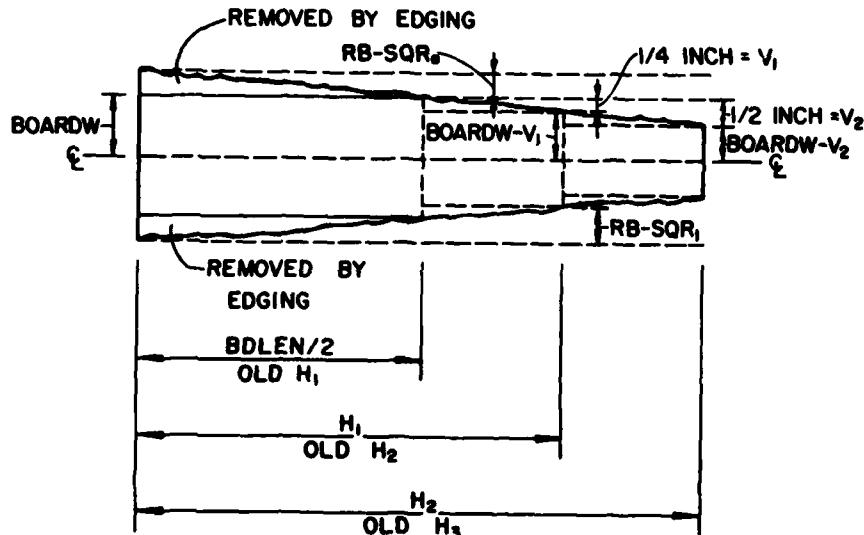


Figure 10.—The outer face of a waney board showing how subroutine WANE edges the board and puts in the wane defects.

(M 148 323)

Table 1.—Board grade assigned by PRICE based on the grade of each board face returned from subroutine GRADE

Grade of second board face	Grade of one board face				
	FAS	Select	1C	2C	Below 2C
FAS	FAS	—	FAS1F	2C	Below 2C
Select	—	1C	1C	2C	Below 2C
1C	FAS1F	1C	1C	2C	Below 2C
2C	2C	2C	2C	2C	Below 2C
Below 2C	Below 2C	Below 2C	Below 2C	Below 2C	Below 2C

\* The NHLA Hardwood Standard Grades allow the option of grading using either the grade of "Selects" or "FAS1F." The FPL grading program was designed to use the "Selects" grade. This sawing simulation program uses the "FAS1F" grade which is determined as shown above.

and Seconds (FAS), either FAS One Face or Selects, One Common (1C), Two Common (2C), and combined 3A/3B. The value of lumber in each grade is user supplied and is measured in dollars per square foot surface measure. The current version of the grading program allows 22 defects per board face. Faces with defects exceeding this are arbitrarily assigned a grade of 3A/3B and a message printed.

## Program Organization

Input, initialization, program control, and printing of results are performed in the main program. Cutting a board from the log and the mathematical description of the resulting board is performed by subroutine KERF. Wane defects are located by subroutine WANE (fig. 10). Knot defects are located by subroutine KNOT (figs. 6a-f). Core defects are located by subroutine CORE (fig. 7). Board value is determined by subroutine PRICE (table 1). Best log face determination in decision sawing is performed by subroutine DECIDE. Rerip for grade in live sawing is performed by subroutine RERIP, which prepares and updates rerip parameters (fig. 8), and RIP, which prepares the grading linkages for each rerip piece (fig. 5). Linkage to the grading subroutine is performed by subroutine GRADE. Subroutine GRADE is the FPL grading program converted for use on an IBM 370/165. For a description, refer to USDA Forest Service Research Paper FPL 157, "Grading Hardwood Lumber by Computer," and "Computer Program for Grading Hardwood Lumber," by Lynn Galiger and Hiram Hallock, both available from the Forest Products Laboratory, Madison, Wis.

## Program Variables

The variables appear in all programs unless otherwise noted.

PROGRAM LEGEND: C (CANT), Q (QUAD), L (LIVE), D (DECID).

A:	argument to functions DEG and RAD (degrees or radians)
ANGBD1:	distance from log axis to board face along the knot edge nearest the board face (in.) (figs. 6c-e)
ANGBD2:	distance from log axis to board face along the knot edge farthest from the board face (in.) (fig. 6c)
ANGLE:	array containing knot horizontal angles within the log (degrees) (user supplied)
AVGS\$	average log value obtained over the 12 rotational positions on the carriage (\$)
AVRIP (L):	average percent of available log surface measure realized under riper over the 12 rotational positions on the carriage
AVRIP\$ (L):	average log value obtained under riper over the 12 rotational positions on the carriage (\$)
AVTOTS (D):	average percent of available log surface measure realized over the 12 rotational positions
BDLEN:	current board length (in.) (fig. 10)
BDWIDS:	current board angular halfwidth at board midlength measured from a line from the log axis perpendicular to the board face (degrees) (fig. 6a)
BF:	board surface measure of the current board
BOARDW:	current board halfwidth measured at board midlength (in.) (figs. 6a, 6c, 6d, 10)
CANG\$:	core defect angular offset measured from a line perpendicular to the board face (degrees) (user supplied) (fig. 7)
CANT (C):	cant size (in.)
CMPLET (D):	array which indicates which log faces are completely cut
COFSET:	core defect linear offset from log axis (in.) (user supplied) (fig. 7)
CONST:	core defect halfwidth on current board face (in.) (fig. 7)
CON1:	calculation constant
CON2:	calculation constant
CORDIA:	core defect diameter (in.) (user supplied)
CORFLG (L):	a flag which indicates whether or not a core defect was found on the current board outer face
CQUENS (D):	array which records the sequence of log face cutting
CRADUS:	core defect radius (in.)
D:	log small-end diameter (in.) (user supplied)
DAT:	contains the current date returned by the DATE function (character string)
DEG:	result of conversion to degrees of an argument supplied in radians (degrees)
DISTBD:	distance from log axis to current board face (in.) (figs. 6a, 6c, 6d, 7)
DSTLFT (D):	distance from log axis to the log face counterclockwise from the current face (in.)
DSTRGT (D):	distance from log axis to the log face clockwise from the current face (in.)
DUMYRP (L):	array which contains the surface measures of the reripped board sections (fbm)
DY:	current knot defect halflength along the length of the current board
EXCESS (D):	discarded slab portion on each face which is outside the first saw cut (in.)
F:	face of the log which is currently being cut
FAS:	value per board foot of FIRST and SECONDS (\$) (user supplied)
FLFT (D):	log face counterclockwise from the current log face
FLX:	array which contains all defect lower X-coordinates on the current board face (1/4-in. units)
FLY:	array which contains all defect lower Y-coordinates on the current board face (1/4-in. units)
FRT (D):	log face clockwise from the current log face
FULWID:	calculated halfwidth of the current knot defect on the board face if a full intersection were to occur
FUX:	array which contains all defect upper X-coordinates on the current board face (1/4-in. units)

**FUY:** array which contains all defect upper Y-coordinates on the current board face (1/4-in. units)  
**GDBEST (D):** grade of the current best log face  
**GDORDR (D):** array which records the board grades in the sequence they are cut from the log  
**GRDCOM (D):** array which contains the current log face grades  
**GRDLFT (L):** grade of left rerip section of the outer face of the current board  
**GRDMID (L):** grade of center rerip section of the outer face of the current board  
**GRDRGT (L):** grade of right rerip section of the outer face of the current board  
**H:** height of the end of the current wane defect on the current board face (1/4-in. units) (fig. 10)  
**HEIGHT:** array which holds the height of the knots measured from log base (in.) (user supplied)  
**HIGH\$:** highest log value obtained over the 12 rotational positions on the carriage (\$)  
**HIRIP (L):** percent of available log surface measure realized by reripping at the position on carriage which resulted in the highest log value while reripping (%)  
**HIRIPS\$ (L):** highest log value obtained from reripping over the 12 rotational positions on the carriage (\$)  
**HITOT (D):** percent of available log surface measure realized at the position on the carriage which resulted in the highest log value (%)  
**I:** loop counter which indicates face of the current board under consideration  
**ID:** array which records types of defects on the current board face  
**II (D):** loop counter used while regrading affected log faces  
**IMGRD (L):** grade of center rerip section of the outer face of the current board  
**INDEX (L):** index of the core defect in the defect array  
**INX:** counter for the 12 rotational increments of the log position on the carriage  
**IOLDLT (L):** grade of left rerip section of the outer face of the current board  
**IOLDRT (L):** grade of right rerip section of the outer face of the current board  
**IREGRD (D):** flag which indicates whether a log face is being regraded  
**IROTAT:** current log position on the carriage for the 12 rotational increments (degrees)  
**ISQUAR (D):** flag which indicates whether the log has been squared at midlength  
**ITYPE (L):** array which saves defect types in the current board during rerip  
**IX:** array used for passing defect type to FPL grading program  
**J:** general do-loop counter  
**JCMPLT (D):** counts log faces which are completely cut  
**JJ (D):** do-loop counter  
**K:** kerf size (in.) (user supplied)  
**KHIGH:** height in the log of the current knot (in.)  
**KLEN:** length of the current knot (in.) (figs. 6d, 6e)  
**KNOT\$:** knot half-angle (degrees) (user supplied) (figs. 6c, 6d)  
**KTANG\$:** angle of the current knot with respect to a line perpendicular to the current board face (degrees) (figs. 6c, 6d)  
**KTLEN:** array which contains the knot lengths (in.) (user supplied)  
**L:** log length (in.) (user supplied)  
**LFTLOC (L):** current board left rerip location (1/4-in.) (figs. 5, 8)  
**LGRADE (L):** grade of the current board face left rerip section  
**LOG\$:** half-angle of log taper (degrees) (user supplied)  
**LORIP (L):** percent of available log surface measure realized by reripping at the rotational position on the carriage which yielded the least log value while reripping  
**LRADUS:** log radius at the height of the current knot (in.)  
**LVOL:** total available log surface measure (ft<sup>2</sup>)  
**LX (L):** array which saves defect lower X-coordinates in the current board during rerip  
**LY (L):** array which saves defect lower Y-coordinates in the current board during rerip  
**M:** current number of defects found in the current board  
**MAX:** number of knots in the log (user supplied)  
**N (L):** current number of defects found in the current rerip section

## Program Variables (Cont.)

NBD:	board counter
NBRDS (D):	number of boards which could ideally be cut from each log face
NODEFC:	array which contains the current number of defects found in the current board face
NPG:	grade of the current board face
OLD:	grade of the outer face of the current board
OLDH:	beginning height of the current wane defect (1/4-in. units) (fig. 10)
OLDVAL (L):	value of the current board before rerip (\$)
ONEC:	dollar value per board foot of 1 Common lumber (\$) (user supplied)
PERC:	total surface measure in each grade realized at the current rotational position on the carriage ( $\text{ft}^2$ )
PERCRP (L):	total surface measure in each grade realized by reripping at the current rotational position on the carriage ( $\text{ft}^2$ )
POSLEN:	board length of those boards which may be shorter than log length due to log taper (in.)
R:	that position of the log face remaining to be cut (in.)
RB:	log radius at the base (in.) (fig. 10)
RCUT (D):	array which contains the portion of each log face which has been cut (in.)
RGRADE (L):	grade of the current right rerip section
RGTLOC (L):	current board right rerip location (1/4-in. units) (figs. 5, 8)
RI:	log top end radius (in.)
RM:	log radius at the height of the middle of the current board (in.)
RPKERF (L):	rerip kerf size (in.) (user supplied)
RPLOSS (L):	current surface measure loss due to rerip ( $\text{ft}^2$ )
RPTOTS (L):	total surface measure realized by reripping at the current rotational position on the carriage ( $\text{ft}^2$ )
RT:	log radius at the top of the current board (in.)
RTBCUT (D):	the portion of each log face which may be cut (in.)
SEL:	dollar value per board foot of Selects lumber (\$) (user supplied)
SM:	rounded surface measure ( $\text{ft}^2$ )
SMALL\$:	smallest log value obtained during the 12 rotational positions on the carriage (\$)
SMBEST (D):	surface measure of the current best log face ( $\text{ft}^2$ )
SMF (D):	array which contains the current surface measure of each log face
SMLFT (L):	surface measure of the current left rerip section ( $\text{ft}^2$ )
SMMID (L):	surface measure of the current center rerip section ( $\text{ft}^2$ )
SMRGT (L):	surface measure of the current right rerip section ( $\text{ft}^2$ )
SMRIP\$ (L):	smallest log value obtained by reripping at any one of the 12 rotational positions on the carriage (\$)
SMTOT (D):	total surface measure realized in the current rotational position ( $\text{ft}^2$ )
SQR:	calculation constant (fig. 10)
T:	board thickness (in.) (user supplied)
TBDCTR:	distance on the current board face from board midwidth to the center point of the intersection of the board face and the core defect (in.) (fig. 7)
TCANG\$ (D):	core angular offset adjusted to reflect the face of the log currently being cut (degrees)
TDIST:	perpendicular distance from the center of the core defect to the current board face (in.) (fig. 7)
THR\$:	dollar value per board foot for combined 3A/3B lumber (\$) (user supplied)
TLOSS (L):	rerip surface measure loss in the current board due to a rerip section falling below 3 inches in width ( $\text{ft}^2$ )
TOTS:	used for surface measure and percent surface measure conversions
TVAL:	total log value obtained at the current rotational position on the carriage (\$)
TVALRP (L):	total log value obtained by reripping at the current rotational position on the carriage (\$)

TWOC: dollar value per board foot for 2 Common lumber (\$) (user supplied)  
UX (L): array which saves defect upper X-coordinates of the current board face during rerip  
UY (L): array which saves defect upper Y-coordinates of the current board face during rerip  
V: current board value (\$)  
WAN: total wane on the current board face from board mid-length to board end, subroutine WANE  
only (1/4-in. units) (fig. 10)  
WIDLFT (L): width of the current left rerip section (in.)  
WIDMID (L): width of the current center rerip section (in.)  
WIDRGT (L): width of the current right rerip section (in.)  
WIDTH (L): saves current board width during rerip (1/4-in. units)  
WMB (Q): current board halfwidth at board midlength when the log has been squared (in.)  
WT: current board halfwidth at the top of the board (in.)  
XL: current defect lower X-coordinate on the current board face (1/4-in. units) (fig. 6d, 6e, 7)  
XR: current defect upper X-coordinate on the current board face (1/4-in. units) (fig. 6d, 6e, 7)  
YB: current defect lower Y-coordinate on the current board face (1/4-in. units)  
YT: current defect upper Y-coordinate on the current board face (1/4-in. units)

## Common Storage Areas

BOARD: contains information pertaining to the log and boards cut from the log  
DECID (Decision sawing only): contains information necessary in the determination of the log face which should yield the best board  
DEFEC: contains information of all defects on the current board face  
FORRIP (Live sawing only): contains information used in rerip for grade  
PRICE: contains pricing information for lumber in each grade

## Programs

### Quadrant Sawing (QUAD)

```

1 C *QUAD* IS A BALANCED METHOD OF SAWING AROUND A CENTRAL SQUARE CANT
2 C CONTAINING EXACTLY FOUR BOARDS. THE PROGRAM CUTS FROM THE CENTRAL
3 C CANT OUTWARD AND COMPLETES ONE QUADRANT BEFORE PROGRESSING TO
4 C ANOTHER
5 C
6 C IMPLICIT REAL (K,L)
7 C REAL FLX(.26,.21),FLY(.26,.2),FLY(.26,.2),PERC(.5),ANGLE(.48),
8 C HEIGHT(.48),KTLEN(.48),DAT48
9 C INTEGER ID(22,2),NODEFC(2),OLD,F
10 C COMMON /DEFECT/ NODEFC,FLX,FLY,FUY,FLY,LD
11 C /BOARD/ L,D,K,T,LOGS,RB,DISTBD,BOLEN,BOARDJ,KNOTS,R,BF
12 C /PRIM/ ,FAS,SEL,ONEC,TUDC,THR8
13 C
14 C
15 C THIS PROGRAM ASSURES INPUT FROM UNIT 5 AND OUTPUTS TO UNITS 6 AND 8
16 C
17 C GET LOG DIAMETER (IN.), LOG LENGTH (IN.), KERF SIZE (IN.). AND
18 C BOARD THICKNESS (IN.)
19 C
20 C
21 C 1 READ (5,100) END=17) D
22 C READ (5,100) L
23 C READ (5,101) K
24 C READ (5,100) T
25 C
26 C GET KNOT HALF-ANGLE (DEGREES) AND LOG HALF-TAPER (DEGREES)
27 C
28 C READ (5,100) KNOTS
29 C READ (5,100) LOGS
30 C
31 C GET CORE DEFECT PARAMETERS: DIAMETER (IN.), LINEAR OFFSET (IN.),
32 C AND ANGULAR OFFSET (DEGREES)
33 C
34 C READ (5,116) CORDIA
35 C READ (5,116) COFSET
36 C READ (5,116) CINC5
37 C
38 C GET PRICE PER BOARD FOOT FOR EACH GRADE: NUMBER OF KNOTS
39 C
40 C READ (5,103) FAS,SEL,ONEC,TUDC,THR8
41 C READ (5,113) MAX
42 C
43 C GET DATE AND CALCULATE RADIUS OF COKE DEFECT
44 C
45 C CALL DATE (DAT)
46 C CRADUS = CORDIA/2.
47 C
48 C GET KNOT ANGLES, HEIGHTS, AND LENGTHS
49 C
50 C DO 2 J=1,MAX
51 C READ (5,117) ANGLE(J),HEIGHT(J)
52 C
53 C 2 CONTINUE
54 C READ (5,116) KTLEN(J)
55 C
56 C CALCULATE LOG RADIUS AT TOP AND BOTTOM: TOTAL LOG CUBIC FEET
57 C
58 C RI = D/2.
59 C PB = PI*4*ATAN(RAD(LOCS))
60 C LVOL = 1/3.*3.141593*(RI*RI*2*PI*H*RI*H*RI*H)/144.
61 C
62 C
63 C OUTPUT HEADER
64 C
65 C WRITE (6,104) DAT
66 C WRITE (6,105) L,D,LOGS,LVOL
67 C WRITE (6,119) MAX
68 C DO 4 J=1,MAX
69 C WRITE (6,115) ANGLE(J),HEIGHT(J),KTLEN(J)
70 C
71 C 4 CONTINUE
72 C WRITE (6,106) KNOTS
73 C WRITE (6,102) CRADUS,COFSET,CANGS

```

```

74 C
75 C WRITE (6,108) FAS,SEL,ONEC,TUDC,THR8
76 C WRITE (6,200) L,D,MAX,CRADUS,COFSET,CANGS,L,VOL,K,T,KNOTS,LOGS
77 C WRITE (6,201) (KTLEN(J),J=1,MAX)
78 C WRITE (6,201) (ANGLE(J),J=1,MAX)
79 C WRITE (6,202) FAS,SEL,ONEC,TUDC,THR8
80 C
81 C INITAILIZE VARIABLES FOR CALCULATING HIGH, LOW, AND AVERAGE
82 C YIELDS FOR EACH LOG
83 C
84 C HIGHS = 0.
85 C SMALLS = 999.
86 C AVGS = 0.
87 C
88 C
89 C SAW LOG IN 12 DIFFERENT ROTATIONAL POSITIONS. 15 DEGREES APART. TO
90 C DETERMINE THE ORIENTATION GIVING THE HIGHEST YIELD
91 C
92 DO 16 IMX=1,12
93 C
94 C INITAILIZE FOR EACH POSITION
95 C
96 C
97 C TVAL = 0.0
98 C NBID = 0
99 C DO 5 J=1,5
100 C PERC(J) = 0.0
101 C
102 C 5 CONTINUE
103 C
104 C FOR FOUR SIDES OF THE LOG...
105 C DO 13 F=1,4
106 C
107 C INITAILIZE PORTION OF LOG TO BE CUT... THIS SETS THE DEPTH OF THE
108 C FIRST CUT. THE SLAB FACES ARE CUT TO A FOUR BOARD CANT
109 C
110 C IF (F,EQ.1,OP,F,ED,3) R = RB-2.*T-1.5*K
111 C
112 C THE CANT FACES ARE KERF CENTERED TO DIVIDE THE CANT
113 C
114 C IF (F,EQ.2,OR,F,ED,4) R = RB+K/2.
115 C
116 C CUT BOARDS FROM THIS FACE UNTIL FACE IS COMPLETELY CUT. THE LOG IS
117 C CUT FROM THE INSIDE OUT
118 C
119 C CALL PERFO (6,11,NBDF,F)
120 C
121 C FOR BOTH SIDES OF THE BOARD...
122 C
123 DO 10 I=1,2
124 C
125 C INITAILIZE DEFECT ARRAYS, EXCEPT F.. (26,1). WHICH HOLDS THE BOARD
126 C DIMENSIONS
127 C
128 DO 7 J=1,25
129 C
130 C FLX(J,1) = -9999.0
131 C FUX(J,1) = -9999.0
132 C FUY(J,1) = -9999.0
133 C
134 C CONTINUE
135 C DO 8 J=1,22
136 C ID(J,1) = 0
137 C
138 C NODEFC(1) = 0
139 C
140 C PUT IN LINE EFFECTS
141 C
142 C CALL LANE0
143 C
144 C PUT IN KNOTS

```

```

145      DO 9 J=1,MAX
146      CALL KNOTO (HEIGHT(J),KTLEN(J),ANGLE(J))
147      9    CONTINUE
148      C    PUT IN CORE DEFECT
149      C
150      CALL COPE (CRADIUS,CDOFFSET,CANGS)
151
152      C    AND GRADE THE FACE
153      C
154      CALL GRADE (INPS)
155      C    SAVE GRADE OF FIRST SIDE OF BOARD
156      C
157      IF (1.EQ.1) OLD = INPS
158      C    SUBTRACT BOARD THICKNESS AND PROCESS THE INNER FACE
159      C
160      DISTRD = DISTRD-T
161      C
162      10    CONTINUE
163      C
164      DETERMINE BOARD GRADE AND VALUE BASED ON GRADES OF BOTH FACES
165      C
166      CALL PRICED (OLD,INPS,TVAL,PERC)
167      C
168      CUT ANOTHER BOARD
169      C
170      GO TO 6
171      C
172      TURN LOG 90 DEGREES TO CUT NEXT FACE
173      C
174      DO 12 J=1,MAX
175      C
176      ANGLE(J) = AHD(ANGLE(J)+90.0,360.0)
177      12    CONTINUE
178      CANGS = AHD(CANGS+90.0,360.0)
179      NBD = 0
180      13    CONTINUE
181
182      C    PRINT RESULTS FOR THIS POSITION
183      C
184      IROTRAT = IMX*15-15
185      LWRITE (6,114) IROTRAT
186      LWRITE (6,113) (PERC(J),J=1,5)
187      TOTS = PERC(1)+PERC(2)+PERC(3)+PERC(4)+PERC(5)
188
189      LWRITE (6,112) TOTS
190      LWRITE (8,283) IROTRAT
191      LWRITE (8,284) (PERC(J),J=1,5)
192      LWRITE (8,284) TOTS
193      DO 14 J=1,5
194      PERC(J) = (PERC(J)/VOL)*100.
195      14    CONTINUE
196      TOTS = (TOTS*VOL)*100.
197      LWRITE (6,111) TOTS
198      IF (TVAL.GT.HIGHS) HIGHS = TVAL
199      LWRITE (6,118) (ANGLE(J),J=1,MAX)
200      LWRITE (6,120) (PERC(J),J=1,5),TVAL
201      LWRITE (8,205) (PERC(J),J=1,5)
202      LWRITE (8,284) TVAL,TOTS
203      C    SAVE HIGH, LOW, AND AVERAGE YIELDS FOR THIS LOG
204      AVGS = AVG*TVAL
205      IF (TVAL.LT.SMALLS) SMALLS = TVAL
206      C    POTE LOG BY 15 DEGREES AND REPROCESS
207
208      DO 15 J=1,MAX
209      ANGLE(J) = AHD(ANGLE(J)+15.0,360.0)
210
211      15    CONTINUE
212      CANGS = AHD(CANGS+15.0,360.0)
213
214      16    CONTINUE
215
216      C    PRINT HIGH, LOW, AND AVERAGE YIELDS FOR THIS LOG
217
218      AVGS = AVG*12.
219      LWRITE (6,183) HIGHS,AVGS,SMALLS
220
221
222      C

```

NOTE# STATEMENT NUMBERS

223 C LOG COMPLETELY PROCESSED, READ PARAMETERS FOR NEXT LOG  
224 C GO TO 1  
225 C  
226 C 17 STOP  
228 C FORMATS FOR DEVICE 6 (PRINTER)  
229 C  
230 C 180 FORMAT (F5.1)  
231 181 FORMAT (F5.3)  
232 182 FORMAT (' CORE DEFECT RADIUS ',F7.2,' LINEAR OFFSET ',F7.2,  
233 ' ANGULAR OFFSET ',F7.2)  
234 183 FORMAT (' \$ YIELD: HIGH ',F6.2,' AVERAGE ',F6.2,' LOW ',F6.2)  
235 184 FORMAT (' QUADRANT SAWING METHOD ',AB)  
236 185 FORMAT (' LTR. PARAME: LENGTH ',F6.1,' DIA ',F6.1,' TAPER ',  
237 ' FFC ',I3,' DEGREE, LOG VOLUME ',F6.2,' CUBIC FEET')  
238 186 FORMAT (' ANGLES MEASURED FROM ZERO DEGREES \* EAST FOR A ',  
239 ' VERTICALLY CUTTING SHA, KNOT TAPER ',F6.2)  
240 187 FORMAT (' CUTTING PARAMETERS: KERF ',F6.4,' BOARD THICKNESS ',  
241 ' F5.2)  
242 188 FORMAT (' PRICES PER BOARD FOOT: ',SF18.4)  
243 189 FORMAT (SF18.4)  
244 190 FORMAT (' KIND ANGLES: ',SF7.2,'(./,13X,1SF7.2))  
245 191 FORMAT (' TOTAL PERCENT YIELD IS ',F5.2)  
246 192 FORMAT (' TOTAL SURFACE MEASURE IS ',F6.2)  
247 193 FORMAT (' SURFACE MEASURE PER GRADE IS ',F6.2)  
248 194 FORMAT (2(./),' ROTATION IS ',SF7.2)  
249 195 FORMAT (SF7.2)  
250 196 FORMAT (SF6.3)  
251 197 FORMAT (SF6.2)  
252 198 FORMAT (SF6.2)  
253 199 FORMAT (12)  
254 200 FORMAT (' THERE ARE ',I3,' KNOTS IN THIS LOG. THEY ARE '  
255 ' IX. ANGLE HEIGHT LENGTH')  
256 201 FORMAT (' X AVAILABLE SURFACE MEASURE IN EACH GRADE ',SF5.1,  
257 ' LOG VALUE IS \$ ',F6.2)

258 C FORMATS FOR DEVICE 8 (STORAGE MEDIUM)  
259 C  
260 260 FORMAT (F6.1,F5.1,I3,3F4.1,F6.2,F5.3,F6.3)  
261 261 FORMAT (13,1SF7.2,'),  
262 262 FORMAT (SF7.4)  
263 263 FORMAT (13)  
264 264 FORMAT (13)  
265 265 FORMAT (SF8.2)  
266 266 FORMAT (10F6.2)  
267 267 END

\* Indicates line at which symbol or number is defined

12	100	21	22	24	26	29	*231
	101	23	*232				
	102	72	*233				
	103	221	*235				
	104	65	*236				
	105	66	*237				
	106	71	*239				
	107	73	*241				
	108	74	*243				
	109	49	*244				
	110	199	*245				
	111	197	*246				
	112	199	*247				
	113	187	*248				
	114	196	*249				
	115	69	*250				
	116	34	*251				
	117	51	*252				
	118	41	*253				
	119	67	*254				
	120	199	*255				
	200	73	*261				
	201	76	*262				
	202	79	*263				
	203	198	*264				
	204	191	192	201	*265		
	205	200	*266				

## YEAR VARIABLES

PROD							
ANGLE	6	451	63	78	146	*177	198
AVGCS	466	*265	*220	221			*212
BOLIN	11						
BP	11	*36					
BODPBM	11						
CANCS	*36	72	75	151	*179	*214	
COPDIP	*33	46					
COPED	151						
CHARUS	446	72	75	151			
D	11	*21	59	66	75		
DATE	8	45	65				
DISTBD	11	*163					
F	In	*105	110	114	119		
FBS	11	*465	74	79			
FLX	8	11	*129				
FLY	8	11	4131				
FUX	8	11	*130				
FUY	8	11	*132				
GHOST	155						
HEIGHT	8	*51	69	77	146		
HIGHS	1	*123	159				
In	10	11	*135				
IKK	*92	165	198				
IMOTAT	10185	186	198				
J	*50	453	464	468	69	76	146
K	11	*22	69	76	146	*176	177
KNOTS	11	69	61	66	75		
KNOTN	146						
KTLEN	8	*54	69	76	146		
L	11	*22	69	66	75		
LOGS	11	*29	60	66	75		
LIVOL	451	66	75	194	196		
MAX	441	50	53	67	75		
	211						

## Subroutine KERF

```

1 C SUBROUTINE *KERF* CUTS A BOARD FROM THE LOG. THE BOARD WIDTH IS
2 C DEFINED AT THE CENTER, AND THE BOARD LENGTH FROM THE BOTTOM
3 C OF THE LOG
4 C
5 C SUBROUTINE KERF (*,NBD,F)
6 C
7 C IMPLICIT REAL (K,L)
8 C REAL FLX(26,2),FLY(26,2),FUX(26,2),FUY(26,2)
9 C INTEGER ID(22,2),NODEFC(2),F
10 C COMMON /DEFER/ NODEFC,FLX,FLY,FUX,FUY
11 C           /BOARD/ L,D,K,T,LOGS,RB,DISTBD,BORDDU,KNOTS,R,BF
12 C
13 C ALL BOARDS MUST BE AT LEAST .48 INCHES LONG, 2.5 INCHES WIDE AT
14 C THE TOP, AND 3 INCHES WIDE AT THE CENTER. IN ADDITION, THE TOTAL
15 C AMOUNT OF WINE FROM THE CENTER TO THE TOP IS LIMITED TO 4 INCHES.
16 C THE BOARD IS CUT BACK IN ONE FOOT INCREMENTS IF THE WIDTH OF WINE
17 C TESTS FAIL. IF THE LENGTH TEST FAILS, ANOTHER BOARD IS CUT FROM
18 C THIS FACE
19 C
20 C FIND OUTER FACE OF BOARD
21 C
22 C
23 C
24 C   RI = D/2.
25 C   1 R = RT-K
26 C   NBD = NBD+1
27 C   BOLLEN = L
28 C
29 C IF THIS FACE OF THE LOG IS COMPLETELY CUT, EXIT
30 C
31 C IF (RI.LT.0.) RETURN
32 C DISTBD = RB-R
33 C LNB = DISTBD-T-K
34 C IF (F.EQ.1.0R.F.EQ.3.) LNB = DISTBD
35 C
36 C THE CENTRAL CANT HAS BEEN DEFINED AS FOUR BOARDS SQUARE
37 C
38 C IF (NBD.LT.3.AND.(F.EQ.2.OR.F.EQ.4.)) LNB = 2.*RT+1.5*M
39 C POSLEN = R*TAN(RAD(.98-.LOGS))
40 C
41 C CHECK FOR BOARDS SHORTER THAN THE LOG DUE TO LOG TAPER
42 C
43 C IF (POSLEN.LT.BPLEN) EDLEN = POSLEN
44 C BOARDS MUST BE AT LEAST 4 FEET LONG
45 C
46 C 2 IF (BDLEN.LT.-48.) GO TO 1
47 C
48 C DETERMINE HALF-WIDTH AT MID-LENGTH OF BOARD
49 C
50 C RI = RT+L-BDLEN/2.*TAN(RAD(.LOGS))
51 C IF (DISTBD.GT.RI) GO TO 3
52 C BORDDU = SORT((R*TAN(.2*DISTBD)*2))
53 C IF (LNB.LT.BORDDU) BORDDU = LNB
54 C
55 C BOARD MUST BE AT LEAST 3 INCHES WIDE AT MID-LENGTH
56 C
57 C IF (BORDDU.LT.1.5) GO TO 3
58 C
59 C DETERMINE HALF-WIDTH AT TOP OF BOARD
60 C
61 C ST = RT+L-BDLEN*TAN(RAD(.LOGS))
62 C IF (DISTBD.GT.RT) GO TO 3
63 C UT = SORT((R*TAN(.2*DISTBD)*2))
64 C IF (LNB.LT.UT) UT = LNB
65 C
66 C BOARD MUST BE AT LEAST 2.5 INCHES WIDE AT THE TOP
67 C
68 C IF (LNB.LT.1.25) GO TO 3
69 C
70 C LIMIT THE TOTAL WINE TO AVOID GENERATING MORE DEFECTS THAN THE
71 C GRADING PROGRAM CAN HANDLE. TOTAL WIDTH OF WINE MUST BE LESS
72 C THAN 4 INCHES
73 C
74 C LBN = BOARD-LBT

