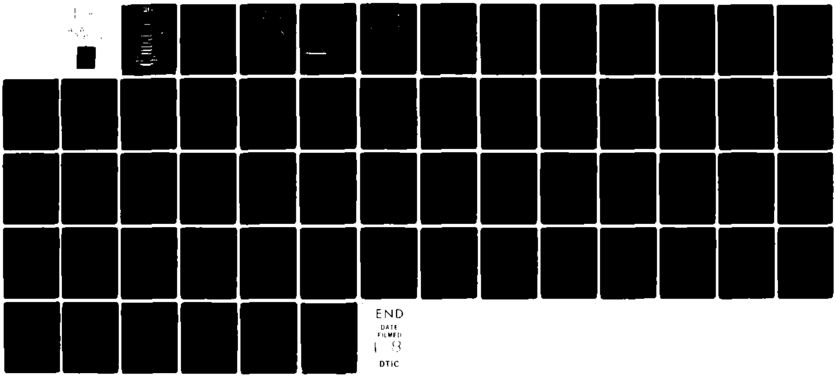
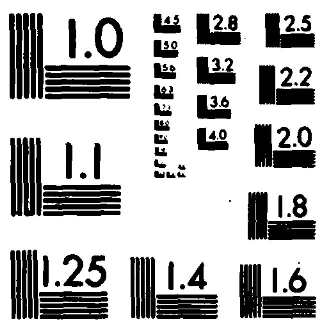
AD-A091 812 FOREST PRODUCTS LAB MADISON WI F/G 13/8
PROGRAMS FOR COMPUTER SIMULATION OF HARDWOOD LOG SAWING.(U)
SEP 80 W K ADKINS, D B RICHARDS, D W LEWIS
UNCLASSIFIED FSRP-FPL-357 NL



END
DATE
FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

United States
Department of
Agriculture

Forest Service

Forest
Products
Laboratory

Research
Paper

FPL 357

September 1980

Programs for Computer Simulation of Hardwood Log Sawing

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 resaw 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

United States Department of Agriculture

Forest Service

Forest Products Laboratory[1]

Research Paper

FPL 357

Programs for Computer Simulation of Hardwood Log Sawing

By
W. K. ADKINS,
D. B. RICHARDS[2]
D. W. LEWIS, Forest Products Technologist
and
E. H. BULGRIN, Supervisory Forest Products Technologist

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 rerip 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 rerip 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

[1] Maintained at Madison, Wis., in cooperation with the University of Wisconsin.

[2] Former graduate student, Dept. of Computer Science, and Professor of Forestry, Dept. of Forestry, University of Kentucky, Lexington, Ky.

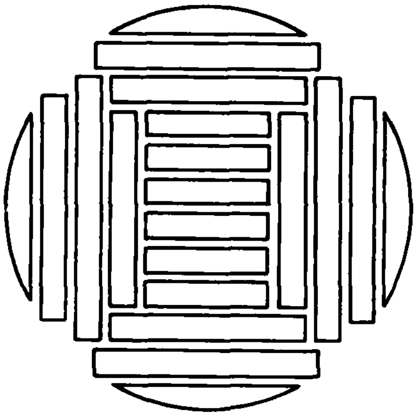


Figure 1.—End view of a quadrant sawn (QUAD) log.
(M 148 324)

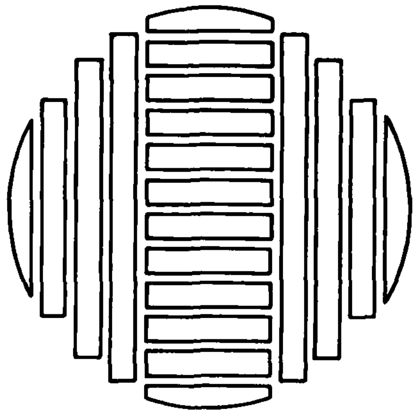


Figure 2.—End view of a cant sawn (CANT) log.
(M 148 326)

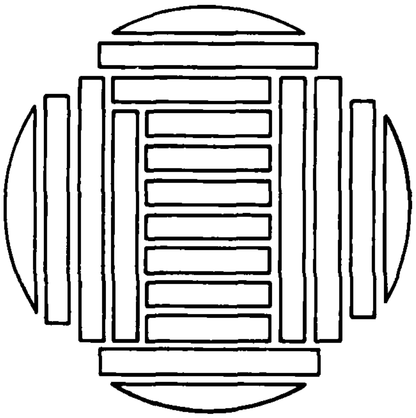


Figure 3.—End view of a decision sawn (DECID) log.
(M 148 325)

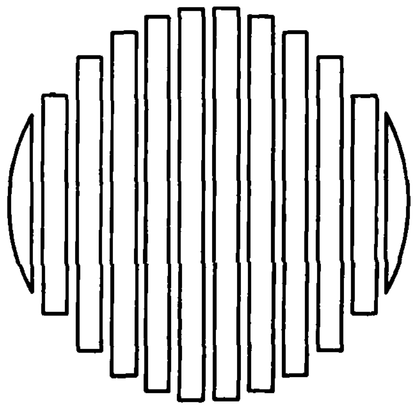


Figure 4.—End view of a live sawn (LIVE) log.
(M 148 327)

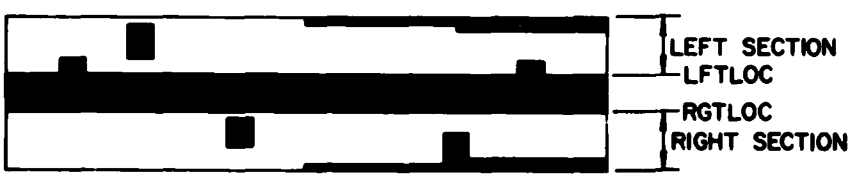


Figure 5.—Outer surface of a board that has been reripped for grade showing the coordinate system of the resulting three board sections as well as the truncation of any knots at the edge of a section.
M 148

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).

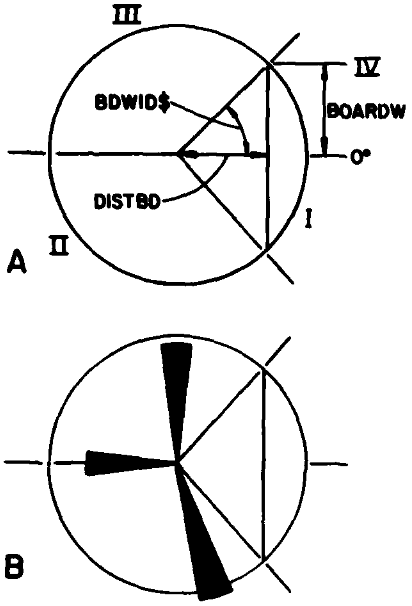
In the live rip method the log is sawed as in LIVE but the outer face of each board is then evaluated for defect type. If the central cylindrical core defect shows up on the outer face of the board, this defect is automatically ripped out and the boards produced are regraded and revalued (fig. 5). If 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

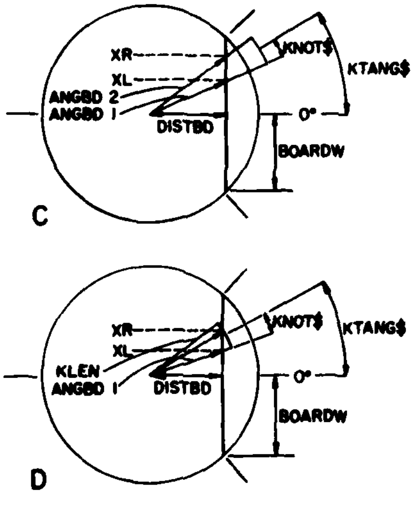
Log: A truncated cone with arbitrary length, small-end diameter, and taper. The length is measured in inches, the diameter in inches, and the taper in degrees.