```

IF (LBN.GT.2.) GO TO 3

```

75 C
76 C CONVERT BOARD WIDTH AND LENGTH TO EVEN QUARTER INCHES
77 C
78 C BOARDU = INT(BORDDU*.1/8.
79 C BOLLEN = INT(BDLEN*.1/8.
80 C
81 C DEFINE BOARD EDGES IN QUARTER INCH UNITS
82 C
83 C KERFO
84 C   KERFO
85 C     FLX(26,1) = BDLEN*.1/8.
86 C     FLY(26,1) = BOARDU*.1/8.
87 C     FUX(26,1) = 0.0
88 C     FUY(26,1) = 0.0
89 C
90 C CALCULATE UNROUNDED BOARD FEET
91 C
92 C KERFO
93 C   KERFO
94 C     DF = RI*EN*BODPHI*.2.*T/144.
95 C     PLT,FLFH
96 C     CUT BACY BOARD BY ONE FOOT
97 C       3 BDLEN = BDLEN-12.
98 C       GO TO 2
99 C
END

```

```

*** STATEMENT NUMBERS ***
1    25      47
2    47      98
3    52      63
4    63      76
5    97
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

```

*** VARIABLE NAMES ***
BDLEN
BF
BOARDU
D
DISTBD
F
FLX
FUX
FUY
ID
KNOTS
L
LOGS
NBD
NODEFC
POGLEN
R
RAD
RB
RT
TAN
UBH
UT

```

## Subroutine WANE

```

1 C SUBROUTINE 'WANE' LOCATES WANE DEFECTS ON THE BOARD
2 C
3 C SUBROUTINE LINED
4 C
5 C THE BOARD EDGE HAS A NEW WANE DEFECT IF THE WIDTH DROPS BY 1/2
6 C INCH (1/4 INCH FOR EACH SIDE) FROM THAT AT THE MIDDLE. THE BOARD
7 C IS SYMMETRICAL SO THE WANE DEFECTS OCCUR AT THE SAME PLACE ON
8 C BOTH EDGES OF THE BOARD
9 C
10 C
11 C IMPLICIT REAL (K,L)
12 C INTEGER ID(22,2),NODEFC(2)
13 C REAL FLX(26,2),FLY(26,2),FLX(26,2),FLY(26,2)
14 C COMMON /BOARD/ L,D,K,T,LOGS,R0,DISTD,BOLEN,BORDU,KNOTS,R,BF
15 C /DEFEC/ NODEFC,FLX,FLY,FUY,FLY,FLX
16 C
17 C CON1 = TH1(RAD(LOGS))
18 C CON2 = RAD(DISTD)*2
19 C IF ((RB-SORT((CON2-BOLEN)/2))>CON1.GE.BOLEN) RETURN
20 C OLDH = BOLEN/2.
21 C V = 0.
22 C
23 C FIND THE PLACE ALONG THE BOARD WHERE THE HALF-WIDTH HAS DECREASED
24 C BY 1/4 INCH. THE DEFECT ENDS HERE AND EXTENDS FROM THE END OF THE
25 C LAST WANE DEFECT (OR FROM THE MIDDLE OF THE BOARD)
26 C
27 C
28 C V = V+0.25
29 C SDR = SORT((CON2-(BOARDU-V)*2)
30 C H = INT(VANTIN((RB-SDR)*4./CON1.BOLEN*4.))
31 C NODEFC(1) = NODEFC(1)+1
32 C M = NODEFC(1)
33 C LIMIT NUMBER OF DEFECTS TO MAXIMUM ALLOWED BY GRADING PROGRAM (22)
34 C
35 C IF (M.GT.22) GO TO 2
36 C
37 C SAVE WANE COORDINATES FOR GRADING PROGRAM
38 C
39 C FUY(M,1) = V*4.+1.
40 C FLY(M,1) = B.
41 C FUX(M,1) = H
42 C FLX(M,1) = OLDH
43 C ID(M,1) = S
44 C
45 C SINCE THE BOARD IS SYMMETRICAL USE THE PREVIOUSLY GENERATED HEIGHT
46 C FOR THE WANE DEFECTS ON THE OTHER EDGE OF THE BOARD
47 C
48 C 2 NODEFC(1) = NODEFC(1)+1
49 C
50 C M = NODEFC(1)
51 C IF ('M.GT.22 GO TO 3
52 C FLY(M,1) = BOARDU-V*4.-1.
53 C FUX(M,1) = BOARDU-H
54 C FLX(M,1) = OLDH
55 C ID(M,1) = S
56 C
57 C WHEN THE DEFECT EXTENDS TO THE END OF THE BOARD, EXIT
58 C
59 C 3 IF (M.GE.BOLEN*4.) RETURN
60 C
61 C THE NEXT WANE DEFECT BEGINS WHERE THIS ONE ENDS
62 C
63 C OLDH = H
64 C GO TO 1
65 C RETURN
66 C
67 C

```

### Subroutine KNOT

```

1 C SUBROUTINE 'KNOT' LOCATES THE KNOT DEFECTS ON THE BOARD FACE.
2 C ANGLES ARE MEASURED CLOCKWISE WITH 0 DEGREES DEFINED AS THE
3 C LINE FROM THE CENTER OF THE LOG PERPENDICULAR TO THE BOARD FACE
4 C
5 C      5 C
6 C      SUBROUTINE KNOT (XHLEN,XLEN,XTANG$)
7 C
8 C      IMPLICIT REAL (X,L)
9 C      INTEGER ID(22,2),NODFEC(2)
10 C      REAL FL(26,2),FLY(26,2),FLX(26,2),FY(26,2)
11 C      COMMON /NODFEC/ NODFEC,FLX,FY,FUX,FUY,ID,
12 C                      .,FORBD,LD,K,T,LOGS,RB,DISTBD,BOLEN,BOARDU,KNOT$,R,BF
13 C
14 C      SEE IF THE KNOT IS LONG ENOUGH TO REACH THE BOARD FACE
15 C
16 C
17 C      IF (XLEN.LT.DISTBD) RETURN
18 C
19 C      CALCULATE LOG RADIUS AT KNOT HEIGHT, BOARD HALF-WIDTH ANGLE, AND
20 C      KNOT HEIGHT HALF-LENGTH AS PROJECTED ON THE FACE
21 C
22 C      RADIUS = RB*YHIGH*TAN(RAD(LOG$))
23 C      BDUDIS = DECAYDIS(BOARDUD,DISTBD),
24 C      DY = DISTBD*TAN(RAD(KNOT$)),
25 C      THE KNOT EFFECTIVELY ENDS AT THE OUTER EDGE OF THE LOG
26 C
27 C      IF (.LE.,LT,1)PNUIS,LEN = LPADUS
28 C
29 C      SINCE THE LOG IS TAPEDED, THE OUTER BOARDS MAY NOT BE THE FULL LOG
30 C      LENGTH, SO KNOTS ABOVE THE END OF THE BOARD ARE NOT CONSIDERED
31 C
32 C      IF (LEN.LT.,LT,DISTBD) RETURN
33 C
34 C      ANGLE OF ROTATION PUTS KNOT COMPLETELY OUTSIDE BOARD
35 C
36 C      IF ((XTANG$.GT.,LT,.00.) .AND. ((KTANG$-XKNOT$).LT.,(360.-BDUDIS))) RETURN
37 C      IF ((YTANG$.LE.,LT,.00.) .AND. ((KTANG$-YKNOT$).LT.,(360.-BDUDIS))) RETURN
38 C
39 C      FIND THE QUADRANT IN WHICH THE KNOT LIES (ASSUME QUADRANT ONE)
40 C
41 C      QUADRANT TWO
42 C
43 C      QUADRANT THREE
44 C      IF ((YTANG$.LT.,LT,.00.), .AND. ((KTANG$-XKNOT$).GT.,BDUDIS)) GO TO 3
45 C
46 C      QUADRANT THREE
47 C
48 C      IF ((YTANG$.GE.,LT,.00.), .AND. ((KTANG$-XKNOT$).LT.,(360.-BDUDIS)))
49 C          - 100 1
50 C
51 C      QUADRANT FOUR
52 C
53 C      IF ((YTANG$.GT.,LT,.00..)) GO TO 2
54 C
55 C      CALCULATE DISTANCE TO FACE ALONG BOTH SIDES OF KNOT
56 C
57 C      QUADRANT ONE
58 C      ANGBD1 = DISTBD*COS(RAD(KTANG$-XKNOT$))
59 C      ANGBD2 = DISTBD*COS(RAD(KTANG$-YKNOT$))
60 C
61 C      SEE IF KNOT CENTER IS WITHIN ONE KNOT HALF-ANGLE OF 0 DEGREES
62 C
63 C      IF ((YTANG$-XKNOT$.LT.,LT,.00..)) GO TO 2
64 C
65 C      DISTBD*COS(RAD(KTANG$-XKNOT$))
66 C      DISTBD*COS(RAD(KTANG$-YKNOT$))
67 C
68 C      KNOT IS NOT LONG ENOUGH TO REACH FACE
69 C
70 C      IF (.LE.,LT,1)PNUIS,LEN = LPADUS
71 C
72 C      FIND INTERSECTION OF NEAR SIDE OF KNOT AND FACE
73 C
74 C      XP = BDUDIS*DISTBD*TAN(RAD(KTANG$-XKNOT$))
75 C      IF (.LE.,LT,1)PNUIS,LEN = LPADUS
    
```

76 C FAR SIDE OF KNOT REACHES FACE

```

77 C      XL = BOARDUD*DISTBD*TAN(RAD(KTANG$-XKNOT$))
78 C      FULWID = (.X-.XL)/2.
79 C      GO TO 8
80 C
81 C      FAR SIDE OF KNOT DOES NOT REACH FACE
82 C
83 C      1 XL = BOARDUD*SORT(LLEN**2-DISTBD**2)
84 C      FULWID =
85 C      .DISTBD*TAN(PAD((KTANG$+XKNOT$))-DISTBD*TAN(RAD(KTANG$-XKNOT$)))
86 C
87 C      KNOT IS WITHIN ONE KNOT HALF-ANGLE OF 0 DEGREES
88 C
89 C      2 XL = AMIN1(SORT(KLEN**2-DISTBD**2),DISTBD*TAN(RAD(KTANG$-XKNOT$)))
90 C      KNOT
91 C      2 XL = BOARDUL-XL
92 C      KNOT
93 C      2 XE = AMIN1(SORT(KLEN**2-DISTBD**2),DISTBD*TAN(RAD(KTANG$-XKNOT$)))
94 C      KNOT
95 C
96 C      KNOTS NEARLY PERPENDICULAR HAVE THEIR WIDTH ESTIMATED BY THEIR
97 C      LENGTH
98 C
99 C      3 ANGEL1 = F1STEN*COS(RAD(KTANG$-YKNOT$))
100 C      TON SHIFT TO REACH SAME
101 C
102 C      QUADRANT TWO
103 C
104 C      IF ((LEN.LT,1)ANGBD1) RETURN
105 C      XP = BOARDUD*SORT(ANGBD1)**2-DISTBD**2
106 C      XL = BOARDUD*SORT(LEN**2-DISTBD**2)
107 C
108 C      IF ((YTANG$-XKNOT$.GE.,89.5))
109 C          .FULWID = LPADUS-DISTBD*TAN(RAD(KTANG$-XKNOT$))
110 C
111 C      IF ((YTANG$-XKNOT$.LT.,89.5))
112 C          .FULWID = AMIN1(1.PADUS,DISTBD*TAN(RAD(KTANG$-XKNOT$)))
113 C
114 C      FULWID = MIN1(FULWID,XP)
115 C      -DISTBD*TAN(RAD(KTANG$-XKNOT$))
116 C
117 C      QUADRANT THREE
118 C
119 C      IF ((YTANG$.LT,1)ANGBD1) RETURN
120 C      XP = BOARDUD*SORT(LEN**2-DISTBD**2)
121 C      XL = BOARDUD*SORT(ANGBD1)**2-DISTBD**2
122 C
123 C      IF ((YTANG$-XKNOT$.LE.,20.5))
124 C          .FULWID = LPADUS-DISTBD*TAN(RAD(360.-KTANG$-XKNOT$))
125 C
126 C      IF ((YTANG$-XKNOT$.LT.,20.5))
127 C          .FULWID = MIN1(FULWID,DISTBD*TAN(RAD(360.-KTANG$-XKNOT$)))
128 C
129 C      QUADRANT FOUR
130 C
131 C
132 C      5 IF ((360.-KTANG$-XKNOT$.LT.,0..)) GO TO 7
133 C      ANGBD1 = DISTBD*COS(RAD(360.-KTANG$-XKNOT$))
134 C      ANGBD2 = DISTBD*COS(RAD(360.-KTANG$-YKNOT$))
135 C
136 C      IF ((XLEN.LT,1)ANGBD1) RETURN
137 C      XE = BOARDUD*DISTBD*TAN(RAD(360.-KTANG$-XKNOT$))
138 C
139 C      IF ((XLEN.LT,1)ANGBD2) GO TO 6
    
```

```

140 C      FAR SIDE OF KNOT DOES NOT REACH FACE
141 C
142 C      6 XP = BOARDUD*DISTBD*TAN(RAD(360.-KTANG$-YKNOT$))
143 C      FULWID = (.X-.XL)/2.
144 C
145 C      FAR SIDE OF KNOT DOES NOT REACH FACE
146 C
147 C
148 C      6 XP = BOARDUD*SORT(LLEN**2-DISTBD**2)
149 C      FULWID = DISTBD*TAN(RAD(360.-KTANG$-XKNOT$))
150 C
151 C
152 C
153 C
    
```

16 KNOT IS WITHIN ONE KNOT HALF-ANGLE OF 0 DEGREES

154 C  $XQ = \text{MIN}(\text{SQR}(KLEN^2 - D1^2 * D2^2))$

155 C  $D = \text{DIST}(XKNOT(XKNOT-XQ), YKNOT(YKNOT-YQ))$

156 C  $XQ = \text{BOARD-XQ}$

157 C  $YQ = \text{MIN}(\text{SQR}(KLEN^2 - DIST(XQ, YQ)^2))$

158 C  $XQ = \text{DIST}(XKNOT(XKNOT-XQ), YKNOT(YKNOT-YQ))$

159 C  $YQ = \text{MIN}(\text{SQR}(KLEN^2 - DIST(XQ, YQ)^2))$

160 C  $D = \text{DIST}(XKNOT(XKNOT-XQ), YKNOT(YKNOT-YQ))$

161 C  $XQ = \text{BOARD-XQ}$

162 C  $YQ = \text{MIN}(\text{SQR}(KLEN^2 - DIST(XQ, YQ)^2))$

163 C  $D = \text{DIST}(XKNOT(XKNOT-XQ), YKNOT(YKNOT-YQ))$

164 C IF A KNOT COMPLETELY INTERSECTS THE FACE, ITS LENGTH IS EXACTLY AS

165 C MEASURED ON THE FACE. OTHERWISE, ITS LENGTH IS ESTIMATED AS A

166 C PROPORTION OF ITS PROJECTED LENGTH

167 C

168 C  $8 = \text{MIN}(DY, \text{DY}((CAR-XQ) * Q2, FULLID))$

169 C  $YB = XHIGH-DY$

170 C  $YT = XHIGH+DY$

171 C DEFECT LOCATIONS FOUND. CHECK THAT THEY DO NOT EXTEND BEYOND THE

172 C BOARD EDGES AND CONVERT TO QUARTER INCH UNITS FOR THE GRADING

173 C PROGRAM

174 C

175 C  $XQ = \text{APRDX}(G, \text{INT}(XQ/4.0), \text{INT}(YQ/4.0), +1)$

176 C  $XP = \text{APRDX}(G, \text{INT}(BOARDX4.0), \text{INT}(YQ/4.0), +1)$

177 C  $YP = \text{APRDX}(G, \text{INT}(YQ/4.0), \text{INT}(YQ/4.0), +1)$

178 C  $YT = \text{APRDX}(G, \text{INT}(BOARDY4.0), \text{INT}(YQ/4.0), +1)$

179 C LIMIT NUMBER OF DEFECTS TO MAXIMUM ALLOWED BY GRADING PROGRAM (12)

180 C  $182 C \quad \text{NDEF}(C1) = \text{NDEF}(C1)+1$

181 C  $\text{IF } (\text{NDEF}(C1) > G1) \text{ RETURN}$

182 C SAVE KNOT COORDINATES FOR GRADING PROGRAM

183 C

184 C  $185 C \quad \text{ID}(\text{NDEF}(C1), 1) = 3$

185 C  $\text{FLY}(\text{NDEF}(C1), 1) = XQ$

186 C  $\text{FLX}(\text{NDEF}(C1), 1) = YB$

187 C  $\text{FLY}(\text{NDEF}(C1), 1) = YQ$

188 C  $\text{FLX}(\text{NDEF}(C1), 1) = YT$

189 C  $\text{RETURN}$

190 C

191 C

192 C

193 C

194 C

195 C STATEMENT NUMBERS  
1 75 104  
2 61 91  
3 45 184  
4 49 129  
5 54 133  
6 138 146  
7 133 156  
8 69 87  
9 100 100  
10 11 23

196 C VARIABLES  
1 76 178  
2 179 113  
3 91 184  
4 78 130  
5 54 133  
6 138 146  
7 133 156  
8 69 87  
9 100 100  
10 11 23

STATEMENT	NUMBER	NAME	NUMBER	NAME
1	75	104		
2	61	91		
3	45	184		
4	49	129		
5	54	133		
6	138	146		
7	133	156		
8	69	87		
9	100	100		
10	11	23		

## Subroutine CORE

```

1 C SUBROUTINE 'CORE' LOCATES THE CORE DEFECT ON THE BOARD FACE
2 C
3 C SUBROUTINE CORE0 (CRADIUS,COFFSET,CANG$)
4 C
5 C IMPLICIT REAL (K,L)
6 C REAL FL((26,2),FLX(26,2),FLY(26,2),FLY(26,2))
7 C INTEGER NODEFC(2,10)(2,2)
8 C COMMON /BOARD/ L,D,K,T,LOGS,R,BDISTD,BOLEN,BOARDJ,KNOT$,R,BF
9 C          /DEFEC/ NODEFC,FLX,FLY,FLX,FLY,1D
10 C
11 C
12 C FIND DISTANCE TO THE BOARD FACE FROM THE CENTER OF THE CORE. EXIT
13 C IF THE BOARD FACE IS BEYOND THE CORE RADIUS
14 C
15 C TDIST = DISTB-COREFACE(RADS(CANG$))
16 C IF (CRADIUS.LE.TDIST) RETURN
17 C      NODEFC(1) = NODEFC(1)+1
18 C
19 C LIMIT NUMBER OF DEFECTS TO MAXIMUM ALLOWED BY GRADING PROGRAM (22)
20 C
21 C
22 C IF (NODEFC(1).GT.22) RETURN
23 C
24 C LOCATE THE CENTER POINT OF THE INTERSECTION OF THE BOARD FACE AND
25 C THE CORE DEFECT. CALCULATE THE DISTANCE ON THE BOARD FROM THE
26 C CENTER POINT TO THE EDGES OF THE CORE DEFECT. THE CORE DEFECT
27 C IS NOT TAPERED. SO IT WILL EXTEND THE FULL LENGTH OF THE BOARD
28 C
29 C CONST = SQRT(CRADIUS**2-TDIST**2)
30 C TBCTR = COSE*WSIN(RADS(CANG$))
31 C X0 = R0PDA*CONST-TBCTR
32 C X1 = BOARDJ-CONST-TBCTR
33 C
34 C SAVE CORE DEFECT COORDINATES FOR GRADING PROGRAM
35 C
36 C FIX(NODEFC(1),1) = INT(BOLEN*4.)
37 C FLX(NODEFC(1)+1)= 0.0
38 C FLY(NODEFC(1)+1)= 0.0
39 C FLX(NODEFC(1)+1)= MIN(WSIN(BOARDJ*PI/180),INT(XP*4.+1.))
40 C FLY(NODEFC(1)+1)= MIN(WSIN(BOARDJ*PI/180),INT(XL*4.+1.))
41 C PT(THREE)
42 C END

```

## WORK VARIABLES

	NAME	VARIABLES	TYPE
1	ANGL	39	
2	ARINC	38	
3	BOLEN	9	
4	BF	9	
5	BOARDJ	9	
6	CANG\$	4	
7	COFSET	4	
8	CONS	*29	
9	CORE0	4	
10	COS	16	
11	CPRADIUS	4	
12	D	9	
13	DISTBD	9	
14	FLX	9	
15	FLY	9	
16	FUX	9	
17	FUY	9	
18	ID	8	
19	INT	3F	
20	K	9	

## Subroutine GRADE

```

1 C SUBROUTINE 'GRADE' CALLS THE U.S. FOREST PRODUCTS LABORATORY
2 C GRADING PROGRAM
3 C
4 C SUBROUTINE GRADE (NPG)
5 C
6 C      INTEGER I1,I2,I3,I4,IMDEF(2),IX(22),
7 C           REAL FL(126,2),FLY(26,2),FUX(26,2),
8 C           COMMON /IMDEF/ IMDEF,FLY,FUX,FLY,FL,I1
9 C
10 C      N = IMDEF(1)
11 C      IF (N.GT.22) GO TO 3
12 C      IF (I.M.EQ.0) GO TO 2
13 C      DO 1 J=1,I1
14 C         IX(J) = ID(J,1)
15 C      1 CONTINUE
16 C      2 NPG = 1
17 C      CALL GDP (N,IX,NPG,SP)
18 C      RETURN
19 C
20 C      3 NPG = 5
21 C      IF NUMBER OF DEFECTS IS MORE THAN GRADING PROGRAM CAN HANDLE.
22 C      THE R.H.S. IS 50.
23 C
24 C      4 UP TO 16,160!
25 C      5 DEFECT LIMIT EXCEEDED.
26 C      6 RETURN
27 C      END
28 C
29 C
30 C
31 C
32 C
33 C
34 C
35 C
36 C
37 C
38 C
39 C
40 C
41 C
42 C
43 C
44 C
45 C
46 C
47 C
48 C
49 C
50 C
51 C
52 C
53 C
54 C
55 C
56 C
57 C
58 C
59 C
60 C
61 C
62 C
63 C
64 C
65 C
66 C
67 C
68 C
69 C
70 C
71 C
72 C
73 C
74 C
75 C
76 C
77 C
78 C
79 C
80 C
81 C
82 C
83 C
84 C
85 C
86 C
87 C
88 C
89 C
90 C
91 C
92 C
93 C
94 C
95 C
96 C
97 C
98 C
99 C
100 C

```

## Subroutine PRICE

```

1 C SUBROUTINE 'PRICE' DETERMINES THE BOARD GRADE BASED ON THE GRADES
2 C OF BOTH SIDES (OLD,NPG) AND CALCULATES THE BOARD VALUE
3 C
4 C
5 C
6 C      SUBROUTINE PRICE (OLD,NPG,VAL,PERC)
7 C
8 C      IMPLICIT REAL (K,L)
9 C      INTEGER OLD
10 C      REAL PERC(.5)
11 C      COMMON /OPCF/ FAS,SEL,GNEC,TUDC,THRB,
12 C                   BOARDP, L,D,K,T,LOGS,PB,DISTBD,BOLIN,BOARDU,KNOTS,R,BF
13 C
14 C      DETERMINE THE BOARD GRADE
15 C
16 C      IF (OLD.FG.EQ.NPG.EQ.5) GO TO 3
17 C      IF (OLD.FG.2.OF.NPG.EQ.2) GO TO 1
18 C      IF (OLD.EQ.NPG .50 TO 4
19 C          MPC = M4*D'GLT,H+G1
20 C          IF (NPG.EQ.4) GO TO 8
21 C          MPC = 2
22 C          IF (NPG .1 GO TO 1
23 C          1 IF (LUTL.EQ.4.UF.NPG.EQ.4) GO TO 2
24 C          MPC = 3
25 C          IF (NPG .3 GO TO 4
26 C          MPC = 4
27 C          GO TO 3
28 C          2 MPC = 5
29 C          GO TO 4
30 C          3 MPC = 6
31 C          AND CALCULATE THE BOARD VALUE
32 C
33 C          5 V = BF*FAS
34 C          GO TO 10
35 C          6 V = BF*SEL
36 C          GO TO 16
37 C          7 V = BF*TUDC
38 C          GO TO 18
39 C          8 V = BF*TUDC
40 C          GO TO 10
41 C          9 V = BF*THRB
42 C
43 C      INFERFACE SURFACE MEASURE IN THIS GRADE AND THE TOTAL LOG VALUE
44 C
45 C      10 PERC (NPG) = BF*FEC(NPG)
46 C      11 VAL = TVAL + V
47 C      RETURN
48 C
49 C
50 C
51 C
52 C
53 C
54 C
55 C
56 C
57 C
58 C
59 C
60 C
61 C
62 C
63 C
64 C
65 C
66 C
67 C
68 C
69 C
70 C
71 C
72 C
73 C
74 C
75 C
76 C
77 C
78 C
79 C
80 C
81 C
82 C
83 C
84 C
85 C
86 C
87 C
88 C
89 C
90 C
91 C
92 C
93 C
94 C
95 C
96 C
97 C
98 C
99 C
100 C

```

\*\*\* STATEMENT NUMBERS \*\*\*

1	15	*17
2	14	*18
3	13	*25
100	26	*27

\*\*\* VARIABLES \*\*\*

FLX	8	9
FLY	8	9
FUX	8	9
FUY	8	9
GRADE	5	
GDP	12	
I1	7	9
IX	7	*16
J	*15	16
M	*12	13
IMDEF	7	9
NPG	5	*18
SH	19	*25

\*\*\* STATEMENT NUMBERS \*\*\*

1	17	*22
2	23	*26
3	15	*28
4	12	22
5	5	*33
6	23	*35
7	7	*37
8	20	*39
9	22	*41
10	7	*45

	VARIABLES	VALUES
BULEN	10	33
BF	10	35
BRDPW	10	37
D	10	39
DISBD	10	41
FAS	10	45
K	10	33
KNOTS	10	35
L	10	37
LNS	10	39
MAX	10	45
NPC	5	16
OLD	45	17
ONE	5	18
ONEC	10	17
PEP	5	19
PRICED	5	23
R	10	25
RB	10	27
SEL	10	35
T	10	37
THRE	10	41
TVAL	5	45
TUDR	10	39
V	*33	*35

### Function DEG

```

1 C FUNCTION 'DEG' CONVERTS ITS ARGUMENT (IN RADIANS) TO DEGREES
2 C
3 C
4 C FUNCTION DEG (A)
5 C
6 C
7 C DEG = A*188.0/3.141592
8 C RETURN
9 C END

```

\*\*\* VARIABLES \*\*\*

A	DEG	4	7
*33	*35	*37	*39
		*41	*45

## 20 Function RAD

```
1 C FUNCTION 'RAD' Converts its argument (in degrees) to radians
2 C
3 C
4 C FUNCTION RAD (A)
5 C
6 C
7 C RAD = A*3.141592/180.0
8 C RETURN
9 C END
```

now VARIABLES now

```
A      4      7
RAD    4      7
```

## Cant Sawing (CANT)

```

1 C CANT CUTS FROM FACES 1 AND 3 LEAVING A CANT OF SPECIFIED
2 C THICKNESS. THIS CANT IS THEN TURNED 90 DEGREES AND CUT
3 C INTO BOARDS
4 C
5 C IMPLICIT REAL (K,L)
6 C REAL FLX(26,2),FLY(26,2),FUX(26,2),FLY(26,2),PERC(5),ANGLE(5B),
7 C ME(5B),KTLEN(5B),DAT(8)
8 C INTEGER ID(22,2),NODFC(2),BLD(2,2),MAX(1,1,MAX)
9 C COMMON /DEFECT/ NODFC(2),BLD(2,2),FLX,FLY,FUX,FUY,TD
10 C /BOARD/ L,D,K,T,LOGS,RB,DISTB,DILEN,BOARDL,KNOTS,CANT,R,BF
11 C /PRICF/ FAS,SEL,ONEC,TUDC,THR
12 C
13 C THIS PROGRAM ASSUMES INPUT FROM UNIT 5 AND OUTPUTS TO UNITS 6 AND 8
14 C
15 C GET LOG DIAMETER (IN.), LOG LENGTH (IN.), KERF SIZE (IN.), AND
16 C BOARD THICKNESS (IN.)
17 C
18 C 1 READ (5,102,END=22) D
19 C READ (5,102) L
20 C READ (5,103) K
21 C READ (5,102) T
22 C READ (5,102) H
23 C READ (5,102) L
24 C GET KNOT HALF-ANGLE (DEGREES) AND LOG HALF-TAPER (DEGREES)
25 C
26 C READ (5,102) KNOTS
27 C READ (5,102) LOGS
28 C
29 C GET CORE DEFECT PARAMETERS: DIAMETER (IN.), LINEAR OFFSET (IN.),
30 C AND ANGLE OF OFFSET (DEGREES)
31 C
32 C READ (5,106) CORDIA
33 C READ (5,106) COFSET
34 C READ (5,106) RADIUS
35 C READ (5,106) CANGS
36 C
37 C GET PRICE PER BOARD FOOT FOR EACH GRADE: NUMBER OF KNOTS
38 C READ (5,115) FAS,SEL,ONEC,TUDC,THR
39 C READ (5,115) MAX
40 C READ (5,115) MIN
41 C GET DATE AND CALCULATE RADIUS OF CORE DEFECT
42 C CALL DATE (DATE)
43 C CORDIA = CORDIA/2.
44 C
45 C GET KNOT ANGLES, HEIGHTS, AND LENGTHS
46 C
47 C 10) ? J=1,MAX
48 C PEAK(5,104) ANGLE(J),HEIGHT(J),
49 C CRADIUS = CORDIA/2.
50 C 2 CONTINUE
51 C DO 3 J=1,MAX
52 C READ (5,106) KTLEN(J)
53 C 3 CONTINUE
54 C
55 C CALCULATE LOG RADIUS AT TOP AND BOTTOM: TOTAL LOG CUBIC FEET
56 C
57 C PI = 3.141592653589793115497701183446811322162389320384027461412531242198655587239790035999104946016548515622978029989846869478418284593371075378344147135296212716613444721545442414140414241434144414541464147414841494150
58 C
59 C PI = 3.141592653589793115497701183446811322162389320384027461412531242198655587239790035999104946016548515622978029989846869478418284593371075378344147135296212716613444721545442414140414241434144414541464147414841494150
60 C
61 C LVM = 1/3 * PI * 141592653589793115497701183446811322162389320384027461412531242198655587239790035999104946016548515622978029989846869478418284593371075378344147135296212716613444721545442414140414241434144414541464147414841494150
62 C
63 C CALCULATE CENTRAL CANT SIZE
64 C CANT = D/2 - 2.
65 C
66 C OUTPUT HERDEP
67 C
68 C WRITE (6,116) DAT
69 C WRITE (6,117) L,D,LOGS,LVL,CANT
70 C WRITE (6,101) TUDC
71 C DO J = 1,MAX
72 C KLTEN(J) = KTLEN(J)
73 C 4 CONTINUE
74 C WRITE (6,118) KNOTS
75 C WRITE (6,119) CRADIUS,COFSET,CANGS
76 C WRITE (6,119) K,T
77 C WRITE (6,120) FAS,SEL,ONEC,TUDC,THR
78 C WRITE (6,200) L,D,MAX,CRADIUS,COFSET,CANGS,LVL,K,T,KNOTS,LOGS
79 C WRITE (6,201) KTLEN(J),J=1,MAX
80 C WRITE (6,201) HEIGHT(J),J=1,MAX
81 C WRITE (6,201) ANGLE(J),J=1,MAX
82 C WRITE (6,202) FAS,SEL,ONEC,TUDC,THR
83 C
84 C INITIALIZE ARRAYS FOR CALCULATING HIGH, LOW, AND AVERAGE
85 C YIELDS FOR EACH LOG
86 C
87 C HIGH'S = 0.
88 C SMALL'S = 999.
89 C AVG'S = 0.
90 C
91 C SHU LOG IN 12 DIFFERENT ROTATIONAL POSITIONS, 15 DEGREES APART. TO
92 C DETERMINE THE ORIENTATION GIVING THE HIGHEST YIELD
93 C
94 C DO 21 IMX=1,12
95 C
96 C INITIALIZE FOR EACH POSITION
97 C
98 C NED = 1
99 C TVAL = 0.0
100 C BF = 0.0
101 C
102 C CUTTING ORDER IS FACE 1, FACE 3, FACE 4, AND FACE 2
103 C
104 C DO 5 J=1,5
105 C PFR(J) = 0.0
106 C 5 CONTINUE
107 C
108 C FOR FOUR FACES OF THE LOG...
109 C
110 C DO 16 F=1,4
111 C
112 C THE SLAB FACES ARE CUT TO THE DEFINED CANT
113 C
114 C IF (F.EQ.1.OR.F.EQ.2) R = RB-CANT/2.
115 C
116 C THE CANT FACES ARE KERF CENTERED TO DIVIDE THE CANT
117 C
118 C IF (F.EQ.3.OR.F.EQ.4) R = RB+K/2.
119 C
120 C CUT BOARDS FROM THIS FACE UNTIL FACE IS COMPLETELY CUT. THE LOG IS
121 C CUT FROM THE INSIDE OUT
122 C
123 C 6 CALL KERFC (411,F)
124 C
125 C FOR BOTH SIDES OF THE BOARD...
126 C
127 C DO 10 I=1,2
128 C
129 C INITIALIZE DEFECT ARRAYS, EXCEPT F..(26-1). WHICH HOLDS THE BOARD
130 C DIMENSIONS
131 C
132 C DO 7 J=1,25
133 C FLX(J,1) = -9999.0
134 C FUX(J,1) = -9999.0
135 C FLY(J,1) = -9999.0
136 C FUY(J,1) = -9999.0
137 C
138 C CONTINUE
139 C DO 8 J=1,22
140 C 1D(J,1) = R
141 C
142 C CONTINUE
143 C PUT IN LAME DEFECTS
144 C
145 C CALL LAMEC
146 C
147 C PUT IN KNOTS
148 C
149 C CANT
150 C
151 C CALL KNOTC(HEIGHT(J),KTLEN(J),ANGLE(J))
152 C
153 C DO 9 J=1,MAX
154 C
155 C

```

```

9 CONTINUE
151 C PUT IN CORE DEFECT
153 C CALL COREC (CRANUS, COFSET,CHNGS)
154 C AND GRADE THE FACE
155 C
156 C IF ((1.E0.) OLD + MPG
157 C SUBTRACT BOARD THICKNESS AND PROCESS THE INNER FACE
158 C
159 C SAVE GRADE OF FIRST SIDE OF BOARD
160 C
161 C
162 C
163 C IF ((1.E0.) OLD + MPG
164 C
165 C DISTBD = DISTBD-T
166 C
167 C CONTINUE
168 C
169 C DETERMINE BOARD GRADE AND VALUE BASED ON GRADES OF BOTH FACES
170 C CALL PRTEC (OLD,MPG,TVAL,PERC)
171 C
172 C IMPERSE BOARD COUNT AND CUT ANOTHER BOARD
173 C
174 C
175 C NBD = NBD + 1
176 C GO TO 6
177 C TURN LOG TO NEXT FACE
178 C
179 C TURN LOG TO NEXT FACE
180 C
181 C GO TO (12,14,12,16). F
182 C
183 C TURN LOG BY 180 DEGREES TO CUT THE OPPOSITE FACE.
184 C
185 C DO 13 J=1,MAX
186 C     ANGLE(J) = AND(ANGLE(J)+180.0,360.0)
187 C     CONTINUE
188 C     CHNGS = AND(CHNGS+180.0,360.0)
189 C
190 C GO TO 16
191 C OR TURN 90 DEGREES TO CUT A CANT FACE
192 C
193 C DO 15 J=1,MAX
194 C     ANGLE(J) = AND(ANGLE(J)+90.0,360.0)
195 C     CONTINUE
196 C     CHNGS = AND(CHNGS+90.0,360.0)
197 C     CONTINUE
198 C
199 C LOG HAS BEEN COMPLETELY CUT...RETUPN ANGLES TO ORIGINAL VALUES
200 C
201 C DO 18 J=1,MAX
202 C     ANGLE(J) = AND(ANGLE(J)+270.0,360.0)
203 C     CONTINUE
204 C     CHNGS = AND(CHNGS+270.,360.)
205 C
206 C PRINT RESULTS FOR THIS POSITION
207 C
208 C IPOTAT = INRA15-15
209 C UPITE (6,187) ROTAT
210 C UPITE ('5,111) *PERC(J),J=1,5)
211 C T015 = PERC(1)*PERC(2)+PERC(3)+PERC(4)+PERC(5)
212 C UPITE (6,112) TOTS
213 C UPITE (6,203) ROTAT
214 C UPITE (6,204) (PERC(J),J=1,5)
215 C UPITE (6,284) TOTS
216 C DO 19 J=1,5
217 C     PERC(J) = PERC(J)/1.VOL*100.
218 C     CONTINUE
219 C     TOTS = TOTS*100./1.VOL
220 C     UPITE (6,113) TOTS
221 C     UPITE (6,188) (ANGLE(J),J=1,MAX)
222 C     UPITE (6,189) (PERC(J),J=1,5),TVAL
223 C     UPITE (6,285) (PERC(J),J=1,5)
224 C     UPITE (6,284) TVAL,TOTS
225 C
226 C SAVE HIGH, LOW, AND AVERAGE YIELDS FOR THIS LOG
227 C     AVG = AVG+TVAL
228 C
229 C IF ((TVAL.GT.HIGHS) HIGHS = TVAL
230 C IF ((TVAL.LT.SMALL) SMALLS = TVAL
231 C
232 C ROTATE LOG BY 15 DEGREES AND REPROCESS
233 C
234 C DO 20 J=1,MAX
235 C     ANGLE(J) = AND(ANGLE(J)+15.0,360.0)
236 C     CONTINUE
237 C     CHNGS = AND(CHNGS+15.0,360.0)
238 C     CONTINUE
239 C     PRINT HIGH, LOW, AND AVERAGE YIELDS FOR THIS LOG
240 C
241 C     AVG = AVG/12.
242 C     WRITE (6,114) HIGHS,AVG,SMALLS
243 C
244 C LOG COMPLETELY PROCESSED. READ PARAMETERS FOR NEXT LOG
245 C
246 C
247 C GO TO 1
248 C
249 C STOP
250 C
251 C FORMATS FOR DEVICE 6 (PRINTER)
252 C
253 C FORMAT (12)
254 C     101 FORMAT ('1THERE ARE',I3,' KNOTS IN THIS LOG.', '2(1.4X,
255 C     , 'ANGLE HEIGHT LENGTH')
256 C     102 FORMAT (2(1.4X,
257 C     , 'ANGLE HEIGHT LENGTH')
258 C     103 FORMAT (F5.1)
259 C     104 FORMAT (2F6.2)
260 C     105 FORMAT (3(2X,F6.2))
261 C     106 FORMAT (F6.3)
262 C     107 FORMAT (2(1/), 'ROTATION 1S',I4)
263 C     108 FORMAT (' KNOT ANGLES: ',15F7.2,1/,'13X,15F7.2')
264 C     109 FORMAT (' AVAILABLE SURFACE MEASURE IN EACH GRADE ',SF5.1,
265 C     , 'LOG VALUE IS ',F6.2)
266 C     110 FORMAT (' CORE DEFECT RADIUS ',F6.3,
267 C     , ' ANGULAR OFFSET ',F6.3)
268 C     111 FORMAT (' SURFACE MEASURE/GRADE IS ',SF7.2)
269 C     112 FORMAT (' TOTAL SURFACE MEASURE IS ',FB8.2)
270 C     113 FORMAT (' TOTAL PERCENT YIELD 1S ',FB8.2)
271 C     114 FORMAT (' $ YIELD: HIGH ',F6.2, ' AVERAGE ',F6.2, ' LOW ',F6.2)
272 C     115 FORMAT (' LOG PARAMETERS: LENGTH ',F6.1,' DIA',F6.1,' TAPER
273 C     , 'F6.4, DEGREES LOG VOLUME ',FB8.2, ' CANT SIZE ',F6.2)
274 C     116 FORMAT (' CANT SAWING METHOD ',AB)
275 C     117 FORMAT (' LOG PARAMETERS: LENGTH ',F6.1,' DIA',F6.1,' TAPER
276 C     , 'F6.4, DEGREES LOG VOLUME ',FB8.2, ' CANT SIZE ',F6.2)
277 C     118 FORMAT (' ANGLES MEASURED FROM ZERO DEGREES = ERST FOR A
278 C     , 'VERTICALLY CUTTING SAW, KNOT TAPER',F6.2)
279 C     119 FORMAT (' CUTTING PARAMETERS: KERF ',F6.4, ' BOARD THICKNESS '
280 C     , 'F5.2)
281 C     120 FORMAT (' PRICES PER BOARD FOOT: ',SF10.4)
282 C
283 C FORMATS FOR DEVICE 8 (STORAGE MEDIUM)
284 C
285 C
286 C
287 C
288 C
289 C
290 C
291 C
292 C
293 C
294 C
295 C
296 C
297 C
298 C
299 C
300 C
301 C
302 C
303 C
304 C
305 C
306 C
307 C
308 C
309 C
310 C
311 C
312 C
313 C
314 C
315 C
316 C
317 C
318 C
319 C
320 C
321 C
322 C
323 C
324 C
325 C
326 C
327 C
328 C
329 C
330 C
331 C
332 C
333 C
334 C
335 C
336 C
337 C
338 C
339 C
340 C
341 C
342 C
343 C
344 C
345 C
346 C
347 C
348 C
349 C
350 C
351 C
352 C
353 C
354 C
355 C
356 C
357 C
358 C
359 C
360 C
361 C
362 C
363 C
364 C
365 C
366 C
367 C
368 C
369 C
370 C
371 C
372 C
373 C
374 C
375 C
376 C
377 C
378 C
379 C
380 C
381 C
382 C
383 C
384 C
385 C
386 C
387 C
388 C
389 C
390 C
391 C
392 C
393 C
394 C
395 C
396 C
397 C
398 C
399 C
400 C
401 C
402 C
403 C
404 C
405 C
406 C
407 C
408 C
409 C
410 C
411 C
412 C
413 C
414 C
415 C
416 C
417 C
418 C
419 C
420 C
421 C
422 C
423 C
424 C
425 C
426 C
427 C
428 C
429 C
430 C
431 C
432 C
433 C
434 C
435 C
436 C
437 C
438 C
439 C
440 C
441 C
442 C
443 C
444 C
445 C
446 C
447 C
448 C
449 C
450 C
451 C
452 C
453 C
454 C
455 C
456 C
457 C
458 C
459 C
460 C
461 C
462 C
463 C
464 C
465 C
466 C
467 C
468 C
469 C
470 C
471 C
472 C
473 C
474 C
475 C
476 C
477 C
478 C
479 C
480 C
481 C
482 C
483 C
484 C
485 C
486 C
487 C
488 C
489 C
490 C
491 C
492 C
493 C
494 C
495 C
496 C
497 C
498 C
499 C
500 C
501 C
502 C
503 C
504 C
505 C
506 C
507 C
508 C
509 C
510 C
511 C
512 C
513 C
514 C
515 C
516 C
517 C
518 C
519 C
520 C
521 C
522 C
523 C
524 C
525 C
526 C
527 C
528 C
529 C
530 C
531 C
532 C
533 C
534 C
535 C
536 C
537 C
538 C
539 C
540 C
541 C
542 C
543 C
544 C
545 C
546 C
547 C
548 C
549 C
550 C
551 C
552 C
553 C
554 C
555 C
556 C
557 C
558 C
559 C
560 C
561 C
562 C
563 C
564 C
565 C
566 C
567 C
568 C
569 C
570 C
571 C
572 C
573 C
574 C
575 C
576 C
577 C
578 C
579 C
580 C
581 C
582 C
583 C
584 C
585 C
586 C
587 C
588 C
589 C
590 C
591 C
592 C
593 C
594 C
595 C
596 C
597 C
598 C
599 C
600 C
601 C
602 C
603 C
604 C
605 C
606 C
607 C
608 C
609 C
610 C
611 C
612 C
613 C
614 C
615 C
616 C
617 C
618 C
619 C
620 C
621 C
622 C
623 C
624 C
625 C
626 C
627 C
628 C
629 C
630 C
631 C
632 C
633 C
634 C
635 C
636 C
637 C
638 C
639 C
640 C
641 C
642 C
643 C
644 C
645 C
646 C
647 C
648 C
649 C
650 C
651 C
652 C
653 C
654 C
655 C
656 C
657 C
658 C
659 C
660 C
661 C
662 C
663 C
664 C
665 C
666 C
667 C
668 C
669 C
670 C
671 C
672 C
673 C
674 C
675 C
676 C
677 C
678 C
679 C
680 C
681 C
682 C
683 C
684 C
685 C
686 C
687 C
688 C
689 C
690 C
691 C
692 C
693 C
694 C
695 C
696 C
697 C
698 C
699 C
700 C
701 C
702 C
703 C
704 C
705 C
706 C
707 C
708 C
709 C
710 C
711 C
712 C
713 C
714 C
715 C
716 C
717 C
718 C
719 C
720 C
721 C
722 C
723 C
724 C
725 C
726 C
727 C
728 C
729 C
730 C
731 C
732 C
733 C
734 C
735 C
736 C
737 C
738 C
739 C
740 C
741 C
742 C
743 C
744 C
745 C
746 C
747 C
748 C
749 C
750 C
751 C
752 C
753 C
754 C
755 C
756 C
757 C
758 C
759 C
760 C
761 C
762 C
763 C
764 C
765 C
766 C
767 C
768 C
769 C
770 C
771 C
772 C
773 C
774 C
775 C
776 C
777 C
778 C
779 C
780 C
781 C
782 C
783 C
784 C
785 C
786 C
787 C
788 C
789 C
790 C
791 C
792 C
793 C
794 C
795 C
796 C
797 C
798 C
799 C
800 C
801 C
802 C
803 C
804 C
805 C
806 C
807 C
808 C
809 C
810 C
811 C
812 C
813 C
814 C
815 C
816 C
817 C
818 C
819 C
820 C
821 C
822 C
823 C
824 C
825 C
826 C
827 C
828 C
829 C
830 C
831 C
832 C
833 C
834 C
835 C
836 C
837 C
838 C
839 C
840 C
841 C
842 C
843 C
844 C
845 C
846 C
847 C
848 C
849 C
850 C
851 C
852 C
853 C
854 C
855 C
856 C
857 C
858 C
859 C
860 C
861 C
862 C
863 C
864 C
865 C
866 C
867 C
868 C
869 C
870 C
871 C
872 C
873 C
874 C
875 C
876 C
877 C
878 C
879 C
880 C
881 C
882 C
883 C
884 C
885 C
886 C
887 C
888 C
889 C
890 C
891 C
892 C
893 C
894 C
895 C
896 C
897 C
898 C
899 C
900 C
901 C
902 C
903 C
904 C
905 C
906 C
907 C
908 C
909 C
910 C
911 C
912 C
913 C
914 C
915 C
916 C
917 C
918 C
919 C
920 C
921 C
922 C
923 C
924 C
925 C
926 C
927 C
928 C
929 C
930 C
931 C
932 C
933 C
934 C
935 C
936 C
937 C
938 C
939 C
940 C
941 C
942 C
943 C
944 C
945 C
946 C
947 C
948 C
949 C
950 C
951 C
952 C
953 C
954 C
955 C
956 C
957 C
958 C
959 C
960 C
961 C
962 C
963 C
964 C
965 C
966 C
967 C
968 C
969 C
970 C
971 C
972 C
973 C
974 C
975 C
976 C
977 C
978 C
979 C
980 C
981 C
982 C
983 C
984 C
985 C
986 C
987 C
988 C
989 C
990 C
991 C
992 C
993 C
994 C
995 C
996 C
997 C
998 C
999 C

```

\*NEW STATEMENT NUMBERS

SAVE HIGH, LOW, AND AVERAGE YIELDS FOR THIS LOG  
AVG = AVG+TVAL  
AVG = AVG+TVAL

1 \*20 247  
2 49 51  
3 52 54  
4 71 73

5	184	*106
6	*123	177
7	132	*137
8	138	*148
9	149	*151
10	127	168
11	123	*181
12	181	*185
13	185	*187
14	181	*193
15	193	*195
16	119	181
18	201	*203
19	116	*218
20	234	*236
21	94	*238
22	28	*249
103	40	*253
101	78	*254
102	29	21
103	22	*257
184	59	*258
185	72	*259
186	33	34
187	289	*261
108	221	*262
169	222	*263
119	75	*265
111	219	*267
112	112	*268
113	229	*269
114	243	*270
115	39	1271
116	63	*272
117	69	*273
118	74	*275
119	76	1277
120	77	*279
206	78	*283
281	79	*288
282	82	*285
283	213	*286
284	214	*287
285	223	*288

\*\*\* VARIABLES \*\*\*

ADDN	186	188	191	196	202	204	206	209	213	217	221	224	227	230	233	237
ANGLF	7	*58	72	69	114	204	235	237	134	135	136	139	140	145	146	147
AVCS	*89	*228	*242	243	150	*186	*194	*202	202	204	206	207	208	209	210	211
BIDLEN	19	*108														
BP	10															
BOARDW	10															
CANES	*35	75	78	155	*188	*196	*204	*207	133	134	135	136	137	138	139	140
CANT	10	*64	69	114	204	235	237	238	133	134	135	136	137	138	139	140
COFSET	*34	75	78	155												
CORDIA	*33	45														
COREC	155															
CRADUS	*45	75	78	155												
D	10	*20	58	64	69	70										
DAT	7	44	68													
DATE	44															
DICTED	10	*167	114	118	123	181										
F	9	*110	114	118	123	181										
FAS	10	*39	77	82												
FLY	7		10	*133												
FLT	7	10	*135													
FUX	7	10	*134													
FUY	7	10	*136													
GRADE	159															
HEIGHT	*50	72	86	156												
HIGHS	*87	*229	243													
ID	9	163														
INX	*94	266														

23  
\*\*\*

C SUBROUTINE 'KERF' CUTS A BOARD FROM THE LOG. THE BOARD WIDTH IS  
2 C DEFINED AT THE CENTER, AND THE BOARD LENGTH FPIR1 THE BUTTON  
3 C OF THE LOG  
4 C

5 C  
6 C SUBROUTINE KERF (\*,F)  
7 C  
8 IMPLICIT REAL (K,L)  
9 INTEGER ID(22,2),NODFC(2),F  
10 REAL FLX(26,2),FLY(26,2),FLU(26,2)  
11 COMMON /BOARD/, L,D,K,T,LOGS,RB,DISTD,BORDU,KNOTS,CANT,R,BF  
12 DEFDEF / NODFC,FLX,FLY,FUY,FLU,TD  
13 C  
14 C ALL BOARDS MUST BE AT LEAST 48 INCHES LONG, 2.5 INCHES WIDE AT  
15 C THE TOP, AND 3 INCHES WIDE AT THE CENTER. IN ADDITION, THE TOTAL  
16 C AMOUNT OF LOGUE FROM THE CENTER TO THE TOP IS LIMITED TO 4 INCHES.  
17 C THE BOARD IS CUT BACK IN ONE FOOT INCREMENTS IF THE WIDTH OR LENGTH  
18 C TESTS FAIL. IF THE LENGTH TEST FAILS, ANOTHER BOARD IS CUT FROM  
19 C THIS FACE  
20 C  
21 C  
22 C FIND OUTER FACE OF BOARD  
23 C  
24 C R1 = D/2,  
25 C 1 R = R-T-K  
26 C BOLEN = L  
27 C  
28 C IF THIS FACE OF THE LOG IS COMPLETELY CUT.. EXIT  
29 C  
30 C IF (R.LT.0.) RETURN  
31 C DISTD = PB-8  
32 C POSLEN = R\*TANH(PAD/90.-LOGS\$)  
33 C  
34 C  
35 C CHECK FOR BOARDS SHORTER THAN THE LOG DUE TO LOG TAPER  
36 C  
37 C IF (POSLEN.LT.BOLEN) BOLEN = POSLEN  
38 C  
39 C BOARDS FIRST BF AT LEAST 4 FEET LONG  
40 C  
41 C  
42 C 2 IF (BOLENS.LT.BOLEN) GO TO 1  
43 C DETERMINE HALF-WIDTH AT MID-LENGTH OF BOARD  
44 C RM = D/2 + (L-BOLENS)/TANH(PAD/LOGS\$)  
45 C IF (DISTD.GT.RM) GO TO 3  
46 C BORDDU = SQR(RM\*\*2-DISTD\*\*2)  
47 C  
48 C FOR CANT CUTS, BOARD IS NO WIDER THAN CANT  
49 C  
50 C  
51 C IF (F.EQ.3.OR.F.EQ.4) BORDRU = APRIH(CANT/2.,BORDDU)  
52 C  
53 C BOARD MUST BE AT LEAST 3 INCHES WIDE AT MID-LENGTH  
54 C  
55 C IF (BORDDU.LT.1.5) GO TO 3  
56 C  
57 C DETERMINE HALF-WIDTH AT TOP OF BOARD  
58 C  
59 C RT = D/2 + (L-BOLENS)\*TANH(PAD/LOGS\$)  
60 C  
61 C IF (DISTD.GT.RT) GO TO 3  
62 C UT = SQR(RT\*\*2-DISTD\*\*2)  
63 C IF (F.EQ.3.OR.F.EQ.4) UT = APRIH(CANT/2.,UT)  
64 C  
65 C BOARD MUST BE AT LEAST 2.5 INCHES WIDE AT THE TOP  
66 C  
67 C IF (UT.LT.1.25) GO TO 3  
68 C  
69 C LIMIT THE TOTAL WIDTH TO AVOID GENERATING MORE DEFECTS THAN THE  
70 C GRADING PROGRAM CAN HANDLE. TOTAL WIDTH OF LOGUE MUST BE LESS  
71 C THAN 4 INCHES  
72 C  
73 C LOGW = BORDDU-UT  
74 C IF (LOGW.GT.2.0) GO TO 3  
75 C

## Subroutine KERF

CONVERT BOARD WIDTH AND LENGTH TO EVEN QUARTER INCHES  
76 C  
77 C  
78 C BBOARDU = INT(BBOARDU+.1).  
79 C BOLEN = INT(BOLEN\*.4.).  
80 C  
81 C DEFINE BOARD EDGES IN QUARTER INCH UNITS  
82 C  
83 FLX(26,1) = BOLEN\*.4.  
84 FLY(26,1) = BBOARDU\*.8.  
85 FLX(26,1) = 0.8  
86 FLY(26,1) = 0.0  
87 C  
88 C CALCULATE UNROUNDED BOARD FEET  
89 C  
90 BF = BOLEN\*BBOARDU\*.2.\*T/144.  
91 RETURN  
92 C  
93 C CUT BACK BOARD BY ONE FOOT  
94 C  
95 3 BOLEN = BOLEN-.12.  
96 GO TO 2  
END  
\*\*\*\* STATEMENT NUMBERS \*\*\*\*  
1 \*26 42  
2 \*42 96  
3 4. 61 67 74 \*95  
\*\*\*\* VARIABLES \*\*\*\*  
AMIN1 BULEN 11 \*27 \*38 42 46 60 \*79 83 98 \*95  
BF BBOARDU 11 \*98 \*52 56 73 \*78 84 98  
CANT CANT 11 48 63 25 46 68  
DISTD DISTD 11 \*32 47 48 61 62  
F F 6 9 52 63  
FLX FLY 10 11 \*83 11 \*84  
FUX FUX 10 11 \*85  
G LOGS 11 33 46 60  
HODEFC POSLEN 11 \*33 38  
K K 11 26  
KERF RAD 11 33 46 60  
KERF KB 11 32  
KERF RI \*25  
KERF RH \*46 47 48  
KERF PT \*60 61 62  
KERF SQT 11 27 46 60  
KERF TAN 11 37 46 60  
KERF WAN 11 \*73 74 67 73  
KERF INT \*63 67 73

### Subroutine WANE

```

1 C SUBROUTINE 'WANE' LOCATES WANE DEFECTS ON THE BOARD
2 C
3 C
4 C SUBROUTINE WANE
5 C
6 C THE BOARD EDGE HAS A NEW WANE DEFECT IF THE WIDTH DROPS BY 1/2
7 C INCH (.14 INCH FOR EACH SIDE) FROM THAT AT THE MIDDLE. THE BOARD
8 C IS SYMMETRICAL SO THE WANE DEFECTS OCCUR AT THE SAME PLACE ON
9 C BOTH SIDES OF THE BOARD
10 C
11 C IMPLICIT REAL (K,L)
12 C INTEGER I,D(22,2),NODEFC(2)
13 C REAL FLX(26,2),FLY(26,2),FUX(26,2)
14 C COMMON /BOARD/,L,D,K,T,L,DS,B,DISBD,BDLEN,BORDU,KNOTS,CANT,R,BF
15 C /DEFEL/
16 C
17 C
18 C CON1 = TAN(BAD(LOG$))
19 C CON2 = DISTBD*LOG$)
20 C IF ((PB-SORT((CON2+BOARDUM=2))>CON1.GE.BDLEN) RETURN
21 C QLNH = BALEM2.
22 C V = 0.
23 C
24 C FIND THE PLACE ALONG THE BOARD WHERE THE HALF-WIDTH HAS DECREASED
25 C BY 1/4 INCH. THE DEFECT ENDS HERE AND EXTENDS FROM THE END OF THE
26 C LAST WANE DEFECT (OR FROM THE MIDDLE OF THE BOARD)
27 C
28 C V = 0.25
29 C SDR = SORT((CON2+(BOARDU-V)*V)*2)
30 C H = INT((TAN1*(PB-SDR)/4.-CON1.BDLEM)*2)
31 C NODEFC(1) = NODEFC(1)+1
32 C H = NODEFC(1)
33 C
34 C LIMIT NUMBER OF DEFECTS TO MAXIMUM ALLOWED BY GRADING PROGRAM (22)
35 C
36 C IF (H.GT.22) GO TO 2
37 C
38 C SAVE WANE COORDINATES FOR GRADING PROGRAM
39 C
40 C FLX(M,1) = 0.
41 C FUY(M,1) = V*4.+1.
42 C FLX(M,1) = H
43 C FLX(M,1) = OLDH
44 C
45 C
46 C SINCE THE BOARD IS SYMMETRICAL, USE THE PREVIOUSLY GENERATED HEIGHT
47 C FOR THE WANE DEFECTS ON THE OTHER EDGE OF THE BOARD
48 C
49 C 2 NODEFC(1) = NODEFC(1)+1
50 C H = NODEFC(1)
51 C IF (H.GT.22) GO TO 3
52 C FLX(M,1) = BOARDUM*-V*4.-1.
53 C FUY(M,1) = BOARDUM*2.
54 C FUX(M,1) = H
55 C FLX(M,1) = OLDH
56 C
57 C WHEN THE DEFECT EXTENDS TO THE END OF THE BOARD, EXIT
58 C
59 C
60 C 3 IF (H.GE.BDLEM*4.) RETURN
61 C
62 C THE NEXT WANE DEFECT BEGINS WHERE THIS ONE ENDS
63 C
64 C OLDH = H
65 C GO TO 1
66 C RETURN
67 C

```

\*\*\* STATEMENT NUMBERS \*\*\*

WANE	1	*28	65
WANE	2	36	*49
WANE	3	51	*68
*** VAR TABLES ***			
ARINI	38	28	21
BRLEN	14	28	30
BF	14	28	52
BOARDU	14	28	53
CANT	14		
CON1	*18	28	39
CONC	*19	28	29
D	14	19	
DISBD	14	19	
FLX	13	14	*43
FLY	13	14	*48
FUX	13	14	*42
FUY	13	14	*53
H	*30	42	54
ID	12	14	*44
INT			
K	14		
KNOTS			
L	14		
LOG\$		18	
M	*32	36	48
N	56	56	41
NODEFC			
OLDH		11	*31
P	14	42	43
RAD	13		
PA	14	28	40
SDR		29	39
SORT		20	29
TAN	14		
V	*22	*28	29
WANE	4		

\*\*\* VAR TABLES \*\*\*

## Subroutine KNOT