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).

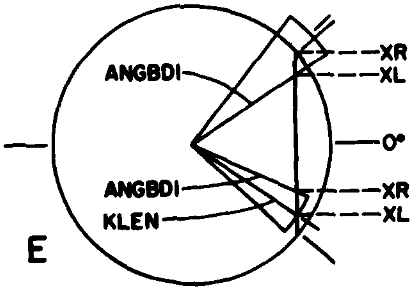
Core Defect: A solid cylinder extending the full log length, which may or may not be centered on the log axis. Core diameter and linear offset from the log axis are measured in inches; angular offset



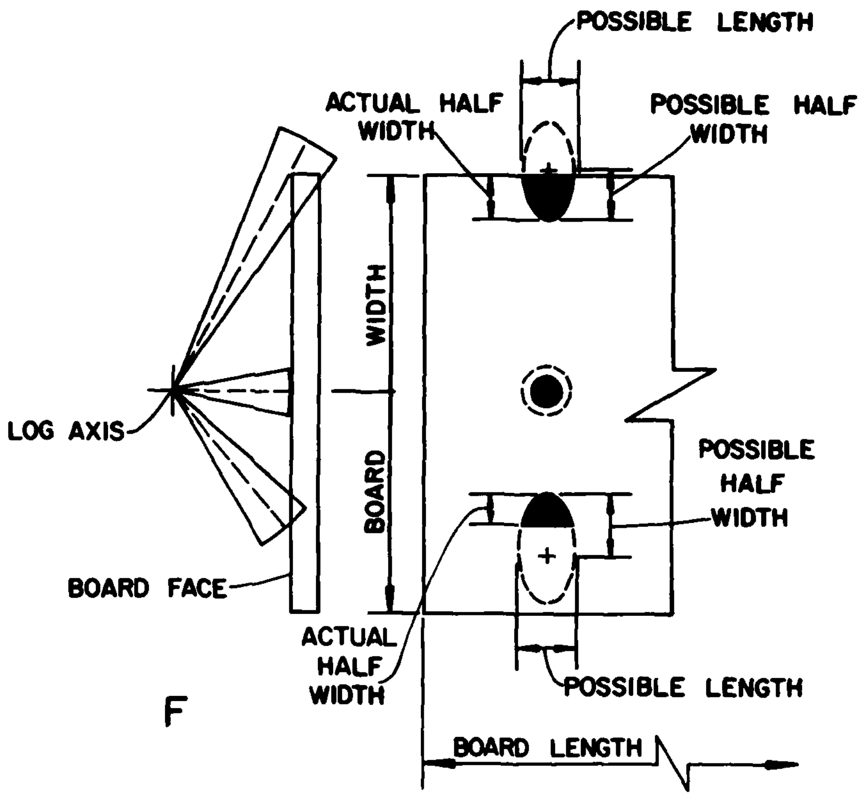
M 148 317



M 148 318



M 148 319



M 148 329

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

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 ¼-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 ≥ 3 inches at midlength; some width ≥ 2.5 inches at the top; some length ≥ 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 ¼-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 ¼ inch. The next wane defect begins at the point the last one ended and extends to the next point of ¼-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° 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

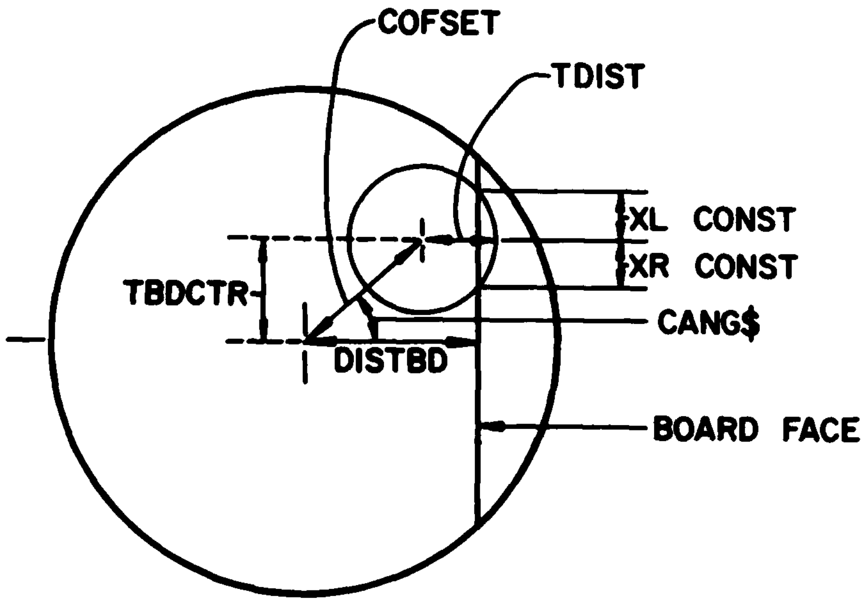


Figure 7.—A log section showing the core defect displaced from the central axis of the log.
(M 148 320)

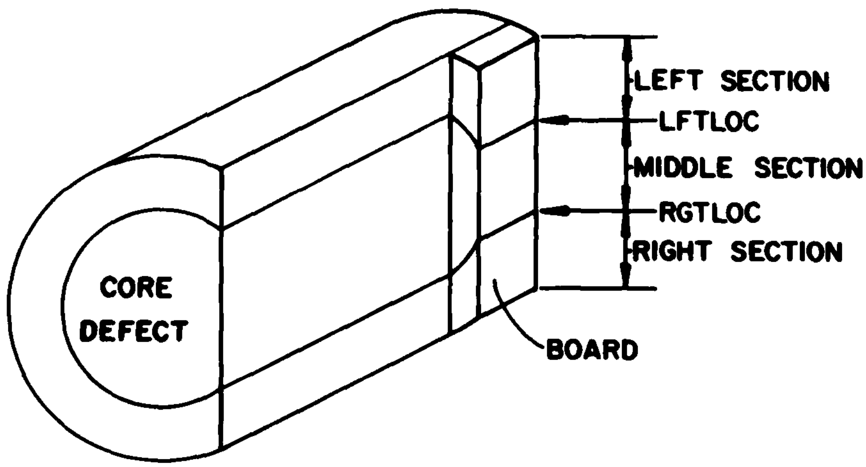


Figure 8.—Cross section of a log showing how the core defect is ripped out of a board if it appears on the outer face of a board. Although the ripping kerf is not shown, it is always taken out of the defective middle section.
(M 148 321)

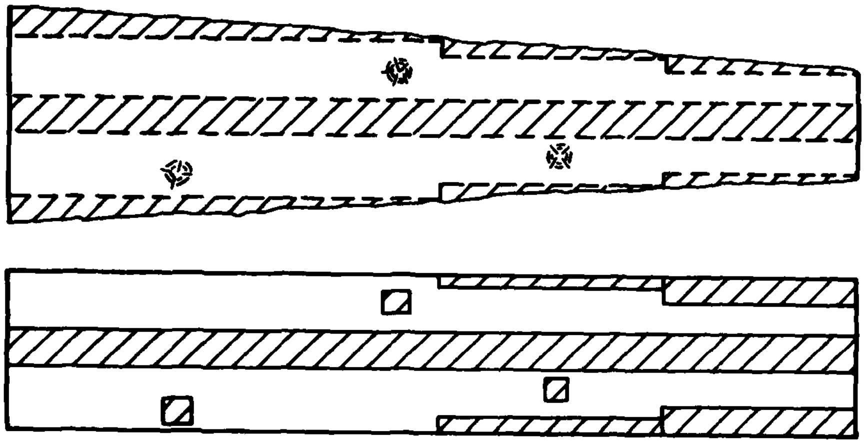


Figure 9.—The manner in which the computer would take the unedged board with knots (top view) and model it as an edged board with rectangular wane, knot, and core defects (bottom view).
(M 148 382)