```

1 C SUBROUTINE 'KNOT' LOCATES THE KNOT DEFECTS ON THE BOARD FACE.
2 C ANGLES ARE MEASURED CLOCKWISE WITH 0 DEGREES DEFINED AS THE
3 C LINE FROM THE CENTER OF THE LOG PERPENDICULAR TO THE BOARD FACE
4 C
5 C SUBROUTINE KNOT (KHIGH,KLEN,KTANGS)
6 C
7 C IMPLICIT REAL (X,L)
8 C INTEGER 10,122,21, NODEFC(2), FUX(26,2), FUY(26,2),
9 C REAL FUX(26,2), FLY(26,2), FUX, FUY, ID,
10 C COMMON /DEFEC/ NODEFC, FLX, FLY, FUX, FUY,
11 C /BOARD/ L,D,K,T,LOGS,RB,DISTBD,BULEN,BOARDL,KMOTS,CANT,R,BF
12 C
13 C SEE IF THE KNOT IS LONG ENOUGH TO REACH THE BOARD FACE
14 C
15 C IF (KLEN.LT.DISTBD) RETURN
16 C
17 C CALCULATE KNOT DEFECT HALF-LENGTH AS PROJECTED ON THE FACE, LOG
18 C RADIUS AT KNOT HEIGHT, AND BOARD HALF-WIDTH ANGLE
19 C
20 C IF (KLEN.GT.LRADUS) KLEN = LRADUS
21 C DY = DISTBD*TAN(RAD(KNOT$))
22 C LRADUS = RB-KHIGH*TAN(RAD(KNOT$))
23 C BULIDS = DEG(ATAN(BORDUD*DISTBD))
24 C
25 C THE KNOT EFFECTIVELY ENDS AT THE OUTER EDGE OF THE LOG
26 C
27 C IF (KLEN.GT.LRADUS) KLEN = LRADUS
28 C SINCE THE LOG IS TAPERED, THE OUTER BOARDS MAY NOT BE THE FULL LOG
29 C LENGTH, SO KNOTS ABOVE THE END OF THE BOARD ARE NOT CONSIDERED
30 C
31 C IF (LRADUS.LT.DISTBD) RETURN
32 C
33 C ANGLE OF ROTATION PUTS KNOT COMPLETELY OUTSIDE BOARD
34 C IF ((KTANGS.GT.180.), AND. ((KTANGS-KNOT$).LT.(360.-BULIDS))), RETURN
35 C IF ((KTANGS.LE.180.), AND. ((KTANGS-KNOT$).GT.BULIDS)), RETURN
36 C
37 C IF ((KTANGS.GT.180.), AND. ((KTANGS-KNOT$).LT.(360.-BULIDS))), RETURN
38 C IF ((KTANGS.LE.180.), AND. ((KTANGS-KNOT$).GT.BULIDS)), RETURN
39 C
40 C FIND THE QUADRANT IN WHICH THE KNOT LIES (ASSUME QUADRANT ONE)
41 C
42 C QUADRANT TWO
43 C
44 C IF ((KTANGS.LT.180.), AND. ((KTANGS-KNOT$).GT.BULIDS)), GO TO 3
45 C
46 C QUADRANT THREE
47 C
48 C IF ((KTANGS.GE.180.), AND. ((KTANGS-KNOT$).LT.(360.-BULIDS))),
49 C GO TO 4
50 C
51 C QUADRANT FOUR
52 C
53 C IF ((KTANGS.GT.270.)) GO TO 5
54 C
55 C QUADRANT ONE
56 C
57 C SEE IF KNOT CENTER IS WITHIN ONE KNOT HALF-ANGLE OF 0 DEGREES
58 C
59 C CALCULATE DISTANCE TO FACE ALONG BOTH SIDES OF KNOT
60 C
61 C IF (KTANGS-KNOT$.LT.0.), GO TO 2
62 C
63 C
64 C ANGBD1 = DISTBD*COS(RAD((KTANGS-KNOT$)))
65 C ANGBD2 = DISTBD*COS(RAD((KTANGS-KNOT$)))
66 C KNOT IS NOT LONG ENOUGH TO REACH FACE
67 C
68 C
69 C IF (KLEN.LT.ANGBD1) RETURN
70 C
71 C FIND INTERSECTION OF NEAR SIDE OF KNOT AND FACE
72 C
73 C XR = BORDUD+DISTBD*TAN(RAD((KTANGS-KNOT$)))
74 C IF (KLEN.LT.ANGBD2) GO TO 1
75 C
76 C FAR SIDE OF KNOT REACHES FACE
77 C
78 C XL = BORDUD-DISTBD*TAN(RAD((KTANGS-KNOT$)))
79 C FULWID = (XR-XL)/2.
80 C GO TO 8
81 C FAR SIDE OF KNOT DOES NOT REACH FACE
82 C
83 C 1 XL = BORDUD-SORT(KLEN**2-DISTBD**2)
84 C FULWID =
85 C . DISTBD*TAN(RAD((KTANGS-KNOT$))-DISTBD*TAN(RAD((KTANGS-KNOT$)))
86 C
87 C GO TO 8
88 C KNOT IS WITHIN ONE KNOT HALF-ANGLE OF 0 DEGREES
89 C
90 C 2 XL = AMINISORT((KLEN**2-DISTBD**2),DISTBD*TAN(RAD((KNOTS-KTANG$)))
91 C
92 C XL = BORDUD-XL
93 C XR = AMINISORT((KLEN**2-DISTBD**2),DISTBD*TAN(RAD((KNOTS-KTANG$)))
94 C
95 C KNOTS NEARLY PERPENDICULAR HAVE THEIR WIDTH ESTIMATED BY THEIR
96 C LENGTH
97 C
98 C FULWID = DY*2.
99 C
100 C IF (KLEN.LT.ANGBD1) RETURN
101 C
102 C 100 GO TO 8
103 C QUADRANT TWO
104 C
105 C 102 DISTBD*COS(RAD((KTANGS-KNOT$)))
106 C
107 C 103 SHOT TO REACH FACE
108 C
109 C 104 IF ((LEN.LT.ANGBD1)**2-DISTBD**2)
110 C
111 C 105 IF ((KTANGS-KNOT$).GE.89.5)
112 C
113 C 106 IF ((KTANGS-KNOT$).LT.89.5)
114 C
115 C 107 -DISTBD*TAN(PAT((KTANGS-KNOT$)))
116 C
117 C 108 GO TO 8
118 C
119 C 109 OUTSIDE! THREE
120 C
121 C 110 IF (KLEN.LT.ANGBD1) RETURN
122 C
123 C 111 IF (BORDUD-SORT(KLEN**2-DISTBD**2))
124 C
125 C 112 IF ((KTANGS-KNOT$).LE.270.5)
126 C
127 C 113 IF ((KTANGS-KNOT$).LT.270.5)
128 C
129 C 114 IF ((KTANGS-KNOT$).LT.360.-KTANGS+KNOT$)
130 C
131 C 115 OUTPHANT FOUR
132 C
133 C 116 IF ((360.-KTANGS-KNOT$.LT.0.)) GO TO 7
134 C
135 C 117 ANGBD1 = DISTBD*COS(RAD((360.-KTANGS-KNOT$)))
136 C
137 C 118 IF (KLEN.LT.ANGBD1) RETURN
138 C
139 C 119 IF ((KTANGS-KNOT$).LT.360.-KTANGS+KNOT$)
140 C
141 C 120 IF ((KTANGS-KNOT$).LT.360.-KTANGS+KNOT$)
142 C
143 C 121 FULWID = (XR-XL)/2.
144 C
145 C 122 FAR SIDE OF KNOT DOES NOT REACH FACE
146 C
147 C 123 IF ((LEN.LT.ANGBD1)**2-DISTBD**2)
148 C
149 C 124 FULWID = BORDUD+SORT(KLEN**2-DISTBD**2)
150 C
151 C 125 DISTBD*TAN(PAD((360.-KTANGS-KNOT$)))
152 C
153 C

```

```

154 C KNOT IS WITHIN ONE YNOT HALF-ANGLE OF 0 DEGREES
155 C
156 C 7 X0 = ANH(1.0*TANH(X0*L/2-DISTBD**2));
157 C DISTBD=TANH(PAD*(X0*TANG$**NOTS-360.0))
158 C X0 = BOARD-X0
159 C Y0 = MIN(X0,SQRT(X0*L/2-DISTBD**2));
160 C DISTBD=TANH(PAD*(X0.-X0*TANG$**NOTS))
161 C X0 = BOARD+D0*X0
162 C FULWID = D1*D0*X0
163 C
164 C IF A KNIT COMPLETELY INTERSECTS THE FACE, ITS LENGTH IS EXACTLY AS
165 C MEASURED ON THE FACE; OTHERWISE, ITS LENGTH IS ESTIMATED AS A
166 C PROPORTION OF ITS PROJECTED LENGTH
167 C
168 C S BY = ANH(1.0*D0*(X0-L1)*2.*FULWID)
169 C X0 = HIGH-DR
170 C Y0 = HIGH+DR
171 C
172 C DEFECT LOCATIONS FOUND. CHECK THAT THEY DO NOT EXTEND BEYOND THE
173 C BOARD EDGES AND CENTER TO QUARTER INCH UNITS FOR THE GRADING
174 C PROGRAM
175 C
176 C X0 = ANH(0.0*TAN(L1*A2.))
177 C Y0 = ANH((INT(BD0*D0*B)+(INT(Y0*A2.))-INT(X0*A2.+1.))
178 C Y0 = CMR(0.0, INT(Y0*A2.))
179 C Y1 = CMR(0.0, INT(Y0*A2.))
180 C LIMIT NUMBER OF DEFECTS TO MAXIMUM ALLOWED BY GRADING PROGRAM (22)
181 C
182 C
183 C NODEF(1), NODEF(1)+1
184 C IF (NODEF(1).GT.32) RETURN
185 C
186 C CA 4 : INPUT COORDINATES FOR GRADING, PROGRAM
187 C
188 C ID(NOREFC(1,1)) = 3
189 C FL=NOREFC(1,1)+1
190 C FL=NOREFC(1,1)+1
191 C FL=NOREFC(1,1)+1
192 C FL=NOREFC(1,1)+1
193 C PETURE
194 C END

```

#### \*\*\* STATEMENT NUMBERS \*\*\*

	L	C	STATEMENT NUMBER
1	1	1	48
2	1	1	151
3	3	4	*164
4	4	43	*120
5	5	54	+133
6	6	14	*142
7	7	135	*156
C	289	87	169
			116
			129
			144
			151
			*168
			142
			148
			158
			161
			177
			142
			148
			158
			161
			142
			148
			158
			161
			135
			65
			66
			104
			120
			134
			135
			91
			94
			85
			125
			134
			125
			124
			123
			122
			123
			124
			125
			126
			127
			128
			129
			130
			131
			132
			133
			134
			135
			136
			137
			138
			139
			140
			141
			142
			143
			144
			145
			146
			147
			148
			149
			150
			151
			152
			153
			154
			155
			156
			157
			158
			159
			160
			161
			162
			163
			164
			165
			166
			167
			168
			169
			170
			171
			172
			173
			174
			175
			176
			177
			178
			179
			180
			181
			182
			183
			184
			185
			186
			187
			188
			189
			190
			191
			192
			193
			194
			195
			196
			197
			198
			199
			200
			201
			202
			203
			204
			205
			206
			207
			208
			209
			210
			211
			212
			213
			214
			215
			216
			217
			218
			219
			220
			221
			222
			223
			224
			225
			226
			227
			228
			229
			230
			231
			232
			233
			234
			235
			236
			237
			238
			239
			240
			241
			242
			243
			244
			245
			246
			247
			248
			249
			250
			251
			252
			253
			254
			255
			256
			257
			258
			259
			260
			261
			262
			263
			264
			265
			266
			267
			268
			269
			270
			271
			272
			273
			274
			275
			276
			277
			278
			279
			280
			281
			282
			283
			284
			285
			286
			287
			288
			289
			290
			291
			292
			293
			294
			295
			296
			297
			298
			299
			300
			301
			302
			303
			304
			305
			306
			307
			308
			309
			310
			311
			312
			313
			314
			315
			316
			317
			318
			319
			320
			321
			322
			323
			324
			325
			326
			327
			328
			329
			330
			331
			332
			333
			334
			335
			336
			337
			338
			339
			340
			341
			342
			343
			344
			345
			346
			347
			348
			349
			350
			351
			352
			353
			354
			355
			356
			357
			358
			359
			360
			361
			362
			363
			364
			365
			366
			367
			368
			369
			370
			371
			372
			373
			374
			375
			376
			377
			378
			379
			380
			381
			382
			383
			384
			385
			386
			387
			388
			389
			390
			391
			392
			393
			394
			395
			396
			397
			398
			399
			400

### Subroutine CORE

```

1 C SUBROUTINE 'CORE' LOCATES THE CORE DEFECT ON THE BOARD FACE
2 C
3 C SUBROUTINE COREC (RADIUS,COFSET,CHNG$)
4 C
5 C IMPLICIT REAL (K,L)
6 C REAL FFLX(26,2),FLY(26,2),FUX(26,2),FUY(26,2)
7 C INTEGER NODEFC(2),ID(22,2)
8 C COMMON /BOARD/ L,D,K,T,LOGS,RB,DISTBD,BOLEN,BOARDU,KNOTS,CANT,R,BF
9 C /DEFEC/ NODEFC,FIX,FLX,FLY,FUI,FUY,ID
10 C
11 C FIND DISTANCE TO THE BOARD FACE FROM THE CENTER OF THE CORE. EXIT
12 C IF THE BOARD FACE IS BEYOND THE CORE RADIUS
13 C
14 C IF (RADIUS.LE.1.D-10) RETURN
15 C TDIST = DISTBD-COFSET*RAD*(CHNG$)
16 C IF (TDIST.LT.0.D0) RETURN
17 C ID(NODEFC(1)) = NODEFC(1)+1
18 C
19 C LIMIT NUMBER OF DEFECTS TO MAXIMUM ALLOWED BY GRADING PROGRAM (22)
20 C
21 C IF (NODEFC(1).GT.22) RETURN
22 C
23 C LOCATE THE CENTER POINT OF THE INTERSECTION OF THE BOARD FACE AND
24 C THE CORE DEFECT. CALCULATE THE DISTANCE ON THE BOARD FROM THE
25 C CENTER POINT TO THE EDGES OF THE CORE DEFECT. THE CORE DEFECT
26 C IS NOT TAPERED. SO IT WILL EXTEND THE FULL LENGTH OF THE BOARD
27 C
28 C CONST = SORT((RADIUS**2-TDIST**2))
29 C TDIRTR COFSET$IN(BAD(CHNG$))
30 C XR = BOARDU*CONST-TDIRTR
31 C XL = BOARDL*CONST-TDIRTR
32 C
33 C SAVE CORE DEFECT COORDINATES FOR GRADING PROGRAM
34 C
35 C
36 C FIX(NODEFC(1),1) = INT(BOLEN/4.)
37 C FLX(NODEFC(1),1) = 0.0
38 C FUY(NODEFC(1),1) = INT((INT(BOARDW),) INT((R44,4+1,1),)
39 C FLY(NODEFC(1),1) = APPROX(0, INT(XL/44,)))
40 C ID(NODEFC(1),1) = 7
41 C RETURN
42 C END

```

```

1 C
2 C
3 C KNOTS
4 C COREC
5 C COREC
6 C COREC
7 C COREC
8 C COREC
9 C COREC
10 C COREC
11 C COREC
12 C COREC
13 C COREC
14 C COREC
15 C COREC
16 C COREC
17 C COREC
18 C COREC
19 C COREC
20 C COREC
21 C SIN
22 C 3
23 C 9
24 C LOGS
25 C 9
26 C NODEFC
27 C 8
28 C R
29 C 9
30 C RAD
31 C 16
32 C RB
33 C 9
34 C 39
35 C 3
36 C 29
37 C 31
38 C 32
39 C 30
40 C 31
41 C 32
42 C 31

```

NAME	VAR/FILES	TYPE
RFLX	39	
RFUI	38	
BOLEN	9	36
ID	9	36
DISTBD	9	16
FLX	7	9
FLY	7	9
FUY	7	9
ID	9	9
INT	36	39

U.S. Forest Products Laboratory.

Programs for computer simulation of hardwood log sawing, by W. K. Adkins, D. B. Richards, D. W. Lewis, and E. H. Bulgrin. Madison, Wis., FPL, 1980.

57 p. (USDA FS Res. Pap. FPL 357)

Four computer programs were developed at the University of Kentucky as simulation models for investigating factors affecting sawn log values over four hardwood sawing methods: quadrant sawing, cant sawing, decision sawing, and live sawing with rerip for grade. The programs are listed along with information on the sawing methods, model assumptions, and program organization.

U.S. Forest Products Laboratory.

Programs for computer simulation of hardwood log sawing, by W. K. Adkins, D. B. Richards, D. W. Lewis, and E. H. Bulgrin. Madison, Wis., FPL, 1980.

57 p. (USDA FS Res. Pap. FPL 357)

Four computer programs were developed at the University of Kentucky as simulation models for investigating factors affecting sawn log values over four hardwood sawing methods: quadrant sawing, cant sawing, decision sawing, and live sawing with rerip for grade. The programs are listed along with information on the sawing methods, model assumptions, and program organization.

U.S. Forest Products Laboratory.

Programs for computer simulation of hardwood log sawing, by W. K. Adkins, D. B. Richards, D. W. Lewis, and E. H. Bulgrin. Madison, Wis., FPL, 1980.

57 p. (USDA FS Res. Pap. FPL 357)

Four computer programs were developed at the University of Kentucky as simulation models for investigating factors affecting sawn log values over four hardwood sawing methods: quadrant sawing, cant sawing, decision sawing, and live sawing with rerip for grade. The programs are listed along with information on the sawing methods, model assumptions, and program organization.

U.S. Forest Products Laboratory.

Programs for computer simulation of hardwood log sawing, by W. K. Adkins, D. B. Richards, D. W. Lewis, and E. H. Bulgrin. Madison, Wis., FPL, 1980.

57 p. (USDA FS Res. Pap. FPL 357)

Four computer programs were developed at the University of Kentucky as simulation models for investigating factors affecting sawn log values over four hardwood sawing methods: quadrant sawing, cant sawing, decision sawing, and live sawing with rerip for grade. The programs are listed along with information on the sawing methods, model assumptions, and program organization.

## Subroutine GRADE

```

1 C SUBROUTINE 'GRADE' CALLS THE U.S. FOREST PRODUCTS LABORATORY
2 C GRADING PROGRAM
3 C
4 C
5 C SUBROUTINE GRADE (NPG)
6 C
7 C INTEGER 1D(22,2), NODEFC(2), IX(22)
8 C REAL FLY(26,2), FUY(26,2), FUX(26,2)
9 C COMMON /DEFEC/ NODEFC, FLX, FLY, FUX, FUY, ID
10 C
11 C
12 C H = NODEFC(1,1)
13 C IF (H .GT. 22) GO TO 3
14 C IF (M .EQ. 0) GO TO 2
15 C DO 1 J=1,M
16 C   1X(J) = 1D(J,1)
17 C 1 CONTINUE
18 C NPG = 1
19 C CALL GRD (M,IX,NPG,SM)
20 C RETURN
21 C
22 C IF NUMBER OF DEFECTS IS MORE THAN GRADING PROGRAM CAN HANDLE,
23 C THE GRADE IS 36.99
24 C
25 C 3 NPG = 5
26 C WRITE (6,100)
27 C 100 FORMAT (F6.100, DEFECT LIMIT EXCEEDED*)
28 C RETURN;
29 C END

```

## Subroutine PRICE

```

1 C SUBROUTINE 'PRICE' DETERMINES THE BOARD GRADE BASED ON THE GRADES
2 C OF BOTH SIDES (OLD,NPG) AND CALCULATES THE BOARD VALUE
3 C
4 C
5 C SUBROUTINE PRICEC (OLD,NPG,TVAL,PERC)
6 C
7 C IMPLICIT REAL (K,L)
8 C INTEGER OLD
9 C COMMON /PERC/ FOS, SEL, ONEC, TWOE, THREE
10 C           /BOARD/ L,D,K,T,LUGS,RB,DISTBD,BULEN,BORDUS,KNOTS,CANT,R,BF
11 C
12 C DETERMINE THE BOARD GRADE
13 C
14 C
15 C IF (OLD.EQ.5.0P, NPG.EQ.5) GO TO 3
16 C IF (OLD.EQ.2.0P, NPG.EQ.2) GO TO 1
17 C IF (OLD.EQ.1.0P, NPG.EQ.1) GO TO 4
18 C IF (OLD.EQ.0.3P, NPG.EQ.0.3) GO TO 4
19 C IF (NPG .EQ. 1) GO TO 8
20 C IF (NPG .EQ. 4) GO TO 8
21 C IFPG = 2
22 C GO TO 4
23 C 1 IF (OLD.EQ.4.0P, NPG.EQ.4) GO TO 2
24 C
25 C
26 C AND CALCULATE THE BOARD VALUE
27 C
28 C
29 C
30 C
31 C 4 GO TO (5,6,7,8,9), NPG
32 C 5 V = BF*PERC
33 C 6 V = BF*SEL
34 C 7 V = BF*TREC
35 C 8 V = BF*TLOC
36 C 9 V = BF*TMPC
37 C 10 PERC(NPG) = BF+PERC(NPG)
38 C 11 V = FF+TMPC
39 C 12 V = FF+TLOC
40 C 13 V = FF*TMPC
41 C 14 V = FF*TLOC
42 C INC PERC SURFACE MEASURE IN THIS GRADE AND THE TOTAL LOG VALUE
43 C
44 C 15 PERC(NPG) = BF+PERC(NPG)
45 C 16 TVAL = TVAL+V
46 C RETURN
47 C
48 C END

```

\*\*\* STATEMENT NUMBERS \*\*\*

```

1 15 *17
2 14 *18
3 13 *25
100 26 *27

```

\*\*\* VARIABLES \*\*\*

FLX	8	9
FLY	9	9
FUX	8	9
FUY	8	9
GRADE	5	
GPT	19	
ID	7	9
IX	7	16
J	*15	16
K	*12	13
NODEFC	7	9
NPG	5	*18
SM	19	*25

\*\*\* STATEMENT NUMBERS \*\*\*

1	17	*23
2	23	*26
3	16	*28
4	18	22
5	32	*33
6	32	*35
7	32	*37
8	20	32
9	32	*39
10	34	36
11	36	40

	VARIABLES	MEM
BLEN	16	33
DEG	16	35
DONRNU	16	37
CNT	16	39
D	16	41
DISTBD	16	43
FRS	16	33
K	16	9
KNOTS	16	
L	16	
LOSS	16	
NPV0	19	
NPV1	5	16
NPV2	5	17
NPV3	5	18
NPV4	5	19
NPV5	5	20
NPV6	5	21
NPV7	5	23
NPV8	5	24
NPV9	5	25
NPV10	5	26
NPV11	5	27
NPV12	5	28
NPV13	5	29
NPV14	5	30
NPV15	5	31
NPV16	5	32
OLD	5	8
ONEC	16	16
PERC	5	16
PRICEC	5	19
R	16	9
RB	16	35
SEL	16	35
T	16	
THR	16	41
TVAL	5	46
TLOC	16	39
V	35	37
		39
		41
		46

### Function DEG

```

1 C FUNCTION 'DEG' CONVERTS ITS ARGUMENT (IN RADIANS) TO DEGREES
2 C
3 C
4 C FUNCTION DEG(A)
5 C
6 C
7 C DEG = A*180.0/3.141592
8 C RETURN
9 C END

```

```

      A          4          7
      DEG        4          *7

```

### Function RAD

```
1 C FUNCTION 'RAD' CONVERTS ITS ARGUMENT (IN DEGREES) TO RADIANS
2 C
3 C
4 C     FUNCTION RAD(D)
5 C
6 C         RAD = D*3.141592/180.0
7 C
8 C         RETURN
9 C
END
```

WORK VARIABLES WORK

a 4 7  
rad 4 .77

## Decision Sawing (DECID)

```

1 C *DECID* SIMULATES THE DECISIONS OF A SKILLED HUMAN SAWER. AFTER
2 C THE LOG IS SAWED AT MID-LENGTH, BOARDS ARE CUT FROM THE 'BEST'
3 C FACE UNTIL A DROP IN GRADE OCCURS, A NEW 'BEST' FACE IS THEN FOUND
4 C AND THE PROCESS REPEATED FOR THE REST OF THE LOG
5 C
6 C
7 C IMPLICIT REAL (X,L)
8 C REAL FLY(26,2),FLY(26,2),PERC(5),BOARDU(2),
9 C DISTBB(4),SPF(4),RCUT(4),ANGLE(40),HEIGHT(40),KTLEN(40),
10 C R(4),DAT(4),
11 C INTEGER ID(22,2),NODEF(2),CPALLET(4),F,GDORR(58),COUEWS(58),
12 C GRCOR(4),GRBEST(1X(22),OLD,
13 C CORBIN(BOARD,L,D,M,T,LOGS,88,15TB),BOLBN,BOARDL,KNOTS,R,RCUT,BF
14 C APRFC,FAS,SEL,DNEC,TDC,TMB,
15 C /DECID/ CPALLET,FAS,SEL,DNEC,SPECIN,SBEST,STF
16 C /DECFC/ NODEFC,FLX,FLY,FLX,FUY,FLY,UFY,1D
17 C
18 C THIS PROGRAM ASSUMES INPUT FROM UNIT 5 AND OUTPUTS TO UNITS 6 AND 8
19 C
20 C GET LOG DIAMETER (IN.). LOG LENGTH (IN.), KERF SIZE (IN.), AND
21 C BOARD THICKNESS (IN.)
22 C
23 C 1 READ (5,116,END * 23) D
24 C     READ (5,116) L
25 C     READ (5,117) K
26 C     READ (5,118) T
27 C
28 C GET KNOT HALF-ANGLE (DEGREES) AND LOG HALF-TAPER (DEGREES)
29 C
30 C READ (5,116) KNOTS
31 C     READ (5,116) LOGS
32 C     READ (5,116) LOGS
33 C GET CORE DEFECT PARAMETERS: DIAMETER (IN.), LINEAR OFFSET (IN.),
34 C AND ANGULAR OFFSET (DEGREES)
35 C
36 C READ (5,108) CORDIA
37 C     READ (5,108) CORDIA
38 C     READ (5,108) COSET
39 C     READ (5,108) CANS
40 C
41 C GET PROFILE BOARD FOOT FOR EACH GRADE: NUMBER OF NODES
42 C
43 C READ (5,123) FAS,SEL,DNEC,TDC,TMB
44 C     READ (5,119) MAX
45 C
46 C GET DATE AND CALCULATE RADIUS OF CORE DEFECT
47 C CALL DATE (DATE)
48 C     CRADIUS = CORDIA/2.
49 C
50 C GET KNOT ANGLES, HEIGHTS, AND LENGTHS
51 C     READ (5,100) KTLEN(J)
52 C
53 DO 2 J=1,MAX
54     READ (5,102) ANGLE(J),HEIGHT(J)
55 2 CONTINUE
56     DO 3 J=1,MAX
57         READ (5,100) KTLEN(J)
58
59 3 CONTINUE
60
61 C CALCULATE LOG RADIUS AT TOP AND BOTTOM: TOTAL LOG CUBIC FEET
62 C
63 C PI = D/2,
64 C     PI = PI*ATAN(PAD(1,LOGS))
65 C     LNL = 1/3.*PI*141593*(RB*LOGS*L*D*H*T*KTLEN(J)*1/44.
66 C
67 C OUTPUT HEADER
68 C     WRITE (6,118) DAT
69 C     WRITE (6,119) L,D,LOGS,LNL
70 C     WRITE (6,118) MAX
71 C     DO 4 J=1,MAX
72 C     WRITE (6,101) ANGLE(J),HEIGHT(J),KTLEN(J)
73 C
74 C     READ (5,120) KNOTS
75 C     WRITE (6,115) CORDIA,COSET,CAMS

```

```

151    DO 9 J=1,25
152      FLX(J,1) = -9999.0
153      FLX(J,1) = -9999.0
154      FLY(J,1) = -9999.0
155      U(J,J,1) = -9999.0
156      CONTINUE
157      DO 10 J=1,22
158        ID(J,1) = 0
159        10  CONTINUE
160        HNDEF(J) = 0
161        C
162        PUT IN LINE DEFECTS
163        CALL LAMED (F,
164        CALL LAMIN (F,
165        C
166        PUT IN HNOTS
167        C
168        PUT IN LINE DEFECTS
169        NO 11 I=1,10
170          U(J,J,I) = HNDEF(J),HNFL(J),HNFL(J),KLEN(J),F
171        11  LAMININE
172        C
173        PUT IN LINE DEFECT
174        CALL COPED (CRADUS,COSSET,CANGS,F)
175        C
176        AND GRADE THE FACE
177        C
178        CALL GRADE (NGF)
179        C
180        SAVE GRADE OF FIRST SIDE OF BOARD
181        C
182        IF (I,EO,1) .NE. 0
183        SUBTRACT BOARD THICKNESS AND PROCESS THE INNER FACE
184        C
185        DISTB(F) = DISTBD(F)-T
186        C
187        12  CONTINUE
188        C
189        IGNORE THE LAST BOARD TAKEN
190        C
191        DISTB(F) = DISTBD(F)+T
192        C
193        DETERMINE BOARD SPARE AND VALUE BASED ON GRADES OF BOTH FACES
194        C
195        CALL FRTRT (OL1,NG1,TML,PFC,
196        C
197        SAVE BOARD SPARE AND SURFACE MEASURE
198        C
199        GTRT(F1) = NG1
200        SFR(F1) = NG2
201        C
202        SEE SEQUENCE OF FACES CUT AND THEIR GRADES
203        C
204        COORDS(HBD) = F
205        SFR(HBD) = NG
206        C
207        SEE IF LOG HAS BEEN SQUARED AT MID-LENGTH
208        C
209        IF (I50)HDF(.0) .GT. 7
210        C
211        SINCE THE LOG HAS BEEN SQUARED, BOARDS ARE CUT FROM THE 'BEST' FACE
212        C
213        UNTIL A DROP IN GRADE OCCURS. AFTER REGARDING THE 'BEST' FACE IS
214        C
215        A NEW 'BEST' FACE IS FOUND. NOTE THAT THE 'BEST' FACE IS INITIALLY
216        C
217        BETTER THAN FG5. FORCING A GRADE DROP ON THE FIRST BOARD
218        C
219        IF ('GRCOM(F1).LT.GRCOM(F2)) GO TO 13
220        C
221        OTHERWISE, SAVE BOARD GRADE AND SURFACE MEASURE AND CONTINUE CUTTING
222        C
223        ON THE SAME FACE
224        C
225        GRCOM(F) = GRCOM(F)
226        SIN(F) = SFR(F)
227        GO TO 8
228        C
229        IF (I,EO,-1) F = 3
230          R(J,J,1) = 0
231          DECID
232          DECID
233          DECID
234          DECID
235          DECID
236          DECID
237          DECID
238          DECID
239          DECID
240          DECID
241          DECID
242          DECID
243          DECID
244          DECID
245          DECID
246          DECID
247          DECID
248          DECID
249          DECID
250          DECID
251          DECID
252          DECID
253          DECID
254          DECID
255          DECID
256          DECID
257          DECID
258          DECID
259          DECID
260          DECID
261          DECID
262          DECID
263          DECID
264          DECID
265          DECID
266          DECID
267          DECID
268          DECID
269          DECID
270          DECID
271          DECID
272          DECID
273          DECID
274          DECID
275          DECID
276          DECID
277          DECID
278          DECID
279          DECID
280          DECID
281          DECID
282          DECID
283          DECID
284          DECID
285          DECID
286          DECID
287          DECID
288          DECID
289          DECID
290          DECID
291          DECID
292          DECID
293          DECID
294          DECID
295          DECID
296          DECID
297          DECID
298          DECID
299          DECID
300          DECID
301          DECID
302          DECID
303          DECID
304          DECID
305          DECID
306          DECID
307          DECID
308          DECID
309          DECID
310          DECID
311          DECID
312          DECID
313          DECID
314          DECID
315          DECID
316          DECID
317          DECID
318          DECID
319          DECID
320          DECID
321          DECID
322          DECID
323          DECID
324          DECID
325          DECID
326          DECID
327          DECID
328          DECID
329          DECID
330          DECID
331          DECID
332          DECID
333          DECID
334          DECID
335          DECID
336          DECID
337          DECID
338          DECID
339          DECID
340          DECID
341          DECID
342          DECID
343          DECID
344          DECID
345          DECID
346          DECID
347          DECID
348          DECID
349          DECID
350          DECID
351          DECID
352          DECID
353          DECID
354          DECID
355          DECID
356          DECID
357          DECID
358          DECID
359          DECID
360          DECID
361          DECID
362          DECID
363          DECID
364          DECID
365          DECID
366          DECID
367          DECID
368          DECID
369          DECID
370          DECID
371          DECID
372          DECID
373          DECID
374          DECID
375          DECID
376          DECID
377          DECID
378          DECID
379          DECID
380          DECID
381          DECID
382          DECID
383          DECID
384          DECID
385          DECID
386          DECID
387          DECID
388          DECID
389          DECID
390          DECID
391          DECID
392          DECID
393          DECID
394          DECID
395          DECID
396          DECID
397          DECID
398          DECID
399          DECID
400          DECID
401          DECID
402          DECID
403          DECID
404          DECID
405          DECID
406          DECID
407          DECID
408          DECID
409          DECID
410          DECID
411          DECID
412          DECID
413          DECID
414          DECID
415          DECID
416          DECID
417          DECID
418          DECID
419          DECID
420          DECID
421          DECID
422          DECID
423          DECID
424          DECID
425          DECID
426          DECID
427          DECID
428          DECID
429          DECID
430          DECID
431          DECID
432          DECID
433          DECID
434          DECID
435          DECID
436          DECID
437          DECID
438          DECID
439          DECID
440          DECID
441          DECID
442          DECID
443          DECID
444          DECID
445          DECID
446          DECID
447          DECID
448          DECID
449          DECID
450          DECID
451          DECID
452          DECID
453          DECID
454          DECID
455          DECID
456          DECID
457          DECID
458          DECID
459          DECID
460          DECID
461          DECID
462          DECID
463          DECID
464          DECID
465          DECID
466          DECID
467          DECID
468          DECID
469          DECID
470          DECID
471          DECID
472          DECID
473          DECID
474          DECID
475          DECID
476          DECID
477          DECID
478          DECID
479          DECID
480          DECID
481          DECID
482          DECID
483          DECID
484          DECID
485          DECID
486          DECID
487          DECID
488          DECID
489          DECID
490          DECID
491          DECID
492          DECID
493          DECID
494          DECID
495          DECID
496          DECID
497          DECID
498          DECID
499          DECID
500          DECID
501          DECID
502          DECID
503          DECID
504          DECID
505          DECID
506          DECID
507          DECID
508          DECID
509          DECID
510          DECID
511          DECID
512          DECID
513          DECID
514          DECID
515          DECID
516          DECID
517          DECID
518          DECID
519          DECID
520          DECID
521          DECID
522          DECID
523          DECID
524          DECID
525          DECID
526          DECID
527          DECID
528          DECID
529          DECID
530          DECID
531          DECID
532          DECID
533          DECID
534          DECID
535          DECID
536          DECID
537          DECID
538          DECID
539          DECID
540          DECID
541          DECID
542          DECID
543          DECID
544          DECID
545          DECID
546          DECID
547          DECID
548          DECID
549          DECID
550          DECID
551          DECID
552          DECID
553          DECID
554          DECID
555          DECID
556          DECID
557          DECID
558          DECID
559          DECID
560          DECID
561          DECID
562          DECID
563          DECID
564          DECID
565          DECID
566          DECID
567          DECID
568          DECID
569          DECID
570          DECID
571          DECID
572          DECID
573          DECID
574          DECID
575          DECID
576          DECID
577          DECID
578          DECID
579          DECID
580          DECID
581          DECID
582          DECID
583          DECID
584          DECID
585          DECID
586          DECID
587          DECID
588          DECID
589          DECID
590          DECID
591          DECID
592          DECID
593          DECID
594          DECID
595          DECID
596          DECID
597          DECID
598          DECID
599          DECID
600          DECID
601          DECID
602          DECID
603          DECID
604          DECID
605          DECID
606          DECID
607          DECID
608          DECID
609          DECID
610          DECID
611          DECID
612          DECID
613          DECID
614          DECID
615          DECID
616          DECID
617          DECID
618          DECID
619          DECID
620          DECID
621          DECID
622          DECID
623          DECID
624          DECID
625          DECID
626          DECID
627          DECID
628          DECID
629          DECID
630          DECID
631          DECID
632          DECID
633          DECID
634          DECID
635          DECID
636          DECID
637          DECID
638          DECID
639          DECID
640          DECID
641          DECID
642          DECID
643          DECID
644          DECID
645          DECID
646          DECID
647          DECID
648          DECID
649          DECID
650          DECID
651          DECID
652          DECID
653          DECID
654          DECID
655          DECID
656          DECID
657          DECID
658          DECID
659          DECID
660          DECID
661          DECID
662          DECID
663          DECID
664          DECID
665          DECID
666          DECID
667          DECID
668          DECID
669          DECID
670          DECID
671          DECID
672          DECID
673          DECID
674          DECID
675          DECID
676          DECID
677          DECID
678          DECID
679          DECID
680          DECID
681          DECID
682          DECID
683          DECID
684          DECID
685          DECID
686          DECID
687          DECID
688          DECID
689          DECID
690          DECID
691          DECID
692          DECID
693          DECID
694          DECID
695          DECID
696          DECID
697          DECID
698          DECID
699          DECID
700          DECID
701          DECID
702          DECID
703          DECID
704          DECID
705          DECID
706          DECID
707          DECID
708          DECID
709          DECID
710          DECID
711          DECID
712          DECID
713          DECID
714          DECID
715          DECID
716          DECID
717          DECID
718          DECID
719          DECID
720          DECID
721          DECID
722          DECID
723          DECID
724          DECID
725          DECID
726          DECID
727          DECID
728          DECID
729          DECID
730          DECID
731          DECID
732          DECID
733          DECID
734          DECID
735          DECID
736          DECID
737          DECID
738          DECID
739          DECID
740          DECID
741          DECID
742          DECID
743          DECID
744          DECID
745          DECID
746          DECID
747          DECID
748          DECID
749          DECID
750          DECID
751          DECID
752          DECID
753          DECID
754          DECID
755          DECID
756          DECID
757          DECID
758          DECID
759          DECID
760          DECID
761          DECID
762          DECID
763          DECID
764          DECID
765          DECID
766          DECID
767          DECID
768          DECID
769          DECID
770          DECID
771          DECID
772          DECID
773          DECID
774          DECID
775          DECID
776          DECID
777          DECID
778          DECID
779          DECID
780          DECID
781          DECID
782          DECID
783          DECID
784          DECID
785          DECID
786          DECID
787          DECID
788          DECID
789          DECID
790          DECID
791          DECID
792          DECID
793          DECID
794          DECID
795          DECID
796          DECID
797          DECID
798          DECID
799          DECID
800          DECID
801          DECID
802          DECID
803          DECID
804          DECID
805          DECID
806          DECID
807          DECID
808          DECID
809          DECID
810          DECID
811          DECID
812          DECID
813          DECID
814          DECID
815          DECID
816          DECID
817          DECID
818          DECID
819          DECID
820          DECID
821          DECID
822          DECID
823          DECID
824          DECID
825          DECID
826          DECID
827          DECID
828          DECID
829          DECID
830          DECID
831          DECID
832          DECID
833          DECID
834          DECID
835          DECID
836          DECID
837          DECID
838          DECID
839          DECID
840          DECID
841          DECID
842          DECID
843          DECID
844          DECID
845          DECID
846          DECID
847          DECID
848          DECID
849          DECID
850          DECID
851          DECID
852          DECID
853          DECID
854          DECID
855          DECID
856          DECID
857          DECID
858          DECID
859          DECID
860          DECID
861          DECID
862          DECID
863          DECID
864          DECID
865          DECID
866          DECID
867          DECID
868          DECID
869          DECID
870          DECID
871          DECID
872          DECID
873          DECID
874          DECID
875          DECID
876          DECID
877          DECID
878          DECID
879          DECID
880          DECID
881          DECID
882          DECID
883          DECID
884          DECID
885          DECID
886          DECID
887          DECID
888          DECID
889          DECID
890          DECID
891          DECID
892          DECID
893          DECID
894          DECID
895          DECID
896          DECID
897          DECID
898          DECID
899          DECID
900          DECID
901          DECID
902          DECID
903          DECID
904          DECID
905          DECID
906          DECID
907          DECID
908          DECID
909          DECID
910          DECID
911          DECID
912          DECID
913          DECID
914          DECID
915          DECID
916          DECID
917          DECID
918          DECID
919          DECID
920          DECID
921          DECID
922          DECID
923          DECID
924          DECID
925          DECID
926          DECID
927          DECID
928          DECID
929          DECID
930          DECID
931          DECID
932          DECID
933          DECID
934          DECID
935          DECID
936          DECID
937          DECID
938          DECID
939          DECID
940          DECID
941          DECID
942          DECID
943          DECID
944          DECID
945          DECID
946          DECID
947          DECID
948          DECID
949          DECID
950          DECID
951          DECID
952          DECID
953          DECID
954          DECID
955          DECID
956          DECID
957          DECID
958          DECID
959          DECID
960          DECID
961          DECID
962          DECID
963          DECID
964          DECID
965          DECID
966          DECID
967          DECID
968          DECID
969          DECID
970          DECID
971          DECID
972          DECID
973          DECID
974          DECID
975          DECID
976          DECID
977          DECID
978          DECID
979          DECID
980          DECID
981          DECID
982          DECID
983          DECID
984          DECID
985          DECID
986          DECID
987          DECID
988          DECID
989          DECID
990          DECID
991          DECID
992          DECID
993          DECID
994          DECID
995          DECID
996          DECID
997          DECID
998          DECID
999          DECID
1000         DECID

```

```

307 WRITE (6,184) (PERC(J),J=1,5),TVAL
308 WRITE (6,110)
309 WRITE (6,187) (COLUMNS(J),J=1,NBD)
310 WRITE (6,187) (GNDROW(J),J=1,NBD)
311 WRITE (6,205) (PERC(J),J=1,5)
312 WRITE (6,284),TVAL,TOTS
313 C SAME HIGH, LOW, AND AVERAGE YIELDS FOR THIS LOG
314 C
315 C AVTOTS = AVTOT*TOTS
316 AVG5 = AVG5*TVAL
317 IF (TVAL.GT.HIGH5) HITOT = TOTS
318 IF (TVAL.LT.SMALL5) SHITOT = TOTS
319 IF (TVAL.GT.HIGH5) HIGHS = TVAL
320 IF (TVAL.LT.SMALL5) SMALLS = TVAL
321 IF (TVAL.GT.HIGH5) SMALLS = TVAL
322 C ROTATE LOG BY 15 DEGREES AND REPROCESS
323 C
324 C DO 21 J=1,MAX
325 ANGLE(J) = ATN(D(ANGLE(J))+15.0,360.0)
326 21 CONTINUE
327 C
328 C RANGS = ATN(D(RANGS)+15.,360.)
329 22 CONTINUE
330 C
331 C PRINT HIGH, LOW, AND AVERAGE YIELDS FOR THIS LOG
332 C
333 AVTOTS = AVTOT/12.
334 AVG5 = AVG5/12.
335 WRITE (6,185) HIGH5,AVG5,SMALLS
336 WRITE (6,186) HITOT,AVTOTS,SHITOT
337 C
338 C LOG COMPLETELY PROCESSED. READ PARAMETERS FOR NEXT LOG
339 C
340 GO TO 1
341 C
342 C 23 STOP
343 C
344 C FORMATS FOR DEVICE 6 (PRINTER)
345 C
346 100 FORMAT (F6.3)
347 101 FORMAT (3.12X,F6.2)
348 102 FORMAT (2P6.2)
349 103 FORMAT (' KNOT ANGLES: ',1SF7.2,'/13X,1SF7.2)
350 104 FORMAT (' OF AVAILABLE SURFACE MEASURE IN EACH GRADE: ',1SF5.1,
     'LOG VALUE IS ',F6.2)
351 105 FORMAT (' $ YIELD: HIGH ',F6.2,' AVERAGE ',F6.2,' LOW ',F6.2)
352 106 FORMAT (' % YIELD: HIGH $ FACE ',F6.2,' AVERAGE ',F6.2,
     '- LOW $ FACE ',F6.2)
353 107 FORMAT (' ',A6,I3)
354 108 FORMAT (' KNOTS IN THIS LOG. THEY ARE',/
     'AK, ANGLE HEIGHT LENGTH')
355 109 FORMAT (12)
356 110 FORMAT (' SEQUENCE OF LOG FACES CUT AND THEIR BOARD GRADES')
357 111 FORMAT ('2./..',ROTATION 15.,14)
358 112 FORMAT (' TOTAL SURFACE MEASURE 15.',F6.2)
359 113 FORMAT (' TOTAL SURFACE MEASURE PER GRADE 15.',SF7.2)
360 114 FORMAT (' SURFACE MEASURE FROM ZERO DEGREES 15.',F6.2)
361 115 FORMAT (' VERTICALLY CUTTING SAW, KNOT TAPER',F6.2)
362 116 FORMAT (' CORE DEFECT RADIUS ',F6.3,' LINEAR OFFSET ',F6.3,
     'ANGULAR OFFSET ',F6.3)
363 117 FORMAT (F5.1)
364 118 FORMAT (F5.3)
365 119 FORMAT (' DECISION SAWING METHOD ',AB)
366 120 FORMAT (' LOG PARAMETERS: LENGTH ',F6.1,', DIAM ',F6.1,', TAPER',
     ', DEGREES, LOG VOLUME: ',F6.2,', CUBIC FEET')
367 121 FORMAT (' ANGLES MEASURED FROM ZERO DEGREES EAST FOR A '
     ', VERTICALLY CUTTING SAW, KNOT TAPER',F6.2)
368 122 FORMAT (' CUTTING PARAME TERS: KERF ',F6.4,', BOARD THICKNESS'
     ',F5.2)
369 123 FORMAT (' PRICES FPP BOARD FOOT: ',SF10.4)
370 C
371 C FORMATS FOR DEVICE 8 (STORAGE MEDIUM)
372 C
373 200 FORMAT (13SF5.1,F6.2,F6.3,F4.1,F6.2,F5.3,F6.3)
374 201 FORMAT (31SF7.2)
375 202 FORMAT (SF7.4)
376 203 FORMAT (11)
377 204 FORMAT (SF6.2)
378 C
379 C
380 205 FORMAT (10SF6.2)
381 206 FORMAT (31SF7.2)
382 207 FORMAT (SF7.4)
383 208 FORMAT (11)
384 209 FORMAT (SF6.2)
385 210 FORMAT (SF6.2)
386 211 FORMAT (SF6.2)
387 212 FORMAT (SF6.2)
388 213 FORMAT (SF6.2)
389 214 FORMAT (SF6.2)
390 215 FORMAT (SF6.2)
391 216 FORMAT (SF6.2)
392 217 FORMAT (SF6.2)
393 218 FORMAT (SF6.2)
394 219 FORMAT (SF6.2)
395 220 FORMAT (SF6.2)
396 221 FORMAT (SF6.2)
397 222 FORMAT (SF6.2)
398 223 FORMAT (SF6.2)
399 224 FORMAT (SF6.2)
400 225 FORMAT (SF6.2)
401 226 FORMAT (SF6.2)
402 227 FORMAT (SF6.2)
403 228 FORMAT (SF6.2)
404 229 FORMAT (SF6.2)
405 230 FORMAT (SF6.2)
406 231 FORMAT (SF6.2)
407 232 FORMAT (SF6.2)
408 233 FORMAT (SF6.2)
409 234 FORMAT (SF6.2)
410 235 FORMAT (SF6.2)
411 236 FORMAT (SF6.2)
412 237 FORMAT (SF6.2)
413 238 FORMAT (SF6.2)
414 239 FORMAT (SF6.2)
415 240 FORMAT (SF6.2)
416 241 FORMAT (SF6.2)
417 242 FORMAT (SF6.2)
418 243 FORMAT (SF6.2)
419 244 FORMAT (SF6.2)
420 245 FORMAT (SF6.2)
421 246 FORMAT (SF6.2)
422 247 FORMAT (SF6.2)
423 248 FORMAT (SF6.2)
424 249 FORMAT (SF6.2)
425 250 FORMAT (SF6.2)
426 251 FORMAT (SF6.2)
427 252 FORMAT (SF6.2)
428 253 FORMAT (SF6.2)
429 254 FORMAT (SF6.2)
430 255 FORMAT (SF6.2)
431 256 FORMAT (SF6.2)
432 257 FORMAT (SF6.2)
433 258 FORMAT (SF6.2)
434 259 FORMAT (SF6.2)
435 260 FORMAT (SF6.2)
436 261 FORMAT (SF6.2)
437 262 FORMAT (SF6.2)
438 263 FORMAT (SF6.2)
439 264 FORMAT (SF6.2)
440 265 FORMAT (SF6.2)
441 266 FORMAT (SF6.2)
442 267 FORMAT (SF6.2)
443 268 FORMAT (SF6.2)
444 269 FORMAT (SF6.2)
445 270 FORMAT (SF6.2)
446 271 FORMAT (SF6.2)
447 272 FORMAT (SF6.2)
448 273 FORMAT (SF6.2)
449 274 FORMAT (SF6.2)
450 275 FORMAT (SF6.2)
451 276 FORMAT (SF6.2)
452 277 FORMAT (SF6.2)
453 278 FORMAT (SF6.2)
454 279 FORMAT (SF6.2)
455 280 FORMAT (SF6.2)
456 281 FORMAT (SF6.2)
457 282 FORMAT (SF6.2)
458 283 FORMAT (SF6.2)
459 284 FORMAT (SF6.2)
460 285 FORMAT (SF6.2)
461 286 FORMAT (SF6.2)
462 287 FORMAT (SF6.2)
463 288 FORMAT (SF6.2)
464 289 FORMAT (SF6.2)
465 290 FORMAT (SF6.2)
466 291 FORMAT (SF6.2)
467 292 FORMAT (SF6.2)
468 293 FORMAT (SF6.2)
469 294 FORMAT (SF6.2)
470 295 FORMAT (SF6.2)
471 296 FORMAT (SF6.2)
472 297 FORMAT (SF6.2)
473 298 FORMAT (SF6.2)
474 299 FORMAT (SF6.2)
475 300 FORMAT (SF6.2)
476 301 FORMAT (SF6.2)
477 302 FORMAT (SF6.2)
478 303 FORMAT (SF6.2)
479 304 FORMAT (SF6.2)
480 305 FORMAT (SF6.2)
481 306 FORMAT (SF6.2)
482 307 FORMAT (SF6.2)
483 308 FORMAT (SF6.2)
484 309 FORMAT (SF6.2)
485 310 FORMAT (SF6.2)
486 311 FORMAT (SF6.2)
487 312 FORMAT (SF6.2)
488 313 FORMAT (SF6.2)
489 314 FORMAT (SF6.2)
490 315 FORMAT (SF6.2)
491 316 FORMAT (SF6.2)
492 317 FORMAT (SF6.2)
493 318 FORMAT (SF6.2)
494 319 FORMAT (SF6.2)
495 320 FORMAT (SF6.2)
496 321 FORMAT (SF6.2)
497 322 FORMAT (SF6.2)
498 323 FORMAT (SF6.2)
499 324 FORMAT (SF6.2)
500 325 FORMAT (SF6.2)
501 326 FORMAT (SF6.2)
502 327 FORMAT (SF6.2)
503 328 FORMAT (SF6.2)
504 329 FORMAT (SF6.2)
505 330 FORMAT (SF6.2)
506 331 FORMAT (SF6.2)
507 332 FORMAT (SF6.2)
508 333 FORMAT (SF6.2)
509 334 FORMAT (SF6.2)
510 335 FORMAT (SF6.2)
511 336 FORMAT (SF6.2)

```

			TVAL	*125	195	307	312	317	318	319	320	321
			TYPE	13	*43	77	82					
			WEIGHT	164	254							
BDLEN	13	*124	280	273								
BF	13	*124	280	273								
BOARDW	8	13										
CANCS	*39	75	78	174	264	*328						
CHPLT	11	13	*121									
CDFSET	*38	75	.78	174	264							
COPRIA	*37	49										
CONFID	17	1	76.4									
CPN165	11	*204	309									
CPN165	*49	75	.78	174	264							
D	11	*24	62	69	.78							
DAT	8	48	68									
DATE	48											
DECIDE	288											
DISTBD	3	13	*117	*186	*191	*278						
EXTS5	*110	111	112	113	118							
F	11	*131	*135	*136	141	164	169	174	186	191	199	200
F	204	216	221	*222	*228	*238	*242	*253	*259	254	254	259
FAS	264	272	273	277	278	*283	*284	288				
FLX	9	13	*152	*239								
FLY	8	13	*154	*241								
FUY	8	13	*153	*238								
FUY	6	13	*155	*249								
GDBEST	11	13	*126	216	*221							
GDORAR	11	*285	310									
GPCE	178	268										
GDC	11	13	*119	*199	216	221	*272					
HEIGHT	5	*54	72	80	169	259						
HIGHS	*87	318	*320	335								
HITOT	*50	*318	336									
I	1	182										
ID	11	13	*158	*244								
II	*221											
INT	189											
INV	*97	293	298									
IPOTAT	*293	294	298									
ISOURP	*181	141	289	250								
J	11	*53	*54	*56	*57	*71	72	79	80	81	*116	117
J	119	120	121	*128	129	*151	152	153	154	155	*157	159
J	119	120	121	*128	129	*151	152	153	154	155	*157	159
J	*162	165	*178	259	295	*309	*361	362	365	367	369	310
J	311	*315	326	340	241	*243	244					
JJ	*221	238	239	246	189	198	199	110	112	113	118	278
K	17	226	76	78								
KEPFID	141	258										
KHDS	13	421	74	78								
KHDTD	163	239										
KTEN	8	*17	72	79	169	259						
L	13	625	63	64	69	78						
LOGS	13	*432	63	69	76							
LVL	664	69	78	302	304							
MAX	*44	53	56	78	71	78	79	90	81	168	258	305
MED	325											
NEPDS	*127	*142	264	285	382	310						
NODEFC	11	13	*168	*246								
NPIC	173	182	195	199	265	268	272					
OLD	11	*182	195									
ONEC	13	*443	77	82								
PERC	2	*129	195	295	296	299	*392	387	311			
PRICED	195											
Q	8	13	*112	*113	*114	*115						
RAN	63	463	64	186	117							
RB	13	63	64	186	117							
RCUT	8	13	*118	*227								
R1	362	63	64									
R7BLT	*188	189	110	*111	112	113						
SEL	13	*43	77	82								
SPALLS	*68	319	*321	335								
SPRST	12	*123	*222	222	*223							
SPPTR	8	12	*120	*200	222							
T	17	*27	76	78	168	169	110	112	113	118	186	191
TAN	63	*43	77	82								
TMR8	13	*43	336	*394	365	312	316	318	319			
TOTS	*295	297	300									

## Subroutine KERF

```

1 C SUBROUTINE 'KERF' CUTS A BOARD FROM THE LOG. THE BOARD WIDTH IS
2 C DEFINED AT THE CENTER, AND THE BOARD LENGTH FROM THE BOTTOM
3 C OF THE LOG
4 C
5 C SUBROUTINE KERF (N, ISOUR, F, IREGD)
6 C
7 C IMPLICIT REAL (K,L)
8 C INTEGER F,FRT,FLT,LD(22,2),CMPLT(40),INDEC(2),SPRCDM(4)
9 C REAL FLX(26,2),FLY(26,2),FLX(26,2),FLY(26,2),BOARDW(2),LT(2),
10 C DISTBD(4),RCUT(4),SF(4),R(4)
11 C COMMON /BOARD/ L,D,K,T,LOGS,RB,DISTBD,BOLEN,BOARDL,KNOTS,R,RCUT,SF
12 C
13 C *DECID CMPLT GDBEST GRCON,SPRCM,SPF
14 C /DEFEC/ NODEFC,FLX,FLY,FLY,PUT,LD
15 C
16 C ALL BOARDS MUST BE AT LEAST .48 INCHES LONG, 2.5 INCHES WIDE AT
17 C THE TOP, AND 3 INCHES WIDE AT THE CENTER. IN ADDITION, THE TOTAL
18 C AMOUNT OF LOGE FROM THE CENTER TO THE TOP IS LIMITED TO 4 INCHES.
19 C THE BOARD IS CUT BACK IN ONE FOOT INCREMENTS IF THE WIDTH OR LOGE
20 C TESTS FAIL. IF THE LENGTH TEST FAILS, ANOTHER BOARD IS CUT FROM
21 C THIS FACE.
22 C
23 C RI = D/2.
24 C
25 C FIND THE TWO ADJACENT FACES
26 C
27 C FLFT = F-1
28 C IF (FLFT.EQ.0) FLFT = 4
29 C FRT = F+1
30 C IF (FRT.EQ.5) FRT = 1
31 C
32 C FIND OUTER FACE OF BOARD
33 C
34 C 1 RCUT(F) = RUT(F)*T**4
35 C DISTBD(F) = RB-RCUT(F)
36 C BOLEN = L
37 C
38 C IF THIS FACE OF THE LOG IS COMPLETELY CUT. EXIT
39 C
40 C IF ((PCLT(F).GT.PRF)) GO TO 6
41 C POSLEN = PCUT(F)*TAN(RAD(.90.-LOG$))
42 C CHECK FOR BOARDS SHORTER THAN THE LOG DUE TO LOG TAPER
43 C
44 C IF (POSLEN.LT.BOLEN) BOLEN = POSLEN
45 C
46 C BOARDS MUST BE AT LEAST 4 FEET LONG
47 C
48 C 2 IF (BOLEN.LT.48.) GO TO 5
49 C
50 C DETERMINE HALF-WIDTHS AT MID-LENGTH OF BOARD. SINCE THE BOARD IS
51 C NOT NECESSARILY SYMMETRICAL ABOUT A LINE FROM THE CENTER OF THE
52 C LOG PERPENDICULAR TO THE BOARD FACE, THE WIDTH OF EACH SIDE MUST
53 C BE DETERMINED SEPARATELY
54 C
55 C RM = R1+(L-BOLEN/2.)*TAN(RAD(.90.-LOG$))
56 C IF (DISTBD(F).GT.RM) GO TO 4
57 C IF (DISTBD(F).LT.RM) GO TO 3
58 C BOREDU(1) = SORT(RHM**2-DISTBD(F)**2)
59 C BOREDU(2) = BOARDW(2)
60 C
61 C THE BOARD WIDTH MAY BE LIMITED BY THE ADJACENT FACES
62 C
63 C DSTLFL = DISTBD(F1)-K
64 C DSTRCT = DISTBD(F1)-K
65 C BOARDW(1) = AMIN(DSTLFL,BOARDW(1))
66 C BOARDW(2) = AMIN(DSTRCT,BOARDW(2))
67 C
68 C IS LOG SQUARED AT MID-LENGTH?
69 C
70 C IF ((BOARDW(2).EQ.DSTRCT)) ISQMK = 0
71 C
72 C BOARD MUST BE AT LEAST 3 INCHES WIDE AT MID-LENGTH
73 C
74 C IF ((BOARDW(1)+BOREDU(2)).LT.3.) GO TO 4
75 C

```

```

76 C DETERMINE HALF-WIDTHS AT TOP OF BOARD
77 C RT = R1+(L-BOLEN)*TAN(RAD(.90.-LOG$))
78 C IF (DISTBD(F1).LT.RT) GO TO 4
79 C UT(1) = SORT(RHM**2-DISTBD(F)**2)
80 C UT(1) = AMIN(DSTLFL,UT(1))
81 C UT(2) = AMIN(DSTRCT,UT(2))
82 C
83 C BOARD MUST BE AT LEAST 2.5 INCHES WIDE AT THE TOP
84 C
85 C IF (UT(1)+UT(2)).LT.2.5) GO TO 4
86 C
87 C LIMIT THE TOTAL LOGE TO AVOID GENERATING MORE DEFECTS THAN THE
88 C GRADING PROGRAM CAN HANDLE. TOTAL WIDTH OF LOGE MUST BE LESS
89 C THAN 4 INCHES
90 C
91 C UPN = BOARDW(1)+BOARDW(2)-UT(1)-UT(2)
92 C
93 C IF (UPN.GT.4.) GO TO 4
94 C
95 C CONVERT BOARD HALF-WIDTHS AND LENGTH TO EVEN QUARTER INCHES
96 C
97 C DO 3 J=1,2
98 C 3 CONTINUE
99 C BOARDW(J) = INT(BOARDW(J)*4.)/4.
100 C
101 C DEFINE BOARD EDGES IN QUARTER INCH UNITS
102 C
103 C 181 = INT(6IN EN*4.)/4.
104 C
105 C FLX(26,1) = BDLEN*4.
106 C FUX(26,1) = 0.0
107 C FLY(26,1) = (BOARDW(1)+BOARDW(2))*4.
108 C FUY(26,1) = 0.0
109 C
110 C CALCULATE UNROUNDED BOARD FEET
111 C
112 C BF = BDLEN*(BOARDW(1)+BOARDW(2))*T/144.
113 C PFTURN
114 C CUT BACK BOARD BY ONE FOOT
115 C
116 C 4 BDLEN = BDLEN-12.
117 C
118 C GO TO 2
119 C
120 C IF THE LOG IS BEING REGRADED, ANOTHER BOARD SHOULD NOT BE CUT
121 C FROM THIS FACE.
122 C
123 C 5 IF ((REGRD.EQ.1)) GO TO 6
124 C
125 C GO TO 1
126 C
127 C FACE HAS BEEN COMPLETELY CUT. DO NOT CUT ANY MORE BOARDS FROM IT
128 C
129 C 6 CPLET(F) = 1
130 C GRDCMF(F) = ?
131 C
132 C IGNORE THE CUT JUST ATTEMPTED
133 C RCUT(F) = RCUT(F)-T-K
134 C DISTBD(F) = RB-RCUT(F)-T
135 C SF(F) = 0.0
136 C RETURN1
137 C
138 C
139 C
140 C
141 C
142 C
143 C
144 C
145 C
146 C
147 C
148 C
149 C
150 C
151 C
152 C
153 C
154 C
155 C
156 C
157 C
158 C
159 C
160 C
161 C
162 C
163 C
164 C
165 C
166 C
167 C
168 C
169 C
170 C
171 C
172 C
173 C
174 C
175 C

```

\*\*\* STATEMENT NUMBERS \*\*\*

1	*35	124
2	*50	118
3	*98	*100

## Subroutine WANE

```

      4   59    75    79    87    94    *117
      6   41    *123   *123
      6     SUBROUTINE 'WANE' LOCATES WANE DEFECTS ON THE BOARD
      7
      8         SUBROUTINE WANE ( F )
      9
      10        THE BOARD EDGE HAS A NEW WANE DEFECT IF THE HALF-WIDTH ON THAT SIDE
      11        DROPS BY 1/4 INCH. SINCE THE BOARD IS NOT NECESSARILY SYMMETRICAL
      12        THE WIDTH OF THE WANE DEFECTS MUST BE CALCULATED SEPARATELY FOR EACH
      13        SIDE
      14
      15        IMPLICIT REAL ( K,L )
      16        INTEGER I,D(22,2),NODEFC(2),F
      17        REAL BOARDW(2),DISTBD(4),FLX(26,2),FLX(26,2),FLX(26,2),
      18           FCUT(4),R(4)
      19        COMMON /BOARD/ L,D,K,T,LOGS,RB,DISTBD,BDLEN,BORDU,KNOTS,R,RCUT,BF
      20           /DEFEC/ NODEFC,FLX,FLY,FLX,FLY,FLY,FLY,FLY,FLY,FLY
      21
      22        LDH = BDLEN/2.
      23        V = 8.
      24
      25        FIND THE PLACE ALONG THE BOARD WHERE THE HALF-WIDTH HAS DECREASED
      26        BY 1/4 INCH. THE DEFECT ENDS HERE AND EXTENDS FROM THE END OF THE
      27        LAST WANE DEFECT (OR FROM THE MIDDLE OF THE BOARD).
      28
      29        1. V = V*0.25
      30        SOR = SORT((CON2+(BOARDW(2)-V)*K*2)
      31          H = INT((M1)((RB-SOR)**4./CON1.BDLEN**2))
      32          NODEFC(1) = NODEFC(1)+1
      33          M = NODEFC(1)
      34
      35        LIMIT NUMBER OF DEFECTS TO MAXIMUM ALLOWED BY GRADING PROGRAM (22)
      36
      37        IF (M.GT.22) GO TO 2
      38
      39        SAVE WANE COORDINATES FOR GRADING PROGRAM
      40
      41        FLY(M,1) = 0.
      42        FLX(M,1) = V*4.+1.
      43        FUX(M,1) = H
      44        FLXM(M,1) = OLDH
      45        ID(M,1) = 5
      46
      47        WHEN WANE EXTENDS TO END OF BOARD, THIS EDGE IS DONE
      48
      49        2. IF (H.GE.BDLEN/2.) GO TO 3
      50
      51        THE NEXT WANE DEFECT BEGINS WHERE THIS ONE ENDS
      52
      53        3. IF ((RB-SORT((CON2+BDLEN/2))-CON1.GE.BDLEN)) RETURN
      54        OLDH = H
      55        GO TO 1
      56
      57        REPEAT FOR OTHER EDGE OF BOARD
      58
      59        3. IF ((RB-SORT((CON2+BDLEN/2))-CON1.GE.BDLEN)) RETURN
      60        OLDH = BDLEN/2.
      61        V = 8.
      62
      63        FIND END OF WANE DEFECT
      64
      65        4. V = V*0.25
      66        SOR = SORT((CON2+(BOARDW(1)-V)*K*2)
      67          H = INT((M1)((RB-SOR)**4./CON1.BDLEN**2))
      68          NODEFC(1) = NODEFC(1)+1
      69          M = NODEFC(1)
      70
      71        LIMIT NUMBER OF DEFECTS TO MAXIMUM ALLOWED BY GRADING PROGRAM (22)
      72
      73        IF (M.GT.22) GO TO 5
      74
      75

```

```

38
76 C SAVE LOGN COORDINATES FOR GRADING PROGRAM
77 C FUY(H,1) = (BOARDW(1)+BOARDW(2))/2, *4.
78 C FLY(H,1) = (BOARDU(1)+BOARDU(2))/2, *4, -1.
79 C FLUX(H,1) = H
80 C FLIX(H,1) = N
81 C ID(H,1) = 5
82 C WHEN THE DEFECT EXTENDS TO THE END OF THE BOARD, EXIT
83 C
84 C IF (H.GE.BDLENH) RETURN
85 C
86 C THE NEXT LOGN DEFECT BEGINS WHERE THIS ONE ENDS
87 C
88 C OLDH = H
89 C GO TO 4
90 C RETURN
91 C
92 C
93 C

```

	STATEMENT NUMBERS	*** VARTABLES ***
1	*70 55	
2	*39 *50	
3	*21 50 *60	
4	*6 91	
5	*74 *86	
AMINI	32 68	
BDLEN	15 21 22	
BF	15 21 31 60 68 69 67 78 79	
BORRN	13 15 21 32 60 68 67 78 79	
CON1	*19 21 31 60 67	
CON2	*20 21 31 60 67	
D	15 20	
DISTBD	13 15 20	
F	4 12 20	
FLX	13 15 *45	
FLY	13 15 *42	
FUX	13 15 *44	
FUY	13 15 *13 *79	
H	*32 44 50 68 80 86 90	
ID	12 15 *46 *82	
INT	32 68	
K	15	
KNOTS	15	
L	15	
LOGS	15 19	
H	*34 38 42 43 44 45 46 *76 74 78 79 80	
MDDEF	12 15 *53 34 *69 78	
OLDH	*77 45 *54 *61 81 *86	
R	13 15	
RD	19	
RB	15 21 32 68 68	
RCUT	12 15 *67 68	
SOR	*31 32 *67 68	
SOPT	21 31 60 67	
T	15	
THI	19	
V	*38 31 43 *62 *66 67 79	
WARMED	22 4	

## Subroutine KNOT