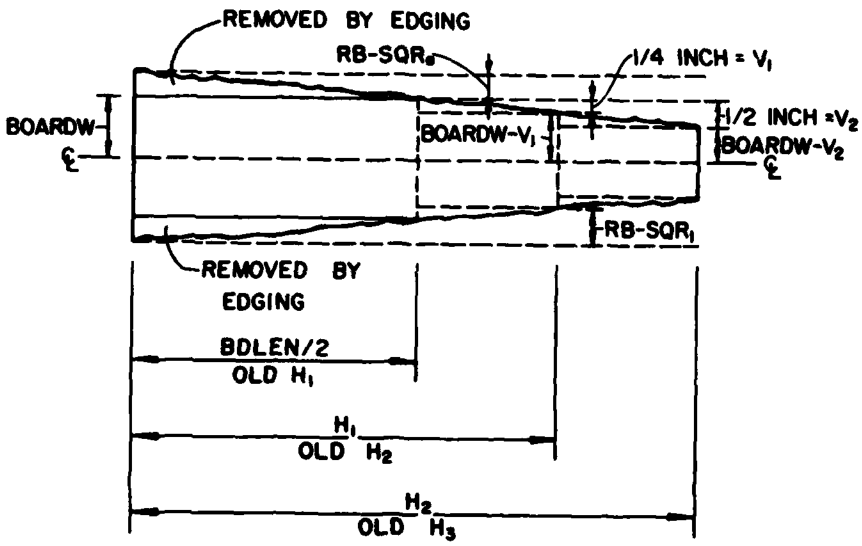


Figure 10.—The outer face of a waney board showing how subroutine WANE edges the board and puts in the wane defects.
(M 148 323)

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[1]	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

[1] 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 (IC), 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)
AVG$	average log value obtained over the 12 rotational positions on the carriage ($)
AVRIP (L):	average percent of available log surface measure realized under rerip over the 12 rotational positions on the carriage
AVRIP$ (L):	average log value obtained under rerip over the 12 rotational positions on the carriage ($)
AVTOTS (D):	average percent of available log surface measure realized over the 12 rotational positions
BDLEN:	current board length (in.) (fig. 10)
BDWID$:	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 (¼-in. units)
FLY:	array which contains all defect lower Y-coordinates on the current board face (¼-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 (¼-in. units)

FUY:	array which contains all defect upper Y-coordinates on the current board face (¼-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 (¼-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 (%)
HIRIP$ (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$:	angle of the current knot with respect to a line perpendicular to the current board face (degrees) (figs. 6c, 6d)
KTLEN:	array which contains the knot lengths (in.) (user supplied)
L:	log length (in.) (user supplied)
LFTLOC (L):	current board left rerip location (¼-in.) (figs. 5, 8)
LGRADE (L):	grade of the current board face left rerip section
LOG$:	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^2)
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 (¼-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 (ft^2)

PERCRP (L): total surface measure in each grade realized by reripping at the current rotational position on the carriage (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 (¼-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 (ft^2)

RPTOTS (L): total surface measure realized by reripping at the current rotational position on the carriage (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 (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 (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 (ft^2)

SMMID (L): surface measure of the current center rerip section (ft^2)

SMRGT (L): surface measure of the current right rerip section (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 (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)

THRB: 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 (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 (¼-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 (¼-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 (¼-in. units) (fig. 6d, 6e, 7)

XR: current defect upper X-coordinate on the current board face (¼-in. units) (fig. 6d, 6e, 7)

YB: current defect lower Y-coordinate on the current board face (¼-in. units)

YT: current defect upper Y-coordinate on the current board face (¼-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