```

1 C SUBROUTINE 'KNOT' LOCATES THE KNOT DEFECTS ON THE BOARD FACE.
2 C ANGLFS ARE RECORDED CLOCKWISE WITH 0 DEGREES DEFINED AS THE
3 C LINE FROM THE CENTER OF THE LOG PERTINENT TO THE BOARD FACE
4 C
5 C
6 C SUBROUTINE KNOT (KHIGH,KTANG$,KLEN,F)
7 C
8 C IMPLICIT REAL (K-L)
9 C INTEGER ID(22,2),NODEFC(2),F
10 C REAL (BOARD(4),RDLTB(4),BDLUDS(2),R(4),FLX(26,2),
11 C FLY(26,2),FLU(26,2),FLY(26,2))
12 C COMMON /DEFEC/ NODEFC,FLX,FLY,FLU,FL,
13 C /BOARD/ L,D,K,T,LGS,RB,DISTBD,BDLN,BOARDL,KNOT$,R,RCUT,BF
14 C
15 C SEE IF THE KNOT IS LONG ENOUGH TO REACH THE BOARD FACE
16 C
17 C IF (KLEN.LT.DISTBD(F)) RETURN
18 C
19 C CALCULATE BOARD HALF-WIDTH ANGLES, KNOT DEFECT HALF-LENGTH AS
20 C PROJECTED ON THE FACE, AND LOG RADIUS AT KNOT HEIGHT
21 C
22 C EDLUDS(1) = DEG(ATAN(BOARDW(1)/DISTBD(F)))
23 C EDLUDS(2) = DEG(ATAN(BOARDW(2)/DISTBD(F)))
24 C DY = DISTBD(F)*ATAN(RD(KNOT$)),
25 C LRADUS = RB-KHIGH*TAN(RD(KNOT$))
26 C
27 C THE KNOT EFFECTIVELY ENDS AT THE OUTER EDGE OF THE LOG
28 C
29 C IF (H.EQ.1.LT.LRADUS) KLEN = LRADUS
30 C SINCE THE LOG IS TAPERED, THE OUTER BOARDS MAY NOT BE THE FULL LOG
31 C LENGTH, SO KNOTS ABOVE THE END OF THE BOARD ARE NOT CONSIDERED
32 C
33 C
34 C IF (LPADUS.LT.DISTBD(F)) RETURN
35 C
36 C ROTATE THE LOG TO THE CURRENT FACE
37 C
38 C IF (F.EQ.0.2) KTANG$ = AMOD(KTANG$+278.,360.)
39 C IF (F.EQ.0.3) KTANG$ = AMOD(KTANG$+180.,360.)
40 C IF (F.EQ.0.4) KTANG$ = AMOD(KTANG$+90.,360.)
41 C ANGLE OF ROTATION PUTS KNOT COMPLETELY OUTSIDE BOARD
42 C
43 C IF ((KTANG$ GT 180.) AND ((KTANG$+KNOT$).LT.(360.-BDLUDS(1)))) GO TO 9
44 C
45 C IF ((KTANG$ LT 180.) AND ((KTANG$+KNOT$).LT.(360.-BDLUDS(1)))) GO TO 9
46 C
47 C IF ((KTANG$.LE.180.) AND ((KTANG$-KNOT$).LT.80.)) GO TO 9
48 C
49 C FIND THE QUADRANT IN WHICH THE KNOT LIES (ASSUME QUADRANT ONE)
50 C
51 C QUADRANT TWO
52 C
53 C QUADRANT THREE
54 C
55 C QUADRANT FOUR
56 C
57 C QUADRANT ONE
58 C
59 C IF ((KTANG$.LT.180.) AND ((KTANG$+KNOT$).LT.(360.-BDLUDS(1)))) GO TO 3
60 C
61 C QUADRANT FOUR
62 C
63 C IF ((KTANG$.LT.90.) GO TO 2
64 C
65 C QUADRANT ONE
66 C
67 C SEE IF KNOT CENTER IS WITHIN ONE KNOT HALF-ANGLE OF 0 DEGREES
68 C
69 C
70 C IF ((KTANG$-KNOT$.LT.0.) GO TO 2
71 C
72 C CALCULATE DISTANCE TO FACE ALONG BOTH SIDES OF KNOT
73 C
74 C ANGBBL1 = DISTBD(F)*COS(RAD((KTANG$-KNOT$)))
75 C ANGBBL2 = DISTBD(F)*COS(RAD((KTANG$+KNOT$)))

```

```

154 FULWID = 1.7*4.12.
155 60 IN 8
156 C FAR SIDE OF KNOT DOES NOT REACH FACE
157 C
158 C
159 C  $\epsilon_2 = \text{BHP}(P_1 + \text{SPT1}(\text{LEN}^*2 - \text{DISTBD}(F))^*2)$ 
160 C FULWID = DISTBD(F) * TANRAD(360. - YTANGS**NOT$)
161 C *DISTBD(F) * TANRAD(360. - YTANGS**NOT$)
162 C
163 C GO TO 8
164 C
165 C KNOT IS WITHIN ONE HALF-ANGLE OF 0 DEGREES
166 C
167 C  $\epsilon_3 = \text{AMIN}(\text{SPT1}(\text{LEN}^*2 - \text{DISTBD}(F))^*2,$ 
168 C *DISTBD(F) * TANRAD(P_1 - L))
169 C
170 C IF KNOT COMPLETELY INTERSECTS THE FACE, ITS LENGTH IS EXACTLY AS
171 C LENGTH OF ITS PROJECTED LENGTH.
172 C P = AMIN(SPT1(\text{LEN}^*2 - \text{DISTBD}(F))^*2,
173 C *DISTBD(F) * TANRAD(P_1 - L))
174 C
175 C KNOT IS NOT COMPLETELY INTERSECTS THE FACE, OTHERWISE, ITS LENGTH IS ESTIMATED AS A
176 C PROJECTION OF ITS PROJECTED LENGTH.
177 C
178 C P = AMIN(DISTBD(F) * SPT1 * YTANGS**NOT$)
179 C
180 C
181 C KNOT IS WITHIN ONE HALF-ANGLE OF 0 DEGREES
182 C
183 C  $\epsilon_4 = \text{AMIN}(\text{SPT1}(\text{LEN}^*2 - \text{DISTBD}(F))^*2,$ 
184 C *DISTBD(F) * TANRAD(NOTS-YTANGS))
185 C
186 C KNOT NEARLY PEPENDICULAR HAVE THE YIP WIDTH ESTIMATED BY THE YIP
187 C LENGTH
188 C
189 C FULWID = YT*2.
190 C
191 C GO TO 8
192 C QUADRANT TWO
193 C
194 C 3 ANGEL = DISTBD(F) * COS(YTANGS - NOT$)
195 C
196 C KNOT IS NOT LONG ENOUGH TO REACH FACE
197 C
198 C IF (YLEN.LT.ANGB01) GO TO 9
199 C FIND INTERSECTION OF NEAR SIDE OF KNOT AND FACE
200 C
201 C  $\epsilon_1 = \text{BHP}(P_1 + \text{SPT1}(\text{LEN}^*2 - \text{DISTBD}(F))^*2,$ 
202 C *YTANGS**NOT$) GE 89.5;
203 C
204 C FULWID = LEGEND(F) * TANRAD(F) * YTANGS - NOT$,
205 C
206 C IF (YTANGS - NOT$) LT -19.5
207 C . . . FULWID = MIN(YLADS.DISTBD(F) * TANRAD(YTANGS - NOT$),
208 C *DISTBD(F) * TANRAD(YTANGS - NOT$))
209 C
210 C GO TO 8
211 C QUADRANT THREE
212 C
213 C 4 ANGBO1 = DISTBD(F) * COS(YTANGS - YTANGS - YNOT$)
214 C
215 C IF (YLEN.LT.ANGB01) GO TO 9
216 C
217 C  $\epsilon_2 = \text{BHP}(P_1 + \text{SPT1}(\text{LEN}^*2 - \text{DISTBD}(F))^*2,$ 
218 C *YTANGS - NOT$) LE 270.5;
219 C
220 C FULWID = LADS.DISTBD(F) * TANRAD(360. - YTANGS - YNOT$)
221 C
222 C IF (YTANGS - NOT$) GT 270.5
223 C . . . FULWID = MIN(YLADS.DISTBD(F) * TANRAD(360. - YTANGS - YNOT$),
224 C *DISTBD(F) * TANRAD(360. - YTANGS - YNOT$))
225 C
226 C GO TO 8
227 C QUADRANT FOUR
228 C
229 C 5 IF (YTANGS - NOT$) LT 0. GO TO 7
230 C ANGB01 = DISTBD(F) * COS(YTANGS - YTANGS - YNOT$)
231 C ONGB02 = DISTBD(F) * COS(YTANGS - YTANGS - YNOT$)
232 C IF (YLEN.LT.ANGB01) GO TO 9
233 C
234 C  $\epsilon_1 = \text{BHP}(P_1 + \text{SPT1}(\text{LEN}^*2 - \text{DISTBD}(F))^*2,$ 
235 C *YTANGS - NOT$) LE 270.5;
236 C
237 C FULWID = LADS.DISTBD(F) * TANRAD(360. - YTANGS - YNOT$)
238 C
239 C IF (YTANGS - NOT$) GT 270.5
240 C . . . DISTBD(F) * TANRAD(360. - YTANGS - YNOT$)
241 C
242 C GO TO 8
243 C QUADRANT FIVE
244 C
245 C * * * STATEMENT NUMBERS * * *
246 C
247 C 1 83 493
248 C 2 70 498
249 C 3 54 415
250 C 4 55 413
251 C 5 63 444
252 C 6 19 153
253 C 7 14 156
254 C 8 80 96 111 127 148 155 162 *178

```

## **Subroutine CORE**

CROSS	4	25	37
D	10	24	10
DISTNS	9	7	17
F	4	10	15
FLX	8	10	445
FLY	8	10	448
FLX	8	10	444
FLY	8	10	446
ID	7	10	449
INT	44	46	46
KNOTS	10	10	44
L	10	10	44
LOGS	10	10	426
NODEFC	7	10	43
R	8	10	43
RD	24	38	43
RB	10	10	43
RCUT	8	10	43
SIN	38	38	43
SORT	37	37	43
T	10	39	43
TDCTR	*38	39	43
TCNS	*16	*17	*19
TDIST	*24	25	37
TD	*40	49	38
TC	*25	46	38
TCNS	*23	46	38

### Subroutine GRADE

```

      1 C SUBROUTINE 'GRADE' CALLS THE U.S. FOREST PRODUCTS LABORATORY
      2 C GRADING PROGRAM
      3 C
      4 C
      5 C SUBROUTINE GRADE (NPG)
      6 C
      7 C INTEGER ID(22,2),NODEFC(2),IX(22),
      8 C REAL FLY(26,2),FLX(26,2),FLY,FUX,FUY,FLX,FLY,FUX,FUY,ID
      9 C COMMON /DEFER/ NODEFC,FLX,FLY,FUX,FUY,FLX
     10 C
     11 C
     12 C M = NODEFC(1)
     13 C IF (M.GT.22) GO TO 3
     14 C IF (M.FD.0) GO TO 2
     15 C DO 1 J=1,M
     16 C     IX(J,J) = ID(J,J)
     17 C CONTINUE
     18 C 2 NPG = 1
     19 C CALL GRD (M,IX,NPG,SM)
     20 C RETURN
     21 C IF NUMBER OF DEFECTS IS MORE THAN GRADING PROGRAM CAN HANDLE,
     22 C THE GRADE IS 34/38
     23 C
     24 C 25 3 NPG = 5
     25 C WRITE (6,100) 'DEFECT LIMIT EXCEEDED'
     26 C 100 FORMAT ('DEFECT LIMIT EXCEEDED')
     27 C
     28 C RETURN
     29 C

```

### \*\*\* STATEMENT NUMBERS \*\*\*

1	15	*17
2	14	*18
3	17	*25
100	26	*27

### \*\*\* VARIABLES \*\*\*

NAME	STATEMENT	NAME	STATEMENT
FLX	8	2	8
FLY	8	9	8
FUX	8	9	8
FUY	8	9	8
GRADE	5		
GRD	19		
ID	7	9	16
IX	7	*16	19
J	*17	16	
M	*12	13	14
NODEFC	7	9	12
NPG	5	*18	19
SM	19	*23	

## Subroutine PRICE

42

```

1 C SUBROUTINE *PRICE* DETERMINES THE BOARD GRADE BASED ON THE GRADES
2 C OF BOTH SIDES (OLD,NPG) AND CALCULATES THE BOARD VALUE
3 C
4 C SUBROUTINE PRICED (OLD,NPG,TVAL,PERC)
5 C
6 C
7 C IMPLICIT REAL (K,L)
8 C INTEGER OLD
9 C REAL PERC(5),BOARDW(2),R(4),DISTBD(4),RCUT(4)
10 C COMMON /BOARD/ L,D,K,T LOGS,R,B,DISTBD,BOARDL,KNOTS,R,RCUT,BF
11 C /PRICED/ FAS,SEL,ONEC,TDOC,THR8
12 C
13 C DETERMINE THE BOARD GRADE
14 C
15 C IF (OLD,ED,5,OR,NPG,ED,5) GO TO 3
16 C IF (OLD,ED,2,OR,NPG,ED,2) GO TO 1
17 C IF (OLD,ED,NPG) GO TO 4
18 C NPG = MP20(OLD,NPG)
19 C IF (NPG,ED,4) GO TO 8
20 C NPG = 2
21 C GO TO 6
22 C 1 IF (OLD,ED,4,OR,NPG,ED,4) GO TO 2
23 C NPG = 3
24 C GO TO 4
25 C 2 NPG = 4
26 C GO TO 2
27 C 3 NPG = 5
28 C
29 C AND CALCULATE THE BOARD VALUE
30 C
31 C
32 C 4 GO TO (5,6,7,8,9). NPG
33 C 5 V = BF*PERC
34 C GO TO 18
35 C 6 V = BF*SEL
36 C GO TO 18
37 C 7 V = BF*ONEC
38 C GO TO 18
39 C 8 V = BF*TDOC
40 C GO TO 18
41 C 9 V = BF*THR8
42 C
43 C INCREASE SURFACE MEASURE IN THIS GRADE AND THE TOTAL LOG VALUE
44 C
45 C 10 PERC(NPG) = BF*PERC(NPG)
46 C TVAL = TVAL+V
47 C RETURN
48 C END

```

## Variables

	NAME	VARIABLES	NAME
1	BDLEN	10	33
2	BF	10	35
3	BOARDL	9	10
4	D	10	
5	DISTBD	9	18
6	FAS	10	33
7	K	10	
8	KNOTS	10	
9	L	10	
10	LOG\$	10	
11	MA:30	10	
12	NPG	5	16
13	PB	10	
14	PCUT	9	10
15	QHCR	10	37
16	PEPC	5	9
17	PRICED	5	
18	PRITED	5	
19	F	9	10
20	PRICED	5	
21	PRICED	5	
22	PRICED	5	
23	PRICED	5	
24	PRICED	5	
25	PRICED	5	
26	PRICED	5	
27	PRICED	5	
28	PRICED	5	
29	PRICED	5	
30	PRICED	5	
31	PRICED	5	
32	PRICED	5	
33	PRICED	5	
34	PRICED	5	
35	PRICED	5	
36	PRICED	5	
37	PRICED	5	
38	PRICED	5	
39	PRICED	5	
40	PRICED	5	
41	PRICED	5	
42	PRICED	5	
43	PRICED	5	
44	PRICED	5	
45	PRICED	5	
46	PRICED	5	
47	PRICED	5	
48	PRICED	5	

## Statement Numbers

1	17	*23
2	23	*25
3	16	*28
4	18	25
5	32	*33
6	22	*32
7	31	*37
8	21	*32
9	17	*41
10	34	*36
11	38	40
12		*45

### Subroutine DECIDE

```

1 C SUBROUTINE 'DECIDE' FINDS THE LOG FACE THAT SHOULD BE CUT NEXT.
2 C USUALLY THIS IS THE FACE WITH THE HIGHEST GRADE, BUT, IF TWO OR
3 C MORE FACES HAVE THE SAME GRADE, THE FACE WITH THE LARGEST SURFACE
4 C MEASURE IS USED. IF TWO OR MORE FACES HAVE THE SAME GRADE AND
5 C SURFACE MEASURE, THE LAST FACE CONSIDERED IS USED.
6 C
7 C SUBROUTINE DECIDE (F,...)
8 C
9 C      IMPLICIT INTEGER (I,C,F,G)
10 C      COMMON /DECID/ CPPLT, GRDCTM(4), SHF(4)
11 C      DIMENSION CPPLT(4), GRDBST, GRDCDM, SHBEST, SHF
12 C
13 C      JCPLT = 0
14 C      GRDBST = 6
15 C      GRDBEST = 6
16 C
17 C      FOR EACH FACE OF THE LOG...
18 C
19 C      DO 5 J=1,4
20 C
21 C      IS FACE COMPLETELY CUT?
22 C
23 C      IF (CPPLT(J) .EQ. 1) GO TO 2
24 C      IF (CPPLT(J) .EQ. 1) GO TO 2
25 C      IF NOT, IS THIS FACE BETTER THAN...
26 C
27 C          IF ('GRDCDM(J).LT.GRDBEST) GO TO 3
28 C
29 C          OP ERASE, TO THE 'BEST' FACE?
30 C          IF ('GRDCDM(J).EQ.GRDBEST) GO TO 4
31 C          IF ('GRDCDM(J).EQ.GRDBEST) GO TO 4
32 C
33 C          Go To 1,
34 C
35 C      IF COMPLETELY CUT, INCREASE COUNT AND CHECK NEXT FACE
36 C
37 C          J = JPPLT + JPPLT+1
38 C          Go To 5
39 C
40 C      THIS FACE IS BETTER. SAVE GRADE AND SURFACE MEASURE: CHECK
41 C      NEXT FACE
42 C
43 C          F = J
44 C          GRDBEST = GRDCDM(J)
45 C          SHBEST = SHF(J)
46 C          Go To 5
47 C
48 C      USE SURFACE MEASURE TO BREAK TIES
49 C
50 C          4 IF (SHF(J).LT.SHBEST) GO TO 5
51 C          F = J
52 C          SHBEST = SHF(J)
53 C
54 C          5 CONTINUE
55 C
56 C          WHEN ALL FOUR FACES ARE COMPLETELY CUT, ROTATE LOG 15 DEGREES
57 C          AND PROCESS
58 C
59 C          IF ('JFPLT.CD.4) RETURN.
60 C
61 C          END

```

### END STATEMENT NUMBERS

	STATEMENT NUMBERS
DECIDE	2
DECIDE	3
DECIDE	4
DECIDE	5
DECIDE	24
DECIDE	28
DECIDE	443
DECIDE	32
DECIDE	450
DECIDE	33
DECIDE	38
DECIDE	46
DECIDE	58
DECIDE	453
DECIDE	24
DECIDE	443
DECIDE	32
DECIDE	444
DECIDE	32
DECIDE	44
DECIDE	32
DECIDE	44
DECIDE	44
DECIDE	32
DECIDE	43
DECIDE	32
DECIDE	43
DECIDE	32
DECIDE	45
DECIDE	36
DECIDE	51
DECIDE	50
DECIDE	52
DECIDE	52
DECIDE	51

#### Function DEG

```
1 C FUNCTION 'DEG' CONVERTS ITS ARGUMENT (IN RADIANS) TO DEGREES
2 C
3 C
4 C
5 C FUNCTION DEG (A)
6 C
7 C DEG = A*180.0/3.141592
8 C RETURN
9 C
END
```

\*\*\*\*\* VARIABLES \*\*\*\*\*

```
A      5      7
DEG    5      7
```

\*\*\*\*\* VARIABLES \*\*\*\*\*

```
a      5      7
RAD    5      7
```

#### Function RAD

```
1 C FUNCTION 'RAD' CONVERTS ITS ARGUMENT (IN DEGREES) TO RADIANS
2 C
3 C
4 C
5 C
6 C
7 C
8 C
9 C
FUNCTION RAD (A)
```

\*\*\*\*\* VARIABLES \*\*\*\*\*

```
a      5      7
RAD    5      7
```

## Live Sawing (LIVE)

```

1 C 'LINE' SAW THE LOG THROUGH ITS CENTRAL AXIS. THE PLANE OF EACH
2 C SUBSEQUENT SAW CUT (AND HENCE EACH BOARD FACE) IS PARALLEL TO THIS
3 C FIRST CUT. IN ADDITION, THE CORE DEFECT IS RIPPED OUT OF ANY BOARDS
4 C IN WHICH IT APPEARS AND THE RESULTING BOARDS REGRADED
5 C
6 C
7 C IMPLICIT REAL (K,L)
8 C REAL FFLX(26,2),FLY(26,2),FUY(26,2),PERC(5),PERCP(5),
9 C ANGLE(40),HEIGHT(40),KTLEN(40),DATA(40)
10 C INTEGER ID(122,2),NONEFC(2),CORFLG,OLD,GRDGET,GRDLFT,GRDMID,F
11 C RUMDN,BOARD,L,D,K,T,LOGS,RG,DISTD,BORDW,KNOTS,R
12 C ,>DEFER> NONEFC,FLX,FLY,FUX,FUY,IDL
13 C ,>TLOSS> RPLLOSS,RPKERF,LFTLOC,GRDGET,GRDLFT,GRDMID,
14 C ,>PRICF> FAS,SEL,ONEC,TUOC,TMFB
15 C
16 C
17 C THIS PROGRAM ASSUMES INPUT FROM UNIT 5 AND OUTPUTS TO UNITS 6 AND 8
18 C
19 C GET LOG DIAMETER (IN.), LOG LENGTH (IN.), KERF SIZES (IN.), AND
20 C BOARD THICKNESS (IN.)
21 C
22 C 1 READ (5,109,END=17) D
23 C READ (5,109,1,PF) ERFF
24 C READ (5,109,1,PF) ERFT
25 C READ (5,109,1,PF) ERT
26 C
27 C GET KNOT HALF-ANGLE (DEGREES) AND LOG HALF-TAPER (DEGREES)
28 C
29 C READ (5,109) KNOTS
30 C READ (5,109) LOGS
31 C
32 C GET CORE DEFECT PARAMETERS: DIAMETER (IN.), LINEAR OFFSET (IN.),
33 C AND ANGULAR OFFSET (DEGREES)
34 C
35 C READ (5,105) CORDIA
36 C READ (5,105) COFSET
37 C READ (5,105,1,PF) ERFF
38 C READ (5,105,1,PF) ERFT
39 C GET PRICE PER BOARD FOOT FOR EACH GRADE: NUMBER OF KNOTS
40 C
41 C READ (5,120) FAS,SEL,ONEC,TUOC,TMFB
42 C READ (5,101) MAX
43 C READ (5,101) MAX
44 C
45 C GET DATA AND CALCULATE RADIUS OF CORE EFFECT
46 C
47 C CALL LATE (DAT)
48 C
49 C GET KNOT ANGLES, WEIGHTS, AND LENGTHS
50 C
51 C DO 2 J=1,MAX
52 C READ (5,104) ANGLE(J),HEIGHT(J)
53 C 2 CONTINUE
54 C DO 3 J=1,MAX
55 C READ (5,105) KTLEN(J)
56 C
57 C GET KNOT ANGLES, WEIGHTS, AND LENGTHS
58 C
59 C CALCULATE LOG RADIUS AT TOP AND BOTTOM: TOTAL LOG CUBIC FEET
60 C
61 C RI = D/2
62 C RB = RI*WTN((RAD1*LOGS))
63 C 1,VOL = 1/3 * .33 * 141593 * (RB**2 + RB*RI + RI**2) * 4 / 144.
64 C
65 C OUTPUT HEADER
66 C
67 C WRITE (6,121) DAT
68 C WRITE (6,122) L,D,LOGS,LVOL
69 C DO 4 J=1,MAX
70 C WRITE (6,106) ANGLE(J),HEIGHT(J),KTLEN(J)
71 C
72 C 4 CONTINUE
73 C WRITE (6,123) KNOTS
74 C WRITE (6,114) CPNS,CFSET,CANGS
75 C WRITE (6,124) K,RPKEFF,T

```

```

151 C PUT IN CORE DEFECT
152 C CALL COREL (CRANS,CDSET,CNGS,CDFILE,1)
153 C
154 C PUT IN 'NOTS'
155 C
156 C DO 9 J=1,MAX
157 C      CALL KNOTL (HEIGHT(J),KTLEN(J),ANGLE(J))
158 C      9      CONTINUE
159 C
160 C PUT IN LINE DEFECTS
161 C      CALL KNOTL (HEIGHT(J),KTLEN(J),ANGLE(J))
162 C
163 C      CALL LNLNL
164 C
165 C AND GRADE THE FACE
166 C      CALL GRADE (NPG)
167 C
168 C SAME GRADE OF FIRST SIDE OF BOARD
169 C      IF (1.EQ.1) OLD = NPG
170 C
171 C DETERMINE BOARD GRADE AND VALUE BASED ON GRADES OF BOTH FACES
172 C
173 C RIP CORE DEFECT OUT OF BOARD
174 C      IF (1.EQ.2) CALL PRICEL (BF,OLD,NPG,TVAL,PERC,V)
175 C
176 C SUBTRACT BOARD THICKNESS AND PROCESS THE INNER FACE
177 C
178 C CALL RERIP (TVALRP,PERC,V,CDFILE,1,BF)
179 C
180 C
181 C
182 C      DISTBD = DISTBD-T
183 C
184 C      10      CONTINUE
185 C
186 C INCREASE BOARD COUNT AND CUT ANOTHER BOARD
187 C      NBD = NBD+1
188 C      GO TO 6
189 C
190 C TURN LOG 180 DEGREES TO CUT OTHER SIDE
191 C
192 C
193 C DO 12 J=1,MAX
194 C      ANGLE(J) = ANGLED(J)+180.0,360.0)
195 C      12      CONTINUE
196 C      CNGS = ANGDC(CNGS+180.0,360.0)
197 C      13      CONTINUE
198 C
199 C PRINT RESULTS FOR THIS POSITION
200 C
201 C IPROTAT = IND015-15
202 C      URITE (6,100) IROTRAT
203 C      UPI1F (6,112) (PERC(J),J=1,5)
204 C      TOTS = PEPE(1)+PERC(2)+PERC(3)+PERC(4)+PERC(5)
205 C      UPITE (6,111) TOTS
206 C      UPITE (6,203) IROTRAT
207 C      URITE (6,100) (PERC(J),J=1,5)
208 C      UPI1F (6,112) TOTS
209 C      DO 14 J=1,5
210 C          PERC(J) = (PERC(J)^AVOL)*100.
211 C          PERC(J) = (PERC(J)^AVOL)*100.
212 C          RPDTOTS = RPDTOTS+PERC(J)
213 C      CONTINUE
214 C      TOTS = MPTS*100./LVL
215 C      URITE (6,107) (ANGLE(J),J=1,MAX)
216 C      UPI1F (6,108) (PERC(J),J=1,5)
217 C      UPITE (6,113) TOTS
218 C      URITE (6,118) RPLOSS
219 C      URITE (6,205) (PERC(J),J=1,5)
220 C      UPI1F (6,204) TVAL,TOTS,RPLOSS,RPOTS
221 C
222 C      SAME HIGH. LOG. AND AVERAGE YIELDS FOR THIS LOG
223 C
224 C      AVGS = AVE*PTVAL
225 C      AVRIPS = AVRIP*TVALRP
226 C      IF (TVAL.GT.HIRIPS) HIRIPS = TVALRP
227 C      IF (TVAL.LT.SHRIPS) SHRIPS = TVALRP
228 C
229 IF (TVALRP.LT.SHRIPS) LORIP = RPOTS
230 IF (TVALRP.GT.HIRIPS) HIRIPS = TVALRP
231 IF (TVALRP.LT.SHRIPS) SHRIPS = TVALRP
232 AVRIP = AVRIP*RPOTS
233 C
234 C      ROTATE LOG BY 15 DEGREES AND REPROCESS
235 C
236 DO 15 J=1,MAX
237      ANGLE(J) = ANDD(ANGLE(J)+15.,360.0)
238      15      CONTINUE
239      CANGS = ANDD(CANGS+15.0,360.0)
240      16      CONTINUE
241 C      PRINT HIGH. LDG. AND AVERAGE YIELDS FOR THIS LOG
242 C
243 C      AVRIP = AVRIP/12.
244 C      AVGS = AVGS/12.
245 C      AVRIPS = AVRIPS/12.
246 C      WRITE (6,115) HIGH,AVGS,SHRILLS
247 C      WRITE (6,116) HIRIPS,AVRIPS,SHRIPS
248 C      WRITE (6,119) HIRIP,AVRIP,LORIP
249 C
250 C
251 C      LOG COMPLETELY PROCESSED. READ PARAMETERS FOR NEXT LOG
252 C
253 C      GO TO 1
254 C
255 C      17 STOP
256 C
257 C      FORMATS FOR DEVICE 6 (PRINTER)
258 C
259 C      180 FORMAT (2(/), ' ROTATION IS ',14)
260 C
261 C      181 FORMAT (12)
262 C      182 FORMAT (' THERE ARE ',13,' KNOTS IN THIS LOG',/-,4X,' ANGLE
263 C      ',/17 LENGTH')
264 C      184 FORMAT (12E2)
265 C      185 FORMAT (12E2)
266 C      186 FORMAT (3(2/,F6.2))
267 C      187 FORMAT (' KNOT ANGLES: ',15F7.2)
268 C      188 FORMAT (' X AVAILABLE SURFACE MEASURE IN EACH GRADE WITHOUT
269 C      ',/18 ANGLE
270 C      189 FORMAT (12E2)
271 C      190 FORMAT (12E2)
272 C      191 FORMAT (12)
273 C      192 FORMAT (' SURFACE MEASURE PER GRADE 15',5F8.2)
274 C      193 FORMAT (' TOTAL PERCENT YIELD IS ',F7.3)
275 C      194 FORMAT (' CORE DEFECT RADIUS ',F6.3,' LINEAR OFFSET ',F6.3,
276 C      ', ANGULAR OFFSET ',F6.3)
277 C      195 FORMAT (' $ YIELD: HIGH ',F6.2,' AVERAGE ',F6.2,' LOW ',F6.2)
278 C      196 FORMAT (' $ YIELD: HIGH ',F6.2,' AVERAGE ',F6.2,' LOW ',F6.2)
279 C      197 FORMAT (' $ YIELD: HIGH ',F6.2,' AVERAGE ',F6.2,' LOW ',F6.2)
280 C      198 FORMAT (' RP RIP SURFACE MEASURE LOSS 15 ',F6.2,' BOARD FEET')
281 C      199 FORMAT (' RP RIP X YIELD: HIGH ',F6.2,' AVERAGE ',F6.2,
282 C      ', LOW ',F6.2)
283 C      200 FORMAT (SF6.2)
284 C      121 FORMAT (' LIVE SAWING WITH RP RIP ',SF6.2)
285 C      122 FORMAT (' LOG PARAMETERS: ',F6.1,' DIG ',F6.1,' TAPER',
286 C      ', FR.4, DEGREES. LOG VOLUME: ',F6.2,' CUBIC FEET')
287 C      123 FORMAT (' ANGLES MEASURED FROM ZERO DEGREES = EAST FOR A ',
288 C      ', VERTICALLY CUTTING SAW, KNOT TAPER ',F6.2)
289 C      124 FORMAT (' RP RIP SURFACE MEASURE LOSS 15 ',F6.2,' BOARD FEET')
290 C      125 FORMAT (' PRICES PER BOARD FOOT ',SF18.4)
291 C
292 C      FORMATS FOR DEVICE 8 (STORAGE MEDIUM)
293 C
294 C      201 FORMAT (F6.1,F5.1,13.3F5.1,F8.2,F6.3,F4.1,F6.2,F5.3,F6.3)
295 C      202 FORMAT (3(10F7.2))
296 C      203 FORMAT (SF7.4)
297 C      204 FORMAT (SF7.4)
298 C      205 FORMAT (13)
299 C      206 FORMAT (SF8.2)
300 C      207 FORMAT (SF6.2)
301 C
302 END

```

## \* \* \* STATEMENT NUMBERS \* \* \*

1	*23	253		FUX	8	11	*142
2	52	454		FUY	3	11	*144
3	55	457		GRADE	167		
4	70	*72		GRDLFT	10	11	
5	111	*114		GRDMID	10	11	
6	*131	189		GSDCT	10	11	
7	140	*145		HEIGHT	9	93	71
8	146	*146		HIGHT	*67	*226	247
9	157	*159		HIGHS	*94	*228	249
10	175	*184		HIPIP	*90	*230	248
11	131	*193		HIPIPS	1	135	153
12	193	*195		ID	10	11	*147
13	121	*157		IMX	*101	201	
14	209	*213		IPOAT	*281	202	
15	236	*238		J	*32	43	45
16	161	*246		KTLEN	2	456	456
17	23	*25		L	*148	141	142
180	702	*259		LEFTLOC	11	*24	62
181	43	*269		LOU\$	11	*31	62
183	69	*261		LORIP	*95	*225	249
184	53	*263		LVL	*63	68	77
185	36	37		MX	*43	52	55
186	71	*265		NBD	*186	*188	
187	215	*266		NODEF.C	10	11	*148
188	216	*267		NPG	167	171	175
189	23	24		OLD	10	*171	175
190	25	*271		MEC	11	*42	76
191	119	265		PERC	3	*112	175
192	111	*272		PERCP.P	8	*113	*211
193	112	*273		PPICEL	175		
194	203	*273		PAN	11	*126	
195	113	*274		PB	11	462	63
196	74	*275		PEP1P	177		
197	114	*275		PI	*51	F2	63
198	247	*277		PF.EFF	11	*25	75
199	115	*277		RLOSS	11	*115	218
200	116	*278		RPTOTS	*188	*212	220
201	213	*281		SEL	11	*42	76
202	75	*287		SHELLS	*88	*227	247
203	124	*289		SITIPS	*91	229	*231
204	76	*291		T	11	*26	75
205	125	*291		THB	62		
206	77	*295		TLLOSS	11	*42	76
207	251	79		TOTS	*187	*204	205
208	80	*296		TVRL	*189	175	224
209	262	*297		TVRLRP	*111	179	225
210	203	*298		TVOC	11	*42	76
211	264	220		V	175		
212	211	*309		WANL	163		
				** * VARIABLES ** *			
200	194	195		1	237	239	
201	496	493		2	71	81	
202	495	494		3	224	*245	
203	493	492		4	232	*244	
204	492	492		5	225	*245	
205	801	801		6	116	153	
206	491	491		7	175	179	
207	204	204		8	131	153	
208	205	205		9	115	179	
209	206	206		10	115	194	
210	207	207		11	115	196	
211	208	208		12	74	77	
212	209	209		13	123	61	
213	210	210		14	47	67	
214	211	211		15	116	169	
215	212	212		16	121	177	
216	213	213		17	62	62	
217	214	214		18	114	141	
218	215	215		19	11	*143	

## Subroutine KERF

```

1 C SUBROUTINE "KERF" CUTS A BOARD FROM THE LOG. THE BOARD WIDTH IS
2 C DEFINED AT THE CENTER. AND THE BOARD LENGTH FROM THE BOTTOM
3 C OF THE LOG
4 C
5 C SUBROUTINE KERF ( *, BF )
6 C
7 C IMPLICIT REAL ( K,L )
8 C REAL FIX(26.2),FLY(26.2),FUX(26.2),FUY(26.2)
9 C INTEGER INT22,22, NODEFC(2)
10 C COMMON /BOARD / L,D,K,I,LOG$R,DISTBD,BOLEN,BOARDL,KNOTS,R
11 C /DEFEC/ NODEFC,FLX,FLY,FUX,FUY,TD
12 C
13 C
14 C ALL BOARDS MUST BE AT LEAST 48 INCHES LONG, 2.5 INCHES WIDE AT
15 C THE TOP, AND 3 INCHES WIDE AT THE CENTER. IN ADDITION, THE TOTAL
16 C AMOUNT OF LOGE FROM THE CENTER TO THE TOP IS LIMITED TO 4 INCHES.
17 C THE BOARD IS CUT BACK IN ONE FOOT INCREMENTS IF THE WIDTH OR LENGTH
18 C TESTS FAIL. IF THE LENGTH TEST FAILS, ANOTHER BOARD IS CUT FROM
19 C THIS FIVE
20 C
21 C
22 C FIND OUTER FACE OF BOARD
23 C
24 C RT = D/2.
25 C R = R-T-K
26 C BOLEN = L
27 C
28 C IF THIS FACE OF THE LOG IS COMPLETELY CUT, EXIT
29 C
30 C IF (R.LT.0.) RETURN
31 C DISTBD = RB-R
32 C POSLEN = RTAN(RAD(90.-LOG$))
33 C
34 C
35 C CHECK FOR BOARD'S SHORTER THAN THE LOG DUE TO LOG TRAPEZOID
36 C
37 C IF (POSLEN.LT.BoLEN) BoLEN = POSLEN
38 C
39 C Boards :UST BF AT LEAST 4 FEET LONG
40 C
41 C 2 IF (BoLEN.LT.48.) GO TO 1
42 C
43 C DETERMINE HALF-WIDTH AT MID-LENGTH OF BOARD
44 C RM = D/2.+(.5-BoLEN/2.)*TAN(RAD(LOG$))
45 C
46 C IF (DISTBD.GT.RM) GO TO 3
47 C BOARD$ = SQR((RT**2-DISTBD**2))
48 C
49 C BOARD MUST BE AT LEAST 3 INCHES WIDE AT MID-LENGTH
50 C
51 C IF (BOARD$.LT.1.5) GO TO 3
52 C
53 C DETERMINE HALF-WIDTH AT TOP OF BOARD
54 C
55 C RT = PI*(L-BoLEN)*TAN(RAD(LOG$))
56 C IF (DISTBD.GT.RT) GO TO 3
57 C LT = SQR((RT**2-DISTBD**2))
58 C
59 C BOARD MUST BE AT LEAST 2.5 INCHES WIDE AT THE TOP
60 C
61 C IF (LT.LT.1.25) GO TO 3
62 C
63 C LIMIT THE TOTAL LOGE TO AVOID GENERATING MORE DEFECTS THAN THE
64 C GRADING PROGRAM CAN HANDLE. TOTAL WIDTH OF LOGE MUST BE LESS
65 C THAN 4 INCHES
66 C
67 C LBN = BOARDL-UT
68 C IF (LBN.GT.2.0) GO TO 3
69 C
70 C CONVERT BOARD WIDTH AND LENGTH TO EVEN QUARTER INCHES
71 C
72 C BOARD$ = INT(BOARD$)
73 C BOLEN = INT(BoLEN*.4)
74 C

```

```

75 C DEFINE BOARD EDGES IN QUARTER INCH UNITS
76 C
77 C FLY(26,1) = BOARDL*.4.
78 C FLX(26,1) = BoLEN*.4.
79 C FUY(26,1) = 0.6
80 C FUX(26,1) = 0.8
81 C
82 C CALCULATE UNROUNDED BOARD FEET
83 C
84 C BF = BoLEN*BOARD$*.4*T/144.
85 C
86 C RETURN
87 C
88 C CUT BoRD BY ONE FOOT
89 C
90 C - BoLEN = BoLEN-.12.
91 C GO TU 2
92 C

```

\*\*\* STATEMENT NUMBERS \*\*\*

1	26	42
2	42	91
3	47	52
4	47	57
5	47	62
6	47	69
7	47	90
8	27	38
9	42	46
10	46	56
11	46	74
12	46	79
13	46	85
14	47	85
15	47	57
16	47	58
17	32	31
18	73	46
19	73	56
20	73	32
21	73	33
22	44	45
23	44	56
24	44	56
25	44	56
26	44	56
27	44	56
28	44	56
29	44	56
30	44	56
31	44	56
32	44	56
33	44	56
34	44	56
35	44	56
36	44	56
37	44	56
38	44	56
39	44	56
40	44	56
41	44	56
42	44	56
43	44	56
44	44	56
45	44	56
46	44	56
47	44	56
48	44	56
49	44	56
50	44	56
51	44	56
52	44	56
53	44	56
54	44	56
55	44	56
56	44	56
57	44	56
58	44	56
59	44	56
60	44	56
61	44	56
62	44	56
63	44	56
64	44	56
65	44	56
66	44	56
67	44	56
68	44	56
69	44	56
70	44	56
71	44	56
72	44	56
73	44	56
74	44	56

\*\*\* STATEMENT NUMBERS \*\*\*

1	27	38
2	42	46
3	47	57
4	47	62
5	47	69
6	47	90
7	27	38
8	42	46
9	47	57
10	47	62
11	47	69
12	47	90
13	27	38
14	42	46
15	47	57
16	47	62
17	47	69
18	47	90
19	27	38
20	42	46
21	47	57
22	47	62
23	47	69
24	47	90
25	27	38
26	42	46
27	47	57
28	47	62
29	47	69
30	47	90
31	27	38
32	42	46
33	47	57
34	47	62
35	47	69
36	47	90
37	27	38
38	42	46
39	47	57
40	47	62
41	47	69
42	47	90
43	27	38
44	42	46
45	47	57
46	47	62
47	47	69
48	47	90
49	27	38
50	42	46
51	47	57
52	47	62
53	47	69
54	47	90
55	27	38
56	42	46
57	47	57
58	47	62
59	47	69
60	47	90
61	27	38
62	42	46
63	47	57
64	47	62
65	47	69
66	47	90
67	27	38
68	42	46
69	47	57
70	47	62
71	47	69
72	47	90
73	27	38
74	42	46

## Subroutine WANE

1 C SUBROUTINE 'WANE' LOCATES LINE DEFECTS ON THE BOARD

2 C

3 C SUBROUTINE LARTEL

4 C

5 C THE BOARD EDGE HAS A NEW LINE DEFECT IF THE WIDTH DROPS BY 1/2  
6 C INCH (1/4 INCH FOR EACH SIDE) FROM THAT AT THE MIDDLE. THE BOARD  
7 C IS SYMMETRICAL SO THE LINE DEFECTS OCCUR AT THE SAME PLACE ON  
8 C BOTH EDGES OF THE BOARD

9 C

10 C IMPLICIT REAL (K,L)

11 C INTEGER ID(22,2),NODEFC(2)

12 C REAL FLX(26,2),FLY(26,2),FLX(26,2),FLY(26,2),BOARD,BOARDL,kNOTs,R

13 C COMMON /BOARD/ L,D,K,T,LGS,RB,D15TB,D,BOLEN,BOARDL,kNOTs,R

14 C

15 C \*REFEC/ NODEFC,FLX,FLY,FLX,FUT,FLX

16 C

17 C CON1 = TAN(BRAD(LGS))

18 C LDN2 = DISTBD\*2

19 C IF (R8-SORT((DN2+BOARDL\*2))/CON1.GE.BoLEN) RETURN

20 C

21 C OLDH = BoLEN\*2.

22 C

23 C

24 C FIND THE PLACE ALONG THE BOARD WHERE THE HALF-WIDTH HAS DECREASED

25 C BY 1/4 INCH. THE DEFECT ENDS HERE AND EXTENDS FROM THE END OF THE

26 C LAST LINE DEFECT (OR FROM THE MIDDLE OF THE BOARD).

27 C

28 C V = V+0.25

29 C SDF = SORT((CON2\*(BOARD-V)/CON1.BoLEN)\*2)

30 C H = INT((M1\*(RB-SDF)/4)/CON1.BoLEN\*4.)

31 C

32 C NODEFC(1) = NODEFC(1)+1

33 C

34 C LIMIT NUMBER OF DEFECTS TO MAXIMUM ALLOWED BY GRADING PROGRAM (22)

35 C

36 C IF (.M.GT.22) GO TO 2

37 C

38 C SAVE LINE COORDINATES FOR GRADING PROGRAM

39 C

40 C FLY(M,1) = VM4.+1

41 C FLY(M,1) = 0.

42 C FLX(M,1) = H

43 C QDM(M,1) = 5

44 C

45 C 2 NODEFC(1) = NODEFC(1)+1

46 C H = NODEFC(1)

47 C SINCE THE BOARD IS SYMMETRICAL, USE THE PREVIOUSLY GENERATED HEIGHT

48 C FOR THE LINE DEFECTS ON THE OTHER EDGE OF THE BOARD

49 C

50 C WHEN THE DEFECT EXTENDS TO THE END OF THE BOARD, EXIT

51 C

52 C 3 IF (H.GE.BoLEN\*4.) RETURN

53 C

54 C THE NEXT LINE DEFECT BEGINS WHERE THIS ONE ENDS

55 C

56 C OLDH = H

57 C GO TO 1

58 C RETURN

59 C

60 C

61 C

62 C

63 C

64 C

65 C

66 C

67 C

\*\*\* STATEMENT NUMBERS \*\*\*

Label	1	*28	65
Label	2	*36	*49
Label	3	*51	*68
Label	4	*30	
Label	5	28	21
Label	6	29	52
Label	7	38	53
Label	8	28	29
Label	9	19	19
Label	10	14	14
Label	11	14	14
Label	12	14	14
Label	13	14	14
Label	14	14	14
Label	15	30	42
Label	16	12	14
Label	17	12	14
Label	18	14	14
Label	19	14	14
Label	20	48	52
Label	21	60	64
Label	22	64	
Label	23	30	30
Label	24	14	14
Label	25	14	14
Label	26	14	14
Label	27	18	18
Label	28	32	36
Label	29	48	48
Label	30	41	42
Label	31	32	43
Label	32	55	55
Label	33	56	56
Label	34	14	14
Label	35	14	14
Label	36	14	14
Label	37	14	14
Label	38	18	18
Label	39	28	28
Label	40	29	29
Label	41	40	53
Label	42	28	28
Label	43	1	1
Label	44		
Label	45		
Label	46		
Label	47		
Label	48		
Label	49		
Label	50		
Label	51		
Label	52		
Label	53		
Label	54		
Label	55		
Label	56		
Label	57		
Label	58		
Label	59		
Label	60		
Label	61		
Label	62		
Label	63		
Label	64		
Label	65		
Label	66		
Label	67		

## Subroutine KNOT

```

1 C SUBROUTINE 'KNOT' LOCATES THE KNOT DEFECTS ON THE BOARD FACE.
2 C ANGLES ARE MEASURED CLOCKWISE WITH 0 DEGREES DEFINED AS THE
3 C LINE FROM THE CENTER OF THE LOG PERPENDICULAR TO THE BOARD FACE
4 C
5 C
6 C SUBROUTINE KNOT( KHIGH, KLEN, KTANGS)
7 C
8 IMPLICIT REAL (K,L)
9 INTEGER ID(22,2), NODEFC(2)
10 REAL FLY(26,2), FLX(26,2), FUY(26,2)
11 COMMON /NODEFC/ NODECF,FLX,FLY,FUX,FUY,1D
12 /BOARD/ L,D,K,T,LOGS,RB,DISTBD,BDLEN,BOARDL,KNOTS,R
13 C
14 C SEE IF THE KNOT IS LONG ENOUGH TO REACH THE BOARD FACE
15 C
16 C IF (KLEN.LT.DISTBD) RETURN
17 C
18 C CALCULATE KNOT DEFECT HALF-LENGTH AS PROJECTED ON THE FACE LOG
19 C RADIUS AT KNOT HEIGHT, AND BOARD HALF-WIDTH ANGLE
20 C
21 C DY = DISTBD*TAN(PAD(KNOT$))
22 C RADIUS = RB-KHIGH*TAN(RAD(LOGS))
23 C RADIAT = (EQUIDISTBD*DISTBD)
24 C
25 C THE KNOT EFFECTIVELY ENDS AT THE OUTER EDGE OF THE LOG
26 C
27 C IF (KLEN.GT.RADIUS) KLEN = RADIUS
28 C SINCE THE LOG IS TAFFET, THE OUTER BOARDS MAY NOT BE THE FULL LOG
29 C LENGTH, 30 KNOTS ABOVE THE END OF THE BOARD ARE NOT CONSIDERED
30 C
31 C IF (RADIUS.LT.DISTBD) RETURN
32 C
33 C WHILE IN PIASTRAH MUL, NOT CUPPLA MUL, OUTSIDE BNP, P:
34 C IF ((KTANGS.GT.180.) .AND. ((KTANGS+KNOT$).LT.(360.-BDWIDS$))) RETURN
35 C IF ((KTANGS.LE.180.) .AND. ((KTANGS+KNOT$).GT.BDWIDS$)) RETURN
36 C
37 C FIND THE QUADRANT IN WHICH THE KNOT LIES (ASSUME QUADRANT ONE)
38 C
39 C IF (KTANGS.GT.270.) GO TO 5
40 C QUADRANT TWO
41 C
42 C IF ((KTANGS.LE.180.) .AND. ((KTANGS+KNOT$).LT.BDWIDS$)) GO TO 3
43 C
44 C IF ((KTANGS.GT.270.) .AND. ((KTANGS+KNOT$).LT.(360.-BDWIDS$)))
45 C GO TO 4
46 C
47 C IF (KTANGS.GT.270.) GO TO 5
48 C
49 C IF (KTANGS.LE.180.) .AND. ((KTANGS+KNOT$).LT.(360.-BDWIDS$)))
50 C GO TO 4
51 C
52 C QUADRANT THREE
53 C
54 C IF (KTANGS.GT.270.) GO TO 5
55 C
56 C QUADRANT ONE
57 C
58 C SEE IF KNOT CENTER IS WITHIN ONE KNOT HALF-ANGLE OF 0 DEGREES
59 C
60 C
61 C IF (KTANGS-KNOT$.LT.0.) GO TO 2
62 C
63 C CALCULATE DISTANCE TO FACE ALONG BOTH SIDES OF KNOT
64 C
65 C ANGBD1 = DISTBD*COS(PAD(KTANGS-KNOT$))
66 C ANGBD2 = DISTBD*COS(PAD(KTANGS+KNOT$))
67 C KNOT IS NOT LONG ENOUGH TO REACH FACE
68 C
69 C IF (KLEN.LT.ANGBD1) RETURN
70 C
71 C FIND INTERSECTION OF NEAR SIDE OF KNOT AND FACE
72 C
73 C XR = BOARDU-DISTBD*TAN(PAD(KTANGS-KNOT$)) GO TO 1
74 C
75 C

```

FAR SIDE OF KNOT REACHES FACE

```

76 C
77 C
78 C XR = EQUIDISTBD*TAN(PAD(KTANGS+KNOT$))
79 C FULWID = (XR-XL)/2.
80 C GO TO 8
81 C
82 C IF (XOF OF KNOT DOES NOT REACH FACE
83 C
84 C 1.21 1.19PI*(1-INT((LEN**2-11*CTRN**2))
85 C 1.21 (FULWID+1)*TAN(PAD(KTANGS-KNOT$))-DISTBD*TAN(CRAYVYTANGS-KNOT$))
86 C
87 C 1.21 GO TO 8
88 C
89 C KNOT IS WITHIN ONE KNOT HALF-ANGLE OF 0 DEGREES
90 C
91 C 2.21 ANG1 = SORT((KLEN**2-DISTBD**2),DISTBD*TAN((RAD(KNOTS+KTANGS$)))
92 C 2.21 XR = BOARDU-A,
93 C 2.21 YP = ANIN((SORT((LEN**2-DISTBD**2),DISTBD*TAN((RAD(KNOTS-KTANGS$)))
94 C 2.21 -R + EQUIDBL+P))
95 C
96 C KNOTS NEARLY PERPENDICULAR HAVE THEIR WIDTH ESTIMATED BY THEIR
97 C LENGTH
98 C
99 C FULWID = DY**2.
100 C
101 C GO TO 8
102 C
103 C QUADRANT TWO
104 C
105 C 3 ANGEND = DISTBD*TAN(PAD(KTANGS+KNOT$))
106 C
107 C TO SHOT TO REACH FACE
108 C
109 C IF ((LEN.LT.ANGEND) RETURN
110 C
111 C 2.21 1.21 BOARDU-SORT((KLEN**2-DISTBD**2)
112 C 2.21 XR = BOARDU-SORT((KLEN**2-DISTBD**2)
113 C 2.21 YP = ANIN((DISTBD*TAN((RAD(KTANGS-KNOT$)))
114 C 2.21 -FULWID + EQUIDBL*DISTBD*TAN((RAD(KTANGS-KNOT$)))
115 C 2.21 -FULWID + ANIN((LEN**2-1.E-85.5))
116 C 2.21 -DISTBD*TAN(PAD(KTANGS+KNOT$))
117 C
118 C QUADRANT THREE
119 C
120 C 4 ANGBD1 = DISTBD*COS(PAD(360.-KTANGS-KNOT$))
121 C 4 IF ((LEN.LT.ANGBD1) RETURN
122 C 4 XR = BOARDU+DOP((KLEN**2-DISTBD**2)
123 C 4 2.21 EQUIDBL+DOP((KLEN**2-DISTBD**2)
124 C 4 2.21 1.21 (FULWID - LEN**2-1.E-270.5)
125 C 4 2.21 (FULWID - LEN**2-1.E-270.5)
126 C 4 2.21 IF ((LEN.LT.ANGBD1) RETURN
127 C 4 2.21 (FULWID - ANIN((LEN**2-1.E-270.5))
128 C 4 2.21 -DISTBD*TAN(PAD(360.-KTANGS-KNOT$))
129 C 4 2.21 -DISTBD*TAN(PAD(360.-KTANGS-KNOT$))
130 C
131 C QUADRANT FOUR
132 C
133 C 5 IF ((LEN.LT.ANGBD1) GO TO 7
134 C 5 IF ((LEN.LT.ANGBD1) DISTBD*COS((RAD(360.-KTANGS-KNOT$)))
135 C 5 ANGBD1 = DISTBD*COS((RAD(360.-KTANGS-KNOT$))
136 C 5 ANGBD2 = DISTBD*COS((RAD(360.-KTANGS-KNOT$))
137 C 5 IF ((LEN.LT.ANGBD1) RETURN
138 C 5 XR = BOARDU+DISTBD*TAN((RAD(360.-KTANGS-KNOT$)))
139 C
140 C
141 C 6 IF ((LEN.LT.ANGBD2) GO TO 6
142 C 6 IF ((LEN.LT.ANGBD2) EQUIDBL+DISTBD*TAN((RAD(360.-KTANGS-KNOT$)))
143 C 6 IF ((LEN.LT.ANGBD2) FULWID = (XR-XL)/2.
144 C
145 C FAR SIDE OF KNOT DOES NOT REACH FACE
146 C
147 C 6 IF ((LEN.LT.ANGBD1) DISTBD*TAN((RAD(360.-KTANGS-KNOT$)))
148 C 6 XR = BOARDU+DISTBD*TAN((RAD(360.-KTANGS-KNOT$)))
149 C 6 -DISTBD*TAN((RAD(360.-KTANGS-KNOT$)))
150 C
151 C GO TO 8
152 C
153 C

```

```

154 C THE KNOT IS WITHIN ONE KNOT HALF-ANGLE OF 0 DEGREES
155 C
156    X0 = MIN(1,SQRT(LEN**2-DISTBD**2),1
157    DISTBD=SQRT(TAN(3.4*NOTS)*360.),1
158    X0 = SQRT(1-X0**2-SQRT(LEN**2-DISTBD**2))
159    NOTS = MIN(1,SQRT(LEN**2-TAN(3.4*TAN(3.4*NOTS))),1
160    DISTBD=TAN(PI/360.*X0*TAN(3.4*NOTS)))
161    X0 = BORDERDX*D
162    FULUID = D*X2
163 C IF A KNOT COMPLETELY INTERSECTS THE FACE, ITS LENGTH IS EXACTLY AS
164 C MEASURED ON THE FACE. OTHERWISE, ITS LENGTH IS ESTIMATED AS A
165 C FRACTION OF IT. PROJECTED LENGTH
166 C
167 L IF T = HIGH-DY, FULUID = 1
168 T = HIGH-DY
169 T = HIGH-HY
170 T = HIGH-MY
171 C DEFECT LOCATIONS FOUND. CHECK THAT THEY DO NOT EXTEND BEYOND THE
172 C BOARD EDGES AND CONVERT TO QUARTER INCH UNITS FOR THE GRAPHING
173 C PROGRAM
174 C
175 C
176    L = MAX(0,INT((Z-4.1)
177    *INT((INT(BOLEN4.1)-INT(Z*4.1)+1.1),
178    *B+0.0,INT(BOLEN4.1)-INT(Z*4.1)+1.1);
179    CT = MIN(0,INT(BOLEN4.1)-INT(Z*4.1)+1.1));
180 C LIMIT NUMBER OF DEFECTS TO MAXIMUM ALLOWED BY GRAPHING PROGRAM (22)
181 C
182 C
183 C     NODEFCT = NODEFCT+1
184 C     IF NODEFCT > 1, RT, 201; RETURN
185 C
186 C     #4 EIGHT COORDINATES FOR GRAPHING. PROGRAM
187 C
188 C     NODEFCT(1,1) = 3
189 C     NODEFCT(1,2) = 1
190 C     NODEFCT(1,3) = 1
191 C     NODEFCT(1,4) = 3
192 C     NODEFCT(2,1) = 1
193 C     NODEFCT(2,2) = 1
194 C     NODEFCT(2,3) = 3
195 C     NODEFCT(2,4) = 1
196 C
197 C
198 C
199 C
200 C
201 C
202 C
203 C
204 C
205 C
206 C
207 C
208 C
209 C
210 C
211 C
212 C
213 C
214 C
215 C
216 C
217 C
218 C
219 C
220 C
221 C
222 C
223 C
224 C
225 C
226 C
227 C
228 C
229 C
230 C
231 C
232 C
233 C
234 C
235 C
236 C
237 C
238 C
239 C
240 C
241 C
242 C
243 C
244 C
245 C
246 C
247 C
248 C
249 C
250 C
251 C
252 C
253 C
254 C
255 C
256 C
257 C
258 C
259 C
260 C
261 C
262 C
263 C
264 C
265 C
266 C
267 C
268 C
269 C
270 C
271 C
272 C
273 C
274 C
275 C
276 C
277 C
278 C
279 C
280 C
281 C
282 C
283 C
284 C
285 C
286 C
287 C
288 C
289 C
290 C
291 C
292 C
293 C
294 C
295 C
296 C
297 C
298 C
299 C
300 C
301 C
302 C
303 C
304 C
305 C
306 C
307 C
308 C
309 C
310 C
311 C
312 C
313 C
314 C
315 C
316 C
317 C
318 C
319 C
320 C
321 C
322 C
323 C
324 C
325 C
326 C
327 C
328 C
329 C
330 C
331 C
332 C
333 C
334 C
335 C
336 C
337 C
338 C
339 C
340 C
341 C
342 C
343 C
344 C
345 C
346 C
347 C
348 C
349 C
350 C
351 C
352 C
353 C
354 C
355 C
356 C
357 C
358 C
359 C
360 C
361 C
362 C
363 C
364 C
365 C
366 C
367 C
368 C
369 C
370 C
371 C
372 C
373 C
374 C
375 C
376 C
377 C
378 C
379 C
380 C
381 C
382 C
383 C
384 C
385 C
386 C
387 C
388 C
389 C
390 C
391 C
392 C
393 C
394 C
395 C
396 C
397 C
398 C
399 C
400 C
401 C
402 C
403 C
404 C
405 C
406 C
407 C
408 C
409 C
410 C
411 C
412 C
413 C
414 C
415 C
416 C
417 C
418 C
419 C
420 C
421 C
422 C
423 C
424 C
425 C
426 C
427 C
428 C
429 C
430 C
431 C
432 C
433 C
434 C
435 C
436 C
437 C
438 C
439 C
440 C
441 C
442 C
443 C
444 C
445 C
446 C
447 C
448 C
449 C
450 C
451 C
452 C
453 C
454 C
455 C
456 C
457 C
458 C
459 C
460 C
461 C
462 C
463 C
464 C
465 C
466 C
467 C
468 C
469 C
470 C
471 C
472 C
473 C
474 C
475 C
476 C
477 C
478 C
479 C
480 C
481 C
482 C
483 C
484 C
485 C
486 C
487 C
488 C
489 C
490 C
491 C
492 C
493 C
494 C
495 C
496 C
497 C
498 C
499 C
500 C
501 C
502 C
503 C
504 C
505 C
506 C
507 C
508 C
509 C
510 C
511 C
512 C
513 C
514 C
515 C
516 C
517 C
518 C
519 C
520 C
521 C
522 C
523 C
524 C
525 C
526 C
527 C
528 C
529 C
530 C
531 C
532 C
533 C
534 C
535 C
536 C
537 C
538 C
539 C
540 C
541 C
542 C
543 C
544 C
545 C
546 C
547 C
548 C
549 C
550 C
551 C
552 C
553 C
554 C
555 C
556 C
557 C
558 C
559 C
560 C
561 C
562 C
563 C
564 C
565 C
566 C
567 C
568 C
569 C
570 C
571 C
572 C
573 C
574 C
575 C
576 C
577 C
578 C
579 C
580 C
581 C
582 C
583 C
584 C
585 C
586 C
587 C
588 C
589 C
590 C
591 C
592 C
593 C
594 C
595 C
596 C
597 C
598 C
599 C
600 C
601 C
602 C
603 C
604 C
605 C
606 C
607 C
608 C
609 C
610 C
611 C
612 C
613 C
614 C
615 C
616 C
617 C
618 C
619 C
620 C
621 C
622 C
623 C
624 C
625 C
626 C
627 C
628 C
629 C
630 C
631 C
632 C
633 C
634 C
635 C
636 C
637 C
638 C
639 C
640 C
641 C
642 C
643 C
644 C
645 C
646 C
647 C
648 C
649 C
650 C
651 C
652 C
653 C
654 C
655 C
656 C
657 C
658 C
659 C
660 C
661 C
662 C
663 C
664 C
665 C
666 C
667 C
668 C
669 C
670 C
671 C
672 C
673 C
674 C
675 C
676 C
677 C
678 C
679 C
680 C
681 C
682 C
683 C
684 C
685 C
686 C
687 C
688 C
689 C
690 C
691 C
692 C
693 C
694 C
695 C
696 C
697 C
698 C
699 C
700 C
701 C
702 C
703 C
704 C
705 C
706 C
707 C
708 C
709 C
7010 C
7011 C
7012 C
7013 C
7014 C
7015 C
7016 C
7017 C
7018 C
7019 C
7020 C
7021 C
7022 C
7023 C
7024 C
7025 C
7026 C
7027 C
7028 C
7029 C
7030 C
7031 C
7032 C
7033 C
7034 C
7035 C
7036 C
7037 C
7038 C
7039 C
7040 C
7041 C
7042 C
7043 C
7044 C
7045 C
7046 C
7047 C
7048 C
7049 C
7050 C
7051 C
7052 C
7053 C
7054 C
7055 C
7056 C
7057 C
7058 C
7059 C
70500 C
70501 C
70502 C
70503 C
70504 C
70505 C
70506 C
70507 C
70508 C
70509 C
70510 C
70511 C
70512 C
70513 C
70514 C
70515 C
70516 C
70517 C
70518 C
70519 C
70520 C
70521 C
70522 C
70523 C
70524 C
70525 C
70526 C
70527 C
70528 C
70529 C
70530 C
70531 C
70532 C
70533 C
70534 C
70535 C
70536 C
70537 C
70538 C
70539 C
70540 C
70541 C
70542 C
70543 C
70544 C
70545 C
70546 C
70547 C
70548 C
70549 C
70550 C
70551 C
70552 C
70553 C
70554 C
70555 C
70556 C
70557 C
70558 C
70559 C
70560 C
70561 C
70562 C
70563 C
70564 C
70565 C
70566 C
70567 C
70568 C
70569 C
70570 C
70571 C
70572 C
70573 C
70574 C
70575 C
70576 C
70577 C
70578 C
70579 C
70580 C
70581 C
70582 C
70583 C
70584 C
70585 C
70586 C
70587 C
70588 C
70589 C
705800 C
705801 C
705802 C
705803 C
705804 C
705805 C
705806 C
705807 C
705808 C
705809 C
705810 C
705811 C
705812 C
705813 C
705814 C
705815 C
705816 C
705817 C
705818 C
705819 C
705820 C
705821 C
705822 C
705823 C
705824 C
705825 C
705826 C
705827 C
705828 C
705829 C
705830 C
705831 C
705832 C
705833 C
705834 C
705835 C
705836 C
705837 C
705838 C
705839 C
705840 C
705841 C
705842 C
705843 C
705844 C
705845 C
705846 C
705847 C
705848 C
705849 C
705850 C
705851 C
705852 C
705853 C
705854 C
705855 C
705856 C
705857 C
705858 C
705859 C
705860 C
705861 C
705862 C
705863 C
705864 C
705865 C
705866 C
705867 C
705868 C
705869 C
705870 C
705871 C
705872 C
705873 C
705874 C
705875 C
705876 C
705877 C
705878 C
705879 C
705880 C
705881 C
705882 C
705883 C
705884 C
705885 C
705886 C
705887 C
705888 C
705889 C
705890 C
705891 C
705892 C
705893 C
705894 C
705895 C
705896 C
705897 C
705898 C
705899 C
7058100 C
7058101 C
7058102 C
7058103 C
7058104 C
7058105 C
7058106 C
7058107 C
7058108 C
7058109 C
7058110 C
7058111 C
7058112 C
7058113 C
7058114 C
7058115 C
7058116 C
7058117 C
7058118 C
7058119 C
7058120 C
7058121 C
7058122 C
7058123 C
7058124 C
7058125 C
7058126 C
7058127 C
7058128 C
7058129 C
7058130 C
7058131 C
7058132 C
7058133 C
7058134 C
7058135 C
7058136 C
7058137 C
7058138 C
7058139 C
7058140 C
7058141 C
7058142 C
7058143 C
7058144 C
7058145 C
7058146 C
7058147 C
7058148 C
7058149 C
7058150 C
7058151 C
7058152 C
7058153 C
7058154 C
7058155 C
7058156 C
7058157 C
7058158 C
7058159 C
7058160 C
7058161 C
7058162 C
7058163 C
7058164 C
7058165 C
7058166 C
7058167 C
7058168 C
7058169 C
7058170 C
7058171 C
7058172 C
7058173 C
7058174 C
7058175 C
7058176 C
7058177 C
7058178 C
7058179 C
7058180 C
7058181 C
7058182 C
7058183 C
7058184 C
7058185 C
7058186 C
7058187 C
7058188 C
7058189 C
7058190 C
7058191 C
7058192 C
7058193 C
7058194 C
7058195 C
7058196 C
7058197 C
7058198 C
7058199 C
7058200 C
7058201 C
7058202 C
7058203 C
7058204 C
7058205 C
7058206 C
7058207 C
7058208 C
7058209 C
7058210 C
7058211 C
7058212 C
7058213 C
7058214 C
7058215 C
7058216 C
7058217 C
7058218 C
7058219 C
7058220 C
7058221 C
7058222 C
7058223 C
7058224 C
7058225 C
7058226 C
7058227 C
7058228 C
7058229 C
70582200 C
70582201 C
70582202 C
70582203 C
70582204 C
70582205 C
70582206 C
70582207 C
70582208 C
70582209 C
70582210 C
70582211 C
70582212 C
70582213 C
70582214 C
70582215 C
70582216 C
70582217 C
70582218 C
70582219 C
70582220 C
70582221 C
70582222 C
70582223 C
70582224 C
70582225 C
70582226 C
70582227 C
70582228 C
70582229 C
70582230 C
70582231 C
70582232 C
70582233 C
70582234 C
70582235 C
70582236 C
70582237 C
70582238 C
70582239 C
70582240 C
70582241 C
70582242 C
70582243 C
70582244 C
70582245 C
70582246 C
70582247 C
70582248 C
70582249 C
70582250 C
70582251 C
70582252 C
70582253 C
70582254 C
70582255 C
70582256 C
70582257 C
70582258 C
70582259 C
70582260 C
70582261 C
70582262 C
70582263 C
70582264 C
70582265 C
70582266 C
70582267 C
70582268 C
70582269 C
70582270 C
70582271 C
70582272 C
70582273 C
70582274 C
70582275 C
70582276 C
70582277 C
70582278 C
70582279 C
70582280 C
70582281 C
70582282 C
70582283 C
70582284 C
70582285 C
70582286 C
70582287 C
70582288 C
70582289 C
70582290 C
70582291 C
70582292 C
70582293 C
70582294 C
70582295 C
70582296 C
70582297 C
70582298 C
70582299 C
705822000 C
705822001 C
705822002 C
705822003 C
705822004 C
705822005 C
705822006 C
705822007 C
705822008 C
705822009 C
705822010 C
705822011 C
705822012 C
705822013 C
705822014 C
705822015 C
705822016 C
705822017 C
705822018 C
705822019 C
705822020 C
705822021 C
705822022 C
705822023 C
705822024 C
705822025 C
705822026 C
705822027 C
705822028 C
705822029 C
705822030 C
705822031 C
705822032 C
705822033 C
705822034 C
705822035 C
705822036 C
705822037 C
705822038 C
705822039 C
705822040 C
705822041 C
705822042 C
705822043 C
705822044 C
705822045 C
705822046 C
705822047 C
705822048 C
705822049 C
705822050 C
705822051 C
705822052 C
705822053 C
705822054 C
705822055 C
705822056 C
705822057 C
705822058 C
705822059 C
705822060 C
705822061 C
705822062 C
705822063 C
705822064 C
705822065 C
705822066 C
705822067 C
705822068 C
705822069 C
705822070 C
705822071 C
705822072 C
705822073 C
705822074 C
705822075 C
705822076 C
705822077 C
705822078 C
705822079 C
705822080 C
705822081 C
705822082 C
705822083 C
705822084 C
705822085 C
705822086 C
705822087 C
705822088 C
705822089 C
705822090 C
705822091 C
705822092 C
705822093 C
705822094 C
705822095 C
705822096 C
705822097 C
705822098 C
705822099 C
705822100 C
705822101 C
705822102 C
705822103 C
705822104 C
705822105 C
705822106 C
705822107 C
705822108 C
705822109 C
705822110 C
705822111 C
705822112 C
705822113 C
705822114 C
705822115 C
705822116 C
705822117 C
705822118 C
705822119 C
705822120 C
705822121 C
705822122 C
705822123 C
705822124 C
705822125 C
705822126 C
705822127 C
705822128 C
705822129 C
705822130 C
705822131 C
705822132 C
705822133 C
705822134 C
705822135 C
705822136 C
705822137 C
705822138 C
705822139 C
705822140 C
705822141 C
705822142 C
705822143 C
705822144 C
705822145 C
705822146 C
705822147 C
705822148 C
705822149 C
705822150 C
705822151 C
705822152 C
705822153 C
705822154 C
705822155 C
705822156 C
705822157 C
705822158 C
705822159 C
705822160 C
705822161 C
705822162 C
705822163 C
705822164 C
705822165 C
705822166 C
705822167 C
705822168 C
705822169 C
705822170 C
705822171 C
705822172 C
705822173 C
705822174 C
705822175 C
705822176 C
705822177 C
705822178 C
705822179 C
705822180 C
705822181 C
705822182 C
705822183 C
705822184 C
705822185 C
705822186 C
705822187 C
705822188 C
705822189 C
705822190 C
705822191 C
705822192 C
705822193 C
705822194 C
705822195 C
705822196 C
705822197 C
705822198 C
705822199 C
705822200 C
705822201 C
705822202 C
705822203 C
705822204 C
705822205 C
705822206 C
705822207 C
705822208 C
705822209 C
705822210 C
705822211 C
705822212 C
705822213 C
705822214 C
705822215 C
705822216 C
705822217 C
705822218 C
705822219 C
705822220 C
705822221 C
705822222 C
705822223 C
705822224 C
705822225 C
705822226 C
705822227 C
705822228 C
705822229 C
7058222200 C
7058222201 C
7058222202 C
7058222203 C
7058222204 C
7058222205 C
7058222206 C
7058222207 C
7058222208 C
7058222209 C
7058222210 C
7058222211 C
7058222212 C
7058222213 C
7058222214 C
7058222215 C
7058222216 C
7058222217 C
7058222218 C
7058222219 C
7058222220 C
7058222221 C
7058222222 C
7058222223 C
7058222224 C
7058222225 C
7058222226 C
7058222227 C
7058222228 C
7058222229 C
705822222000 C
705822222001 C
705822222002 C
705822222003 C
705822222004 C
705822222005 C
705822222006 C
705822222007 C
705822222008 C
705822222009 C
705822222010 C
705822222011 C
705822222012 C
705822222013 C
705822222014 C
705822222015 C
705822222016 C
705822222017 C
705822222018 C
705822222019 C
705822222020 C
705822222021 C
705822222022 C
705822222023 C
705822222024 C
705822222025 C
705822222026 C
705822222027 C
705822222028 C
705822222029 C
705822222030 C
705822222031 C
705822222032 C
705822222033 C
705822222034 C
705822222035 C
705822222036 C
705822222037 C
705822222038 C
705822222039 C
705822222040 C
705822222041 C
705822222042 C
705822222043 C
705822222044 C
705822222045 C
705822222046 C
705822222047 C
705822222048 C
705822222049 C
705822222050 C
705822222051 C
705822222052 C
705822222053 C
705822222054 C
705822222055 C
705822222056 C
705822222057 C
705822222058 C
705822222059 C
705822222060 C
705822222061 C
705822222062 C
705822222063 C
705822222064 C
705822222065 C
705822222066 C
705822222067 C
705822222068 C
705822222069 C
705822222070 C
705822222071 C
705822222072 C
705822222073 C
705822222074 C
705822222075 C
705822222076 C
705822222077 C
705822222078 C
705822222079 C
705822222080 C
705822222081 C
705822222082 C
70582222
```

## Subroutine CORE