Quadrant Sawing (QUAD)

```
C     'QUAD' IS A BALANCED METHOD OF SAWING AROUND A CENTRAL SQUARE CANT    QUAD
C     CONTAINING EXACTLY FOUR BOARDS.  THE PROGRAM CUTS FROM THE CENTRAL    QUAD
C     CANT OUTWARD AND COMPLETES ONE QUADRANT BEFORE PROGRESSING TO         QUAD
C     ANOTHER                                                               QUAD
C                                                                           QUAD
C                                                                           QUAD
        IMPLICIT REAL (K,L)                                                 QUAD
        REAL FLX(26,2),FLY(26,2),FUX(26,2),FUY(26,2),PERC(5),ANGLE(40),     QUAD
     .       HEIGHT(40),KTLEN(40),DAT*8                                     QUAD
        INTEGER ID(22,2),NODEFC(2),OLD,F                                    QUAD
        COMMON /DEFEC/ NODEFC,FLX,FLY,FUX,FUY,ID                            QUAD
     .         /BOARD/ L,D,K,T,LOGS,RB,DISTBD,BDLEN,BOARDW,KNOTS,R,BF       QUAD
     .         /PRICE/ FAS,SEL,ONEC,TWOC,THRB                               QUAD
C                                                                           QUAD
C                                                                           QUAD
C     THIS PROGRAM ASSUMES INPUT FROM UNIT 5 AND OUTPUTS TO UNITS 6 AND 8   QUAD
C                                                                           QUAD
C     GET LOG DIAMETER (IN.), LOG LENGTH (IN.), KERF SIZE (IN.), AND        QUAD
C     BOARD THICKNESS (IN.)                                                 QUAD
C                                                                           QUAD
      1 READ (5,100,END=17) D                                               QUAD
        READ (5,100) L                                                      QUAD
        READ (5,101) K                                                      QUAD
        READ (5,100) T                                                      QUAD
C                                                                           QUAD
C     GET KNOT HALF-ANGLE (DEGREES) AND LOG HALF-TAPER (DEGREES)            QUAD
C                                                                           QUAD
        READ (5,100) KNOTS                                                  QUAD
        READ (5,100) LOGS                                                   QUAD
C                                                                           QUAD
C     GET CORE DEFECT PARAMETERS: DIAMETER (IN.), LINEAR OFFSET (IN.),      QUAD
C     AND ANGULAR OFFSET (DEGREES)                                          QUAD
C                                                                           QUAD
        READ (5,116) CORDIA                                                 QUAD
        READ (5,116) COFSET                                                 QUAD
        READ (5,116) CANGS                                                  QUAD
C                                                                           QUAD
C     GET PRICE PER BOARD FOOT FOR EACH GRADE; NUMBER OF KNOTS              QUAD
C                                                                           QUAD
        READ (5,103) FAS,SEL,ONEC,TWOC,THRB                                 QUAD
        READ (5,119) MAX                                                    QUAD
C                                                                           QUAD
C     GET DATE AND CALCULATE RADIUS OF CORE DEFECT                          QUAD
C                                                                           QUAD
        CALL DATE (DAT)                                                     QUAD
        CRADUS = CORDIA/2.                                                  QUAD
C                                                                           QUAD
C     GET KNOT ANGLES, HEIGHTS, AND LENGTHS                                 QUAD
C                                                                           QUAD
        DO 2 J=1,MAX                                                        QUAD
          READ (5,117) ANGLE(J),HEIGHT(J)                                   QUAD
      2 CONTINUE                                                            QUAD
        DO 3 J=1,MAX                                                        QUAD
          READ (5,116) KTLEN(J)                                             QUAD
      3 CONTINUE                                                            QUAD
C                                                                           QUAD
C     CALCULATE LOG RADIUS AT TOP AND BOTTOM; TOTAL LOG CUBIC FEET          QUAD
C                                                                           QUAD
        RI = D/2.                                                           QUAD
        RB = RI+L*TAN(RAD(LOGS))                                            QUAD
        LVOL = 1/3.*3.141593*(RB**2+RB*RI+RI**2)*L/144.                     QUAD
C                                                                           QUAD
C     OUTPUT HEADER                                                         QUAD
C                                                                           QUAD
        WRITE (6,104) DAT                                                   QUAD
        WRITE (6,105) L,D,LOGS,LVOL                                         QUAD
        WRITE (6,119) MAX                                                   QUAD
        DO 4 J=1,MAX                                                        QUAD
          WRITE (6,115) ANGLE(J),HEIGHT(J),KTLEN(J)                         QUAD
      4 CONTINUE                                                            QUAD
        WRITE (6,106) KNOTS                                                 QUAD
        WRITE (6,102) CRADUS,COFSET,CANGS                                   QUAD
        WRITE (6,107) K,T                                                   QUAD
        WRITE (6,108) FAS,SEL,ONEC,TWOC,THRB                                QUAD
        WRITE (8,200) L,D,MAX,CRADUS,COFSET,CANGS,LVOL,K,T,KNOTS,LOGS       QUAD
        WRITE (8,201) (KTLEN(J),J=1,MAX)                                    QUAD
        WRITE (8,201) (HEIGHT(J),J=1,MAX)                                   QUAD
        WRITE (8,201) (ANGLE(J),J=1,MAX)                                    QUAD
        WRITE (8,202) FAS,SEL,ONEC,TWOC,THRB                                QUAD
C                                                                           QUAD
C     INITIALIZE VARIABLES FOR CALCULATING HIGH, LOW, AND AVERAGE           QUAD
C     YIELDS FOR EACH LOG                                                   QUAD
C                                                                           QUAD
        HIGHS = 0.                                                          QUAD
        SMALLS = 999.                                                       QUAD
        AVGS = 0.                                                           QUAD
C                                                                           QUAD
C                                                                           QUAD
C     SAW LOG IN 12 DIFFERENT ROTATIONAL POSITIONS, 15 DEGREES APART, TO    QUAD
C     DETERMINE THE ORIENTATION GIVING THE HIGHEST YIELD                    QUAD
C                                                                           QUAD
        DO 16 INX=1,12                                                      QUAD
C                                                                           QUAD
C     INITIALIZE FOR EACH POSITION                                          QUAD
C                                                                           QUAD
          BF = 0.0                                                          QUAD
          TVAL = 0.0                                                        QUAD
          NBD = 0                                                           QUAD
          DO 5 J=1,5                                                        QUAD
            PERC(J) = 0.0                                                   QUAD
      5   CONTINUE                                                          QUAD
C                                                                           QUAD
C     FOR FOUR SIDES OF THE LOG...                                          QUAD
C                                                                           QUAD
          DO 13 F=1,4                                                       QUAD
C                                                                           QUAD
C     INITIALIZE PORTION OF LOG TO BE CUT... THIS SETS THE DEPTH OF THE     QUAD
C     FIRST CUT.  THE SLAB FACES ARE CUT TO A FOUR BOARD CANT               QUAD
C                                                                           QUAD
            IF (F.EQ.1.OR.F.EQ.3) R = RB-2.*T-1.5*K                         QUAD
C                                                                           QUAD
C     THE CANT FACES ARE KERF CENTERED TO DIVIDE THE CANT                   QUAD
C                                                                           QUAD
            IF (F.EQ.2.OR.F.EQ.4) R = RB+K/2.                               QUAD
C                                                                           QUAD
C     CUT BOARDS FROM THIS FACE UNTIL FACE IS COMPLETELY CUT.  THE LOG IS   QUAD
C     CUT FROM THE INSIDE OUT                                               QUAD
C                                                                           QUAD
      6     CALL KERFQ (&11,NBD,F)                                          QUAD
C                                                                           QUAD
C     FOR BOTH SIDES OF THE BOARD...                                        QUAD
C                                                                           QUAD
            DO 10 I=1,2                                                     QUAD
C                                                                           QUAD
C     INITIALIZE DEFECT ARRAYS, EXCEPT F..(26,1), WHICH HOLDS THE BOARD     QUAD
C     DIMENSIONS                                                            QUAD
C                                                                           QUAD
              DO 7 J=1,25                                                   QUAD
                FLX(J,I) = -9999.0                                          QUAD
                FUX(J,I) = -9999.0                                          QUAD
                FLY(J,I) = -9999.0                                          QUAD
                FUY(J,I) = -9999.0                                          QUAD
      7       CONTINUE                                                      QUAD
              DO 8 J=1,22                                                   QUAD
                ID(J,I) = 0                                                 QUAD
      8       CONTINUE                                                      QUAD
              NODEFC(I) = 0                                                 QUAD
C                                                                           QUAD
C     PUT IN WANE DEFECTS                                                   QUAD
C                                                                           QUAD
              CALL WANEQ                                                    QUAD
C                                                                           QUAD
C     PUT IN KNOTS                                                          QUAD
C                                                                           QUAD
```

```
145           DO 9 J=1,MAX                                                        QUAD
146             CALL KNOTD (HEIGHT(J),KTLEN(J),ANGLE(J))                          QUAD
147     9       CONTINUE                                                          QUAD
148 C                                                                             QUAD
149 C   PUT IN CORE DEFECT                                                        QUAD
150 C                                                                             QUAD
151           CALL COPED (CRADUS,COFSET,CANGS)                                    QUAD
152 C                                                                             QUAD
153 C   AND GRADE THE FACE                                                        QUAD
154 C                                                                             QUAD
155           CALL GRADE (NPG)                                                    QUAD
156 C                                                                             QUAD
157 C   SAVE GRADE OF FIRST SIDE OF BOARD                                         QUAD
158 C                                                                             QUAD
159           IF (I.EQ.1) OLD = NPG                                               QUAD
160 C                                                                             QUAD
161 C   SUBTRACT BOARD THICKNESS AND PROCESS THE INNER FACE                       QUAD
162 C                                                                             QUAD
163           DISTBD = DISTBD-T                                                   QUAD
164    10     CONTINUE                                                            QUAD
165 C                                                                             QUAD
166 C   DETERMINE BOARD GRADE AND VALUE BASED ON GRADES OF BOTH FACES             QUAD
167 C                                                                             QUAD
168         CALL PRICED (OLD,NPG,TVAL,PERC)                                       QUAD
169 C                                                                             QUAD
170 C   CUT ANOTHER BOARD                                                         QUAD
171 C                                                                             QUAD
172         GO TO 6                                                               QUAD
173 C                                                                             QUAD
174 C   TURN LOG 90 DEGREES TO CUT NEXT FACE                                      QUAD
175 C                                                                             QUAD
176    11   DO 12 J=1,MAX                                                         QUAD
177           ANGLE(J) = AMOD(ANGLE(J)+090.0,360.0)                               QUAD
178    12   CONTINUE                                                              QUAD
179         CANGS = AMOD(CANGS+90.0,360.0)                                        QUAD
180         NBD = 0                                                               QUAD
181    13   CONTINUE                                                              QUAD
182 C                                                                             QUAD
183 C   PRINT RESULTS FOR THIS POSITION                                           QUAD
184 C                                                                             QUAD
185       IROTAT = INX*15-15                                                      QUAD
186       WRITE (6,114) IROTAT                                                    QUAD
187       WRITE (6,113) (PERC(J),J=1,5)                                           QUAD
188       TOTS = PERC(1)+PERC(2)+PERC(3)+PERC(4)+PERC(5)                          QUAD
189       WRITE (6,112) TOTS                                                      QUAD
190       WRITE (8,203) IROTAT                                                    QUAD
191       WRITE (8,204) (PERC(J),J=1,5)                                           QUAD
192       WRITE (8,204) TOTS                                                      QUAD
193       DO 14 J=1,5                                                             QUAD
194         PERC(J) = (PERC(J)/LVOL)*100.                                         QUAD
195    14 CONTINUE                                                                QUAD
196       TOTS = (TOTS/LVOL)*100.                                                 QUAD
197       WRITE (6,111) TOTS                                                      QUAD
198       WRITE (6,110) (ANGLE(J),J=1,MAX)                                        QUAD
199       WRITE (6,120) (PERC(J),J=1,5),TVAL                                      QUAD
200       WRITE (8,205) (PERC(J),J=1,5)                                           QUAD
201       WRITE (8,204) TVAL,TOTS                                                 QUAD
202 C                                                                             QUAD
203 C   SAVE HIGH, LOW, AND AVERAGE YIELDS FOR THIS LOG                           QUAD
204 C                                                                             QUAD
205       AVGS = AVGS+TVAL                                                        QUAD
206       IF (TVAL.GT.HIGHS) HIGHS = TVAL                                         QUAD
207       IF (TVAL.LT.SMALLS) SMALLS = TVAL                                       QUAD
208 C                                                                             QUAD
209 C   ROTATE LOG BY 15 DEGREES AND REPROCESS                                    QUAD
210 C                                                                             QUAD
211       DO 15 J=1,MAX                                                           QUAD
212         ANGLE(J) = AMOD(ANGLE(J)+15.0,360.0)                                  QUAD
213    15 CONTINUE                                                                QUAD
214       CANGS = AMOD(CANGS+15.0,360.0)                                          QUAD
215    16 CONTINUE                                                                QUAD
216 C                                                                             QUAD
217 C                                                                             QUAD
218 C   PRINT HIGH, LOW, AND AVERAGE YIELDS FOR THIS LOG                          QUAD
219 C                                                                             QUAD
220     AVGS = AVGS/12.                                                           QUAD
221     WRITE(6,103) HIGHS,AVGS,SMALLS                                            QUAD
222 C                                                                             QUAD
223 C   LOG COMPLETELY PROCESSED, READ PARAMETERS FOR NEXT LOG                    QUAD
224 C                                                                             QUAD
225       GO TO 1                                                                 QUAD
226 C                                                                             QUAD
227    17 STOP                                                                    QUAD
228 C                                                                             QUAD
229 C   FORMATS FOR DEVICE 6 (PRINTER)                                            QUAD
230 C                                                                             QUAD
231   100 FORMAT (F5.1)                                                           QUAD
232   101 FORMAT (F5.3)                                                           QUAD
233   102 FORMAT (' CORE DEFECT RADIUS ',F7.2,'LINEAR OFFSET ',F7.2,              QUAD
234      . 'ANGULAR OFFSET ',F7.2)                                                QUAD
235   103 FORMAT (' $ YIELD: HIGH ',F6.2,'  AVERAGE ',F6.2,'  LOW ',F6.2)         QUAD
236   104 FORMAT (' QUADRANT SAWING METHOD ',A8)                                  QUAD
237   105 FORMAT (' LOG PARAMETERS:  LENGTH',F6.1,'  DIA',F6.1,'  TAPER',         QUAD
238      . F6.4,' DEGREES.  LOG VOLUME:',F8.2,' CUBIC FEET')                      QUAD
239   106 FORMAT (' ANGLES MEASURED FROM ZERO DEGREES = EAST FOR A ',             QUAD
240      . 'VERTICALLY CUTTING SAW.  KNOT TAPER',F6.2)                            QUAD
241   107 FORMAT (' CUTTING PARAMETERS:  KERF',F6.4,'  BOARD THICKNESS',          QUAD
242      . F5.2)                                                                  QUAD
243   108 FORMAT (' PRICES PER BOARD FOOT: ',5F10.4)                              QUAD
244   109 FORMAT (5F10.4)                                                         QUAD
245   110 FORMAT (' KNOT ANGLES: ',15F7.2,2(/,13X,15F7.2))                        QUAD
246   111 FORMAT (' TOTAL PERCENT YIELD IS ',F5.2)                                QUAD
247   112 FORMAT (' TOTAL SURFACE MEASURE IS',F8.2)                               QUAD
248   113 FORMAT (' SURFACE MEASURE PER GRADE IS ',5F7.2)                         QUAD
249   114 FORMAT (2(/),'  ROTATION IS ',I4)                                       QUAD
250   115 FORMAT (3F7.2)                                                          QUAD
251   116 FORMAT (F6.3)                                                           QUAD
252   117 FORMAT (2F6.2)                                                          QUAD
253   118 FORMAT (I2)                                                             QUAD
254   119 FORMAT ('1THERE ARE',I3,' KNOTS IN THIS LOG. THEY ARE'                  QUAD
255      . 1X,'ANGLE  HEIGHT  LENGTH')                                            QUAD
256   120 FORMAT (' % AVAILABLE SURFACE MEASURE IN EACH GRADE ',5F5.1,            QUAD
257      .    LOG VALUE IS $',F6.2)                                               QUAD
258 C                                                                             QUAD
259 C   FORMATS FOR DEVICE 8 (STORAGE MEDIUM)                                     QUAD
260 C                                                                             QUAD
261   200 FORMAT (F6.1,F5.1,I3,3F5.1,F8.2,F6.3,F4.1,F6.2,F5.3,F6.3)               QUAD
262   201 FORMAT (3(10F7.2/))                                                     QUAD
263   202 FORMAT (5F7.4)                                                          QUAD
264   203 FORMAT (I3)                                                             QUAD
265   204 FORMAT (5F8.2)                                                          QUAD
266   205 FORMAT (10F6.2)                                                         QUAD
267       END                                                                     QUAD
```

*** STATEMENT NUMBERS ***

1	*21	225
2	58	*52
3	53	*55
4	68	*70
5	99	*101
6	*119	172
7	128	*133
8	134	*136
9	145	*147
10	123	*164
11	119	*176
12	176	*178
13	105	*181
14	193	*195
15	211	*213
16	92	*215
17	21	*227

* Indicates line at which symbol or number is defined

```
100     21    22    24    28    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     40  *244
110    198  *245
111    197  *246
112    189  *247
113    187  *248
114    186  *249
115     69  *250
116     34    35    36    54  *251
117     51  *252
118     41  *253
119     67  *254
120    199  *256
200     75  *261
201     76    77    78  *262
202     79  *263
203    190  *264
204    191   192   201  *265
205    200  *266
```

*** VARIABLES ***

```
AMOD     177   179   212   214   146  *177   198  *212
ANGLE      8   *51    69    78
AVGS     *86  *205  *220   221
BDLEN     11
BF        11   *96
BOARDW    11
CANGS    *36    72    75   151  *179  *214
COFSET   *35    72    75   151
CORDIA   *34    46
COPEO    151
CRAIUS   *46    72    75   151
D         11   *21    59    66    75
DAT        8    45    65
DATE      45
DISTBD    11  *103
F         10  *105   110   114   119
FAS       11   *40    74    79
FLX        8    11  *129
FLY        8    11  *131
FUX        8    11  *130
FUY        8    11  *132
GRADE    155
HEIGHT     8   *51    69    77   146
HIGHS    *84  *206   221
I       *123   159
IN        10    11  *135
INX      *92   105
IROTAT  *185   186   190
J        *50   *51   *53   *54   *68    69    76    77    78   *99   100  *128
         129   130   131   132  *134   135  *145   146  *176   177   187   191
        *193   194   196   199   200  *211   212
K         11   *23    73    75   110   114
KERFO    119
KNOTS     11   *28    71    75
KNOTO    146
KTLEN      8   *54    69    76   146
L         11   *22    60    61    66    75
LOGS      11   *29    60    66    75
LVOL     *61    66    75   194   196
MAX      *41    50    53    67    68    75    76    77    78   145   176   198
         211
```

```
NBD      *98   119  *100
NODEFC    10    11  *137
NPG      155   159   168
OLD       10  *159   168
ONEC      11   *40    74    79
PERC       8  *100   168   187   188   191  *194   199   200
PRICED   168
R         11  *110  *114
RAD       60
RB        11   *60    61   110   114
PI       *59    60    61
SEL       11   *40    74    79
SMALL$   *85  *207   221
T         11   *24    73    75   110   163
TAN       60
THRB      11   *40    74    79
TOTS    *188   189   192  *196   197   201
TVAL     *97   168   199   201   205   206   207
TWOC      11   *40    74    79
WANEO    141
```

Subroutine KERF

```
 1 C   SUBROUTINE 'KERF' CUTS A BOARD FROM THE LOG.  THE BOARD WIDTH IS     KERFO
 2 C   DEFINED AT THE CENTER. AND THE BOARD LENGTH FROM THE BOTTOM          KERFO
 3 C   OF THE LOG                                                           KERFO
 4 C                                                                        KERFO
 5 C                                                                        KERFO
 6       SUBROUTINE KERFO (*,NBD,F)                                         KERFO
 7 C                                                                        KERFO
 8       IMPLICIT REAL (K,L)                                                KERFO
 9       REAL FLX(26,2),FLY(26,2),FUX(26,2),FUY(26,2)                       KERFO
10       INTEGER ID(22,2),NODEFC(2),F                                       KERFO
11       COMMON /DEFEC/ NODEFC,FLX,FLY,FUX,FUY,ID                           KERFO
12               /BOARD/ L,D,K,T,LOGS,RB,DISTBD,BDLEN,BOARDW,KNOTS,R,BF     KERFO
13 C                                                                        KERFO
14 C                                                                        KERFO
15 C   ALL BOARDS MUST BE AT LEAST 48 INCHES LONG, 2.5 INCHES WIDE AT       KERFO
16 C   THE TOP, AND 3 INCHES WIDE AT THE CENTER.  IN ADDITION, THE TOTAL    KERFO
17 C   AMOUNT OF WANE FROM THE CENTER TO THE TOP IS LIMITED TO 4 INCHES.    KERFO
18 C   THE BOARD IS CUT BACK IN ONE FOOT INCREMENTS IF THE WIDTH OR WANE    KERFO
19 C   TESTS FAIL.  IF THE LENGTH TEST FAILS, ANOTHER BOARD IS CUT FROM     KERFO
20 C   THIS FACE                                                            KERFO
21 C                                                                        KERFO
22 C   FIND OUTER FACE OF BOARD                                             KERFO
23 C                                                                        KERFO
24       RI = D/2.                                                          KERFO
25     1 R = R-T-K                                                          KERFO
26       NBD = NBD+1                                                        KERFO
27       BDLEN = L                                                          KERFO
28 C                                                                        KERFO
29 C   IF THIS FACE OF THE LOG IS COMPLETELY CUT, EXIT                      KERFO
30 C                                                                        KERFO
31       IF (R.LT.0.) RETURN1                                               KERFO
32       DISTBD = RB-R                                                      KERFO
33       WMB = DISTBD-T-K                                                   KERFO
34       IF (F.EQ.1.OR.F.EQ.3) WMB = DISTBD                                 KERFO
35 C                                                                        KERFO
36 C   THE CENTRAL CANT HAS BEEN DEFINED AS FOUR BOARDS SQUARE              KERFO
37 C                                                                        KERFO
38       IF (NBD.LT.3.AND.(F.EQ.2.OR.F.EQ.4)) WMB = 2.*T+1.5*K              KERFO
39       POSLEN = R*TAN(RAD(90.-LOGS))                                      KERFO
40 C                                                                        KERFO
41 C   CHECK FOR BOARDS SHORTER THAN THE LOG DUE TO LOG TAPER               KERFO
42 C                                                                        KERFO
43       IF (POSLEN.LT.BDLEN) BDLEN = POSLEN                                KERFO
44 C                                                                        KERFO
45 C   BOARDS MUST BE AT LEAST 4 FEET LONG                                  KERFO
46 C                                                                        KERFO
47     2 IF (BDLEN.LT.48.) GO TO 1                                          KERFO
48 C                                                                        KERFO
49 C   DETERMINE HALF-WIDTH AT MID-LENGTH OF BOARD                          KERFO
50 C                                                                        KERFO
51       RM = RI+(L-BDLEN/2.)*TAN(RAD(LOGS))                                KERFO
52       IF (DISTBD.GT.RM) GO TO 3                                          KERFO
53       BOARDW = SQRT(RM**2-DISTBD**2)                                     KERFO
54       IF (WMB.LT.BOARDW) BOARDW = WMB                                    KERFO
55 C                                                                        KERFO
56 C   BOARD MUST BE AT LEAST 3 INCHES WIDE AT MID-LENGTH                   KERFO
57 C                                                                        KERFO
58       IF (BOARDW.LT.1.5) GO TO 3                                         KERFO
59 C                                                                        KERFO
60 C   DETERMINE HALF-WIDTH AT TOP OF BOARD                                 KERFO
61 C                                                                        KERFO
62       RT = RI+(L-BDLEN)*TAN(RAD(LOGS))                                   KERFO
63       IF (DISTBD.GT.RT) GO TO 3                                          KERFO
64       WT = SQRT(RT**2-DISTBD**2)                                         KERFO
65       IF (WMB.LT.WT) WT = WMB                                            KERFO
66 C                                                                        KERFO
67 C   BOARD MUST BE AT LEAST 2.5 INCHES WIDE AT THE TOP                    KERFO
68 C                                                                        KERFO
69       IF (WT.LT.1.25) GO TO 3                                            KERFO
70 C                                                                        KERFO
71 C   LIMIT THE TOTAL WANE TO AVOID GENERATING MORE DEFECTS THAN THE       KERFO
72 C   GRADING PROGRAM CAN HANDLE.  TOTAL WIDTH OF WANE MUST BE LESS        KERFO
73 C   THAN 4 INCHES                                                        KERFO
74 C                                                                        KERFO
75       WAN = BOARDW-WT                                                    KERFO
76       IF (WAN.GT.2.) GO TO 3                                             KERFO
77 C                                                                        KERFO
78 C   CONVERT BOARD WIDTH AND LENGTH TO EVEN QUARTER INCHES                KERFO
79 C                                                                        KERFO
80       BOARDW = INT(BOARDW*8.)/8.                                         KERFO
81       BDLEN = INT(BDLEN*4.)/4.                                           KERFO
82 C                                                                        KERFO
83 C   DEFINE BOARD EDGES IN QUARTER INCH UNITS                             KERFO
84 C                                                                        KERFO
85       FLX(26,1) = BDLEN*4.                                               KERFO
86       FLY(26,1) = BOARDW*8.                                              KERFO
87       FUX(26,1) = 0.0                                                    KERFO
88       FUY(26,1) = 0.0                                                    KERFO
89 C                                                                        KERFO
90 C   CALCULATE UNROUNDED BOARD FEET                                       KERFO
91 C                                                                        KERFO
92       BF = BDLEN*BOARDW/2.*T/144.                                        KERFO
93       RETURN                                                             KERFO
94 C                                                                        KERFO
95 C   CUT BACK BOARD BY ONE FOOT                                           KERFO
96 C                                                                        KERFO
97     3 BDLEN = BDLEN-12.                                                  KERFO
98       GO TO 2                                                            KERFO
99       END                                                                KERFO
```

*** STATEMENT NUMBERS ***

1	*25	47				
2	*47	98				
3	52	58	63	69	76	*97

*** VARIABLES ***

BDLEN	11	*27	*43	47	51	62	*81	85	92	*97
BF	11	*92								
BOARDW	11	*53	*54	58	75	*80	86	92		
D	11	24								
DISTBD	11	*32	33	34	52	53	63	64		
F	6	10	34	38						
FLX	9	11	*85							
FLY	9	11	*86							
FUX	9	11	*87							
FUY	9	11	*88							
ID	10	11								
INT	80	81								
K	11	25	33	38						
KERFO	6									
KNOTS	11									
L	11	27	51	62						
LOGS	11	39	51	62						
NBD	6	*26	38							
NODEFC	10	11								
POSLEN	*39	43								
R	11	*25	31	32	39					
RAD	39	51	62							
RB	11	32								
RI	*24	51	62							
RM	*51	52	53							
RT	*62	63	64							
SQRT	53	64								
T	11	25	33	38	92					
TAN	39	51	62							
WAN	*75	76								
WMB	*33	*34	*38	54	65					
WT	*64	*65	69	75						

Subroutine WANE

```
 1 C   SUBROUTINE 'WANE' LOCATES WANE DEFECTS ON THE BOARD              WANEO
 2 C                                                                    WANEO
 3 C                                                                    WANEO
 4       SUBROUTINE WANEO                                               WANEO
 5 C                                                                    WANEO
 6 C   THE BOARD EDGE HAS A NEW WANE DEFECT IF THE WIDTH DROPS BY 1/2   WANEO
 7 C   INCH (1/4 INCH FOR EACH SIDE) FROM THAT AT THE MIDDLE.  THE BOARD WANEO
 8 C   IS SYMMETRICAL SO THE WANE DEFECTS OCCUR AT THE SAME PLACE ON    WANEO
 9 C   BOTH EDGES OF THE BOARD                                          WANEO
10 C                                                                    WANEO
11       IMPLICIT REAL (K,L)                                            WANEO
12       INTEGER ID(22,2),NODEFC(2)                                     WANEO
13       REAL FLX(26,2),FLY(26,2),FUX(26,2),FUY(26,2)                   WANEO
14       COMMON /BOARD/ L,D,K,T,LOGS,RB,DISTBD,BDLEN,BOARDW,KNOTS,R,BF  WANEO
15           /DEFEC/ NODEFC,FLX,FLY,FUX,FUY,ID                          WANEO
16 C                                                                    WANEO
17 C                                                                    WANEO
18       CON1 = TAN(RAD(LOGS))                                          WANEO
19       CON2 = DISTBD**2                                               WANEO
20       IF ((RB-SQRT(CON2+BOARDW**2))/CON1.GE.BDLEN) RETURN            WANEO
21       OLDH = BDLEN*2.                                                WANEO
22       V = 0.                                                         WANEO
23 C                                                                    WANEO
24 C   FIND THE PLACE ALONG THE BOARD WHERE THE HALF-WIDTH HAS DECREASED WANEO
25 C   BY 1/4 INCH.  THE DEFECT ENDS HERE AND EXTENDS FROM THE END OF THE WANEO
26 C   LAST WANE DEFECT (OR FROM THE MIDDLE OF THE BOARD)               WANEO
27 C                                                                    WANEO
28     1 V = V+0.25                                                     WANEO
29       SQR = SQRT(CON2+(BOARDW-V)**2)                                 WANEO
30       H = INT(AMIN1((RB-SQR)*4./CON1,BDLEN*4.))                      WANEO
31       NODEFC(1) = NODEFC(1)+1                                        WANEO
32       M = NODEFC(1)                                                  WANEO
33 C                                                                    WANEO
34 C   LIMIT NUMBER OF DEFECTS TO MAXIMUM ALLOWED BY GRADING PROGRAM (22) WANEO
35 C                                                                    WANEO
36       IF (M.GT.22) GO TO 2                                           WANEO
37 C                                                                    WANEO
38 C   SAVE WANE COORDINATES FOR GRADING PROGRAM                        WANEO
39 C                                                                    WANEO
40       FUY(M,1) = V*4.+1.                                             WANEO
41       FLY(M,1) = 0.                                                  WANEO
42       FUX(M,1) = H                                                   WANEO
43       FLX(M,1) = OLDH                                                WANEO
44       ID(M,1) = 5                                                    WANEO
45 C                                                                    WANEO
46 C   SINCE THE BOARD IS SYMMETRICAL, USE THE PREVIOUSLY GENERATED HEIGHT WANEO
47 C   FOR THE WANE DEFECTS ON THE OTHER EDGE OF THE BOARD              WANEO
48 C                                                                    WANEO
49     2 NODEFC(1) = NODEFC(1)+1                                        WANEO
50       M = NODEFC(1)                                                  WANEO
51       IF (M.GT.22) GO TO 3                                           WANEO
52       FLY(M,1) = BOARDW*8.-V*4.-1.                                   WANEO
53       FUY(M,1) = BOARDW*8.                                           WANEO
54       FUX(M,1) = H                                                   WANEO
55       FLX(M,1) = OLDH                                                WANEO
56       ID(M,1) = 5                                                    WANEO
57 C                                                                    WANEO
58 C   WHEN THE DEFECT EXTENDS TO THE END OF THE BOARD, EXIT            WANEO
59 C                                                                    WANEO
60     3 IF (H.GE.BDLEN*4.) RETURN                                      WANEO
61 C                                                                    WANEO
62 C   THE NEXT WANE DEFECT BEGINS WHERE THIS ONE ENDS                  WANEO
63 C                                                                    WANEO
64       OLDH = H                                                       WANEO
65       GO TO 1                                                        WANEO
66       RETURN                                                         WANEO
67       END                                                            WANEO
```

```
*** STATEMENT NUMBERS ***

    1      *28   65
    2       36  *49
    3       51  *60

*** VARIABLES ***

AMIN1     30
BDLEN     14   20   21   30   60
BF        14
BOARDW    14   20   29   52   53
CON1     *18   20   30
CON2     *19   20   29
D         14
DISTBD    14   19
FLX       13   14  *43  *55
FLY       13   14  *41  *52
FUX       13   14  *42  *54
FUY       13   14  *40  *53
H        *30   42   54   60   64
ID        12   14  *44  *56
INT       30
K         14
KNOTS     14
L         14
LOGS      14   18
M        *32   36   40   41   42   43   44  *50   51   52   53   54
          55   56
NODEFC    12   14  *31   32  *49   50
OLDH     *21   43   55  *64
R         14
RAD       18
RB        14   20   30
SQR      *29   30
SQRT      20   29
T         14
TAN       18
V        *22  *28   29   40   52
WANEO      4
```

Subroutine KNOT

```
C     SUBROUTINE 'KNOT' LOCATES THE KNOT DEFECTS ON THE BOARD FACE.
C     ANGLES ARE MEASURED CLOCKWISE WITH 0 DEGREES DEFINED AS THE
C     LINE FROM THE CENTER OF THE LOG PERPENDICULAR TO THE BOARD FACE
C
C
      SUBROUTINE KNOTO (KHIGH,KLEN,KTANG$)
C
      IMPLICIT REAL (K,L)
      INTEGER ID(22,2),NODEFC(2)
      REAL FLX(26,2),FLY(26,2),FUX(26,2),FUY(26,2)
      COMMON /DEFEC/ NODEFC,FLX,FLY,FUX,FUY,ID
     .       /BOARD/ L,D,K,T,LOG$,RB,DISTBD,BDLEN,BOARDW,KNOT$,R,BF
C
C
C     SEE IF THE KNOT IS LONG ENOUGH TO REACH THE BOARD FACE
C
      IF (KLEN.LT.DISTBD) RETURN
C
C     CALCULATE LOG RADIUS AT KNOT HEIGHT, BOARD HALF-WIDTH ANGLE, AND
C     KNOT DEFECT HALF-LENGTH AS PROJECTED ON THE FACE
C
      LRADUS = RB-KHIGH*TAN(RAD(LOG$))
      BDWID$ = DEG(ATAN(BOARDW/DISTBD))
      DY = DISTBD*TAN(RAD(KNOT$))
C
C     THE KNOT EFFECTIVELY ENDS AT THE OUTER EDGE OF THE LOG
C
      IF (KLEN.GT.LRADUS) KLEN = LRADUS
C
C     SINCE THE LOG IS TAPERED, THE OUTER BOARDS MAY NOT BE THE FULL LOG
C     LENGTH, SO KNOTS ABOVE THE END OF THE BOARD ARE NOT CONSIDERED
C
      IF (LRADUS.LT.DISTBD) PETURN
C
C     ANGLE OF ROTATION PUTS KNOT COMPLETELY OUTSIDE BOARD
C
      IF ((KTANG$.GT.180.).AND.((KTANG$+KNOT$).LT.(360.-BDWID$))) PETURN
      IF ((KTANG$.LE.180.).AND.((KTANG$-KNOT$).GT.BDWID$)) RETURN
C
C
C     FIND THE QUADRANT IN WHICH THE KNOT LIES (ASSUME QUADRANT ONE)
C
C     QUADRANT TWO
C
      IF ((KTANG$.LT.180.).AND.((KTANG$+KNOT$).GT.BDWID$)) GO TO 3
C
C     QUADRANT THREE
C
      IF ((KTANG$.GE.180.).AND.((KTANG$-KNOT$).LT.(360.-BDWID$)))
     . GO TO 4
C
C     QUADRANT FOUR
C
      IF (KTANG$.GT.270.) GO TO 5
C
C
C     QUADRANT ONE
C
C     SEE IF KNOT CENTER IS WITHIN ONE KNOT HALF-ANGLE OF 0 DEGREES
C
      IF (KTANG$-KNOT$.LT.0.) GO TO 2
C
C     CALCULATE DISTANCE TO FACE ALONG BOTH SIDES OF KNOT
C
      ANGBD1 = DISTBD/COS(RAD(KTANG$-KNOT$))
      ANGBD2 = DISTBD/COS(RAD(KTANG$+KNOT$))
C
C     KNOT IS NOT LONG ENOUGH TO REACH FACE
C
      IF (KLEN.LT.ANGBD1) PETURN
C
C     FIND INTERSECTION OF NEAR SIDE OF KNOT AND FACE
C
      XP = BOARDW-DISTBD*TAN(RAD(KTANG$-KNOT$))
      IF (KLEN.LT.ANGBD2) GO TO 1
```

```
C     FAR SIDE OF KNOT REACHES FACE
C
      XL = BOARDW-DISTBD*TAN(RAD(KTANG$+KNOT$))
      FULWID = (XR-XL)/2.
      GO TO 8
C
C     FAR SIDE OF KNOT DOES NOT REACH FACE
C
    1 XL = BOARDW-SQRT(KLEN**2-DISTBD**2)
      FULWID =
     . DISTBD*TAN(RAD(KTANG$+KNOT$))-DISTBD*TAN(RAD(KTANG$-KNOT$))
      GO TO 8
C
C     KNOT IS WITHIN ONE KNOT HALF-ANGLE OF 0 DEGREES
C
    2 XL = AMIN1(SQRT(KLEN**2-DISTBD**2),DISTBD*TAN(RAD(KNOT$+KTANG$)))
      XL = BOARDW-XL
      XR = AMIN1(SQRT(KLEN**2-DISTBD**2),DISTBD*TAN(RAD(KNOT$-KTANG$)))
      XR = BOARDW+XR
C
C     KNOTS NEARLY PERPENDICULAR HAVE THEIR WIDTH ESTIMATED BY THEIR
C     LENGTH
C
      FULWID = DY*2.
      GO TO 8
C
C     QUADRANT TWO
C
    3 ANGBD1 = DISTBD/COS(RAD(KTANG$-KNOT$))
C
C     TOO SHORT TO REACH FACE
C
      IF (KLEN.LT.ANGBD1) RETURN
      XP = BOARDW-SQRT(ANGBD1**2-DISTBD**2)
      XL = BOARDW-SQRT(KLEN**2-DISTBD**2)
      IF ((KTANG$+KNOT$).GE.89.5)
     . FULWID = LPADUS-DISTBD*TAN(RAD(KTANG$-KNOT$))
      IF ((KTANG$+KNOT$).LT.89.5)
     . FULWID = AMIN1(LRADUS,DISTBD*TAN(RAD(KTANG$+KNOT$)))
     .        -DISTBD*TAN(RAD(KTANG$-KNOT$))
      GO TO 8
C
C     QUADRANT THREE
C
    4 ANGBD1 = DISTBD/COS(RAD(360.-KTANG$-KNOT$))
      IF (KLEN.LT.ANGBD1) RETURN
      XR = BOARDW+SQRT(KLEN**2-DISTBD**2)
      XL = BOARDW+SQRT(ANGBD1**2-DISTBD**2)
      IF ((KTANG$-KNOT$).LE.270.5)
     . FULWID = LPADUS-DISTBD*TAN(RAD(360.-KTANG$-KNOT$))
      IF ((KTANG$-KNOT$).GT.270.5)
     . FULWID = AMIN1(LRADUS,DISTBD*TAN(RAD(360.-KTANG$+KNOT$)))
     .        -DISTBD*TAN(RAD(360.-KTANG$-KNOT$))
      GO TO 8
C
C     QUADRANT FOUR
C
    5 IF (360.-KTANG$-KNOT$.LT.0.) GO TO 7
      ANGBD1 = DISTBD/COS(RAD(360.-KTANG$-KNOT$))
      ANGBD2 = DISTBD/COS(RAD(360.-KTANG$+KNOT$))
      IF (KLEN.LT.ANGBD1) RETURN
      XL = BOARDW+DISTBD*TAN(RAD(360.-KTANG$-KNOT