```

1 C SUBROUTINE 'CORE' LOCATES THE CORE DEFECT ON THE BOARD FACE
2 C
3 C SUBROUTINE COREL (FRANU, COFSET, CANG$, COREFLG, I,
4 C
5 C IMPLICIT REAL (K,L)
6 C
7 CINTERP 10(22,-2), NODEFC(2), COFELG
8 C      OREAL, FLY(26,-2), FLY(26,-2), FUX(26,-2),
9 C      COFFL, BOARD, LD, K, TLOGS, RD, DISTD, BDLEN, BOARDU, KNOTS, R
10 C      NODEF, / NODEFC, FLY, FUX, FUY, FUL, 10
11 C
12 C
13 C IS DISTANCE TO THE BOARD FACE FROM THE CENTER OF THE CORE. EXIT
14 C IF THE BOARD FACE IS BEYOND THE CORE RADIUS
15 C
16 TDIST = DISTN-COFSET*COS(CRAD(CANG$))
17 C
18 C IF ((FRANU.LE.TDIST) RETURN
19 C NODEFC(11) = NODEFC(11)+1
20 C LIMIT NUMBER OF DEFECTS TO MAXIMUM ALLOWED BY GRADING PROGRAM '22'
21 C
22 C IF (INHDEF(11,GT,1)) PPUTUP
23 C
24 C LOCATE THE CENTER POINT OF THE INTERSECTION OF THE BUMPT FACE WITH
25 C THE CORE DEFECT. CALCULATE THE DISTANCE ON THE BOARD FROM THE
26 C CENTER POINT TO THE EDGES OF THE CORE DEFECT. THE CORE DEFECT
27 C IS NOT TAPESTRY. SO IT WILL EXTEND THE FULL LENGTH OF THE BOARD
28 C
29 C CONST = SQR(TDUDS**2-TDIST**2)
30 C TBUFCP = COFSET*(RAD(CANG$))
31 C XP = BOARDU*CONST-TBUFCP
32 C YL = BOARDU*CONST-TBUFCP
33 C
34 C SAVE CORE DEFECT COORDINATES FOR GRADING PROGRAM!
35 C
36 C FUX(INHDEF(11,1)) = INT(BDLEN**4,1)
37 C FLY(INHDEF(11,1)) = 0.0
38 C FUR(INHDEF(11,1)) = ANG(0,INT(YL/4,1),INT(XL/4,1))
39 C INHDEF(11,1) = 7
40 C
41 C IF A CORE DEFECT IS FOUND ON THE OUTER FACE OF BOARD, THE BOARD
42 C WILL BE RE-PEPPED IN AN ATTEMPT TO IMPROVE ITS WHITE HOLE VALUE
43 C
44 C IF (LEQ(1, COREFL)) 1
45 C      RETURN
46 C
47 C

```

\*\*\* VARTABLES \*\*\*

NAME	29	30	31	32	33	34	35	36	37	38
ANHINO	33									
BDLEN	9									
BOARDU										
CANG\$	4	16	30							
COREFL	4	16	30							
CORELT	*22	31	32							
FUXFL	1									
TBUFCP	11.	7	*15							
YLFL	1.	1.	1.							
YLUR	1.	1.	1.							

## Subroutine GRADE

```

1 C SUBROUTINE 'GRADE' CALLS THE U.S. FOREST PRODUCTS LABORATORY
2 C GRADING PROGRAM
3 C
4 C SUPPORTIVE GRADE 'NPG'
5 C
6 C INTEGER 1D(22,2), NODEFC(2), IX(22)
7 C PERL FLX(26,2), FLY(26,2), FUY(26,2)
8 C CUPON, DEFECT, NODEFC, FLX,FLY,FUY, ID
9 C
10 C
11 C
12 C M = NODEFC(1)
13 C IF (M,0,0) GO TO 2
14 C IF (M,GT,22) GO TO 3
15 C DO 1 J=1,H
16 C   IX(J) = 1D(J,1)
17 C
18 C 1 CONTINUE
19 C NPG = 1
20 C CALL GRD (M,IX,NPG,SP)
21 C
22 C IF NUMBER OF DEFECTS IS MORE THAN GRADING PROGRAM CAN HANDLE.
23 C THE GRADE IS 3A-3B
24 C
25 C 3 WRITE (6,190,
26 C 190 FORMAT (1X,NPG,5X, 'DEFECT LIMIT EXCEEDED')
27 C NPG = 5,
28 C RETURN
29 C

```

## Subroutine PRICE

```

1 C SUBROUTINE 'PRICE' DETERMINES THE BOARD GRADE BASED ON THE GRADES
2 C OF BOTH SIDES (OLD,NPG) AND CALCULATES THE BOARD VALUE
3 C
4 C
5 C SUBROUTINE PRICE (BF,OLD,NPG,TVAL,PERC,V)
6 C
7 C INTEGER OLD
8 C REAL PERC(.5)
9 C COMMON /PRICF/ FHS,SEL,ONEC,TLOC,THRS
10 C
11 C
12 C SIDE BELOW GRADE?
13 C IF (OLD,ED,5,0,P,NG,ED,6) RETURN
14 C
15 C DETERMINE THE BOARD GRADE
16 C
17 C IF (OLD,ED,5,0,R,NPG,ED,5) GO TO 3
18 C
19 C IF (OLD,ED,0,2,0,P,NPG,ED,2) GO TO 1
20 C IF (OLD,ED,0,P,NPG,ED,4) GO TO 4
21 C NPG = M4(0,OLD,NPG)
22 C   (NPG,ED,4) GO TO 8
23 C
24 C
25 C 1 IF (OLD,ED,4,0,P,NPG,ED,4) GO TO 2
26 C   (NPG = 3
27 C 2 IF (NPG = 4
28 C   CO TO 4
29 C
30 C
31 C 3 IFPG = 5
32 C   4 GO TO (5,6,7,8,9). NPG
33 C
34 C 5 IF PG = 5
35 C   6 V = BF+FGS
36 C   GO TO 10
37 C   6 V = UP+SEL
38 C   GO TO 18
39 C   7 V = BF+ONEC
40 C   GO TO 18
41 C   8 V = F+TLOC
42 C   GO TO 19
43 C   9 V = S+TLOC
44 C
45 C INCREASE SURFACE MEASURE IN THIS GRADE AND THE TOTAL LOS VALUE
46 C
47 C 10 PERC(NPG) = BF+PERC(NPG),
48 C TVAL = TVAL+V
49 C
50 C
*** STATEMENT NUMBERS ***
1 15 *17
2 17 *18
3 14 *19
4 15 *20
160
*** VARIABLE NAMES ***
FLX 9 9
FLY 8 9
FLX 8 9
FUY 8 9
GRADE 5
GRC 19
ID 7 9
IX 7 *16 19
J 15 *12 13 14 15 19
MDEFCT 5 12
NPG 5 *16 19 *27
SH 15
*** STATEMENT NUMBERS ***
1 19 *25
2 25 *28
3 18 *30
4 20 27 29 *31
5 31 *35
6 24 31 *37
7 31 *39
8 32 31 *41
9 31 *43
10 33 34 40 42 *47

```

**Subroutine RERIP**

```

      SUBROUTINE 'RERIP' DETERMINES WHETHER RIPING A BOARD INTO TWO OR
      THREE PIECES WILL IMPROVE ITS VALUE WHEN A CORE DEFECT IS FOUND ON
      THE OUTER FACE (CORFLG = 1) AND THE GRADE IS BELOW SELECTS. THE
      CORE DEFECT BOUNDRIES ON THE OUTER FACE DETERMINE THE RIP LINES
      1 C
      2 C
      3 C
      4 C
      5 C
      6 C
      7 C
      8 C
      9 C
      10 C
      11 C
      12 C
      13 C
      14 C
      15 C
      16 C
      17 C
      18 C
      19 C
      20 C
      21 C
      22 C
      23 C
      24 C
      25 C
      26 C
      27 C
      28 C
      29 C
      30 C
      31 C
      32 C
      33 C
      34 C
      35 C
      36 C
      37 C
      38 C
      39 C
      40 C
      41 C
      42 C
      43 C
      44 C
      45 C
      46 C
      47 C
      48 C
      49 C
      50 C
      51 C
      52 C
      53 C
      54 C
      55 C
      56 C
      57 C
      58 C
      59 C
      60 C
      61 C
      62 C
      63 C
      64 C
      65 C
      66 C
      67 C
      68 C
      69 C
      70 C
      71 C
      72 C
      73 C
      74 C
      75 C

      IMPLICIT REAL (F,L)
      INTEGER CORFLG, ID(22,2), NODEFC(2), GRDID, RGRADE, GRDLFT,
      LGRADE
      REAL FLY(26,2), FUX(26,2), FUY(26,2), PERCRP(5), DUMYRP(5)
      COMMON /DEFCL/ NODEFC, FLY, FUX, FUY, ID
      COMMON /RIPCL/ RFLOC, RLLOC, RGTLOC, GRDRT, GRDLFT, GRDID,
      TLLOSS, RPLSS, RPKRF, LDTLOC, RGTLOC, GRDRT, GRDLFT, GRDID,
      /BOARD/ L,D,K,T,LGOS$, RB, DISTBD, BDLEN, BOARBL, KNOT$, R
      SUBROUTINE RERIP (TVALRP, PERCRP, NPS, V, CORFLC, I, BF)

      OLD      31      47
      ONEC     9       39      41      43      47
      PERC     5       18      19      20      *21      22      25
      PRICL    5       14      18      19      20      *21      22      25
      SEL      9       37      43
      THIN    5       48
      TWO     5       41
      TUDC    5       37      39      41      43      48
      V        5

      CONVTRT  18      19      20      *21      22      25      *26      *28      *30
      WIDTH   28      29      30      *31      32      33      *34      35      *36
      TLLOSS  21      22      23      *24      25      26      *27      28      *29
      IF (I.EQ.2) GO TO 8
      IF NO CORE DEFECT. DO NOTHING
      TLLOSS = 0.0
      IF (CORFLG.EQ.0) RETURN
      FIND THE CORE DEFECT
      I = NODEFC(1)
      INDEX = 1
      DO 1 I=1,M
      IF (ID(I,1).EQ.7) INDEX = J
      1 CONTINUE
      IF (INDEX.EQ.1) NE,7) GO TO 12
      DEFINE RIP LOCATIONS IN QUARTER INCH UNITS
      LFTLOC = FLY1 INDEX,1
      RGTLOC = FUY1 INDEX,1
      GRADE OF MIDDLE PIECE ASSURED 3A/3B. OTHERS FAS
      GRDID = 5
      RGRADE = 1
      LGRADE = 1
      CALCULATE PIECE WIDTHS. IF EITHER SIDE IS TOO SMALL TO MAKE LUMBER
      (1/3"), ADD IT TO THE MIDDLE PIECE
      WDRGT = WIDTH-LFTLOC/4.
      IF (WDRGT<3.) 2,3,3
      2 RGTOR = WIDTH*4.*PKRF*4.
      GRADE = 6
      3 WDLFT = LFTLOC/4.
      IF (WDLFT<3.) 4,5,5
      4 LFTLOC = -PKRF*4.
      GRADE = 6
      5 WIDMD = (RGTLOC-LFTLOC)/4.-2.*PKRF
      IF (WIDMD<3.) 6,7,7
      6 TLLOSS = PKRF*(WIDMD-PKRF)
      7 GET GRADE OF OUTER FACES
      8 CALL RIP (RGRADE,LGRADE)
      9 GRADE = 6
      10 IF MIDDLE IS TOO SMALL TO MAKE LUMBER. ADD ITS SURFACE MEASURE TO
      THE RIP LOGS
      11 TLLOSS = PKRF*(WIDMD-PKRF)
      12 THE RIP LOGS
      13 GRADE = 6
      14 GRADE = 6
      15 GRADE = 6
      16 GRADE = 6
      17 GRADE = 6
      18 GRADE = 6
      19 GRADE = 6
      20 GRADE = 6
      21 GRADE = 6
      22 GRADE = 6
      23 GRADE = 6
      24 GRADE = 6
      25 GRADE = 6
      26 GRADE = 6
      27 GRADE = 6
      28 GRADE = 6
      29 GRADE = 6
      30 GRADE = 6
      31 GRADE = 6
      32 GRADE = 6
      33 GRADE = 6
      34 GRADE = 6
      35 GRADE = 6
      36 GRADE = 6
      37 GRADE = 6
      38 GRADE = 6
      39 GRADE = 6
      40 GRADE = 6
      41 GRADE = 6
      42 GRADE = 6
      43 GRADE = 6
      44 GRADE = 6
      45 GRADE = 6
      46 GRADE = 6
      47 GRADE = 6
      48 GRADE = 6
      49 GRADE = 6
      50 GRADE = 6
      51 GRADE = 6
      52 GRADE = 6
      53 GRADE = 6
      54 GRADE = 6
      55 GRADE = 6
      56 GRADE = 6
      57 GRADE = 6
      58 GRADE = 6
      59 GRADE = 6
      60 GRADE = 6
      61 GRADE = 6
      62 GRADE = 6
      63 GRADE = 6
      64 GRADE = 6
      65 GRADE = 6
      66 GRADE = 6
      67 GRADE = 6
      68 GRADE = 6
      69 GRADE = 6
      70 GRADE = 6
      71 GRADE = 6
      72 GRADE = 6
      73 GRADE = 6
      74 GRADE = 6
      75 GRADE = 6
      
```

```

76 C SAVE OUTER FACE GRADES
77 C GRDGT = RGRADE
78 C GDLFT = LGRADE
79 C RETURN
80 C
81 C INNER FACE OF BOARD TO BE RIPPED...
82 C
83 C
84 C
85 C 9 RIPVAL = 0.
86 C OLDRY = V
87 C DO 9 J=1,5
88 C      JUMPRP(J) = 0.
89 C
90 C IF BOARD IS FAS OR SELECTS.
91 C
92 C IF (MPC.ED.1.OR.MPG.ED.2) GO TO 11
93 C
94 C OR IF NO OUTER FACE CORE DEFECT FOUND, DO NOT RIP
95 C
96 C IF (CORFLG.ED.6) GO TO 11
97 C
98 C ANY PIECE THAT IS TOO SMALL TO MAKE LUMBER IS BELOW GRADE
99 C
100 C IT (CGRPLT.ED.6) RGRADE = 6
101 C IF (GRPLFT.ED.6) LGRADE = 6
102 C
103 C GET GRADE OF INNER FACES
104 C
105 C
106 C CALL PIP (GRADE, LGRADE)
107 C
108 C CALCULATE SURFACE MEASURE OF EACH PIECE
109 C
110 C WIDUD = (RATLLOC-LFTLOC)/4.-2.*RPKRF
111 C SPWID = WIDUD*HDENHT/144.
112 C SPFLT = (WIDTH-RGTLOC/4.)*WDLENHT/144.
113 C SPFLT = LFTLOC/4.*WDLENHT/144.
114 C DETERMINE GRADE AND VALUE OF EACH PIECE
115 C
116 C
117 C TOLDRY = GRDGT
118 C INDLT = GDLFT
119 C ITPD = GDMID
120 C CALL PIP(CSPFLT, IOLDRY, RGRADE, PIPVAL, DUMRPP, V)
121 C CALL PIP(CSPFLT, IOLDRY, LGRADE, RPIPVAL, DUMRPP, V)
122 C CALL PIP(CSPFLT, IOLDRY, IMRD, RPIPVAL, DUMRPP, V)
123 C
124 C IF RIPPING DOES NOT IMPROVE THE VALUE, DO NOT DO IT
125 C
126 C IF (RPIPVAL.LE.DUMRPP) GO TO 11
127 C OTHERWISE, INCREASE RERIP VALUES
128 C
129 C
130 C TWRPIP = TWLRPIP+RPIPVAL
131 C DO 18 J=1,5
132 C      PIPRIP(J) = PERCP(J)+JUMPRP(J)
133 C
134 C 10 CONTINUE
135 C
136 C
137 C CORFLG = 0
138 C RPLOSS = RPLOSS+TLLOSS
139 C      IF (GRADE.NE.6) RPLOSS = RPLOSS+HDENHT/144.
140 C      IF (LGRADE.NE.6) RPLOSS = RPLOSS+HDENHT/144.
141 C
142 C
143 C BOARD IS NOT TO BE RIPPED
144 C
145 C 11 PERCP(MPC) = PERCP(MPC)+RPLOSS
146 C 12 CORFLG = 6
147 C
148 C
149 C

```

86

## Subroutine RIP

```

1 C SUBROUTINE 'RIP' RIPS THE FACE INTO TWO OR THREE PIECES AND GETS
2 C THE GRADE OF EACH PIECE
3 C
4 C SUBROUTINE RIP (GRADE,LGRADE)
5 C
6 C IMPLICIT REAL (K,L)
7 C INTEGER ID(22,2),NODEFC(2),ITYPE(22),GRADE,GRADE,LGRADE
8 C
9 C      REAL FIX(26,2),FLY(26,2),FLUX(26,2),LX(22),LY(22),RIP
10 C      UV(22)
11 C      COMMON /RIP/ RPLSS,RPKFF,LFTLOC,RGTLOC,GRTF,GRLFT,GDMID,
12 C      TLOSS
13 C      DEFECT / NODEFC,FLX,FLY,FLUX,FUY,UV /
14 C
15 C      SAVE BOARD WIDTH
16 C
17 C      WIDTH = FLY(26,1)
18 C
19 C      DO 1 J=1,22
20 C      WIDTH = FLY(26,1)
21 C      SAVE TYPE, LOCATION, AND NUMBER OF DEFECTS
22 C      DO 1 J=1,22
23 C      ITYPE(J) = ID(J,1)
24 C      BX(J) = FLIX(J,1)
25 C      LY(J) = FUY(J,1)
26 C      LX(J) = FLX(J,1)
27 C      LY(J) = FLY(J,1)
28 C
29 C      1 CONTINUE
30 C      M = NODEFC(1)
31 C
32 C      IF EITHER PIECE IS BELOW GRADE, NOTHING IS DONE TO THAT PIECE
33 C      IF (GRADE.EQ.0) GO TO 34
34 C      RIGHT (UPPER) PIECE
35 C      IF (GRADE.EQ.6) GO TO 36
36 C
37 C      LOCATE DEFECTS IN THIS PIECE. IGNORING THOSE WHICH LIE BELOW THE
38 C      RIP LINE
39 C
40 C
41 C      N = 0
42 C      DO 2 J=1,M
43 C      IF (UY(J),LE,RGTLOC) GO TO 2
44 C      N = N+1
45 C      FLY(N,1) = UV(J)-RGTLOC
46 C      FLX(N,1) = APMAX(0,(LY(J))-RGTLOC)
47 C      TUX(N,1) = UX(J)
48 C      ELX(N,1) = LX(J)
49 C      ID(N,J) = ITYPE(J)
50 C
51 C      2 CONTINUE
52 C      DETERMINE PIECE WIDTH: NUMBER OF DEFECTS
53 C      FLY(26,1) = WIDTH-RGTLOC
54 C      NODEFC(1) = N
55 C
56 C      GRADE RIGHT (UPPER) PIECE
57 C
58 C      CALL, GRADE (GRADE)
59 C
60 C      LOCATE DEFECTS IN THIS PIECE. IGNORING THOSE WHICH LIE ABOVE THE
61 C      RIP LINE
62 C
63 C      3 IF (GRADE.EQ.6) GO TO 5
64 C
65 C      LOCATE DEFECTS IN THIS PIECE. IGNORING THOSE WHICH LIE ABOVE THE
66 C      RIP LINE
67 C
68 C      N = 0
69 C      DO 4 J=1,M
70 C      IF ((LY(J).GE.LFTLOC)) GO TO 4
71 C      N = N+1
72 C      FUY(N,1) = APMIN(LFTLOC,UY(J))
73 C      FLUX(N,1) = UX(J)
74 C      ELX(N,1) = LX(J)
75 C
76 C      ID(N,1) = ITYPE(J)
77 C      4 CONTINUE
78 C      DETERMINE PIECE WIDTH: NUMBER OF DEFECTS
79 C      FLY(26,1) = LFTLOC
80 C      NODEFC(1) = N
81 C
82 C      GRADE LEFT (LOWER) PIECE
83 C
84 C      CALL, GRADE (GRADE)
85 C
86 C      RESTORE TYPE, LOCATION, AND NUMBER OF DEFECTS: BOARD WIDTH
87 C
88 C      5 ID(6,J)=1,22
89 C      1 ID(J,1) = ITYPE(J)
90 C      1 FLY(J,1) = LY(J)
91 C      1 FUY(J,1) = UY(J)
92 C      1 FLX(J,1) = LX(J)
93 C      1 FUX(J,1) = UX(J)
94 C
95 C      6 CONTINUE
96 C      NODEFC(1) = M
97 C      FLY(26,1) = WIDTH
98 C
99 C      RETURN
100 C
**** STATEMENT NUMBERS ****
1 25 42 43 *58
2 36 *63
3 45 69 78 *77
4 53 63 *96
5 62 90 *96
6 63 96
**** VARIABLES ****
12 27 *49 *75 *94
13 12 19 *46 *54 *73 *81 *92 *98
14 10 12 25 *47 *74 *95
15 10 12 26 *45 *72 *93
16 51 86
17 6 12
18 94 95
19 12 24 *49 *76 *91
20 10 *27 48 75 94
21 10 *28 46 78 73 92
22 11 *30 42 63 97
23 11 *41 *44 45 46 47 48 49 55 *68 *71 72 73
24 74 75 76 *22 *55 *82 *97
25 8 12 32 *55 *82 *97
26 5 8 36 59
27 12 13 45 46 54
28 12 13 45 46 54
29 12 13 45 46 54
30 12 13 45 46 54
31 12 13 45 46 54
32 12 13 45 46 54
33 12 13 45 46 54
34 12 13 45 46 54
35 12 13 45 46 54
36 12 13 45 46 54
37 12 13 45 46 54
38 12 13 45 46 54
39 12 13 45 46 54
40 12 13 45 46 54
41 12 13 45 46 54
42 12 13 45 46 54
43 12 13 45 46 54
44 12 13 45 46 54
45 12 13 45 46 54
46 12 13 45 46 54
47 12 13 45 46 54
48 12 13 45 46 54
49 12 13 45 46 54
50 12 13 45 46 54
51 12 13 45 46 54
52 12 13 45 46 54
53 12 13 45 46 54
54 12 13 45 46 54
55 12 13 45 46 54
56 12 13 45 46 54
57 12 13 45 46 54
58 12 13 45 46 54
59 12 13 45 46 54
60 12 13 45 46 54
61 12 13 45 46 54
62 12 13 45 46 54
63 12 13 45 46 54
64 12 13 45 46 54
65 12 13 45 46 54
66 12 13 45 46 54
67 12 13 45 46 54
68 12 13 45 46 54
69 12 13 45 46 54
70 12 13 45 46 54
71 12 13 45 46 54
72 12 13 45 46 54
73 12 13 45 46 54
74 12 13 45 46 54
75 12 13 45 46 54

```

2.5-57-9/80

Function DEG

Emission RAD

THE JOURNAL OF CONVENTION AND ACTIVISM IN THE U.S. 1970-2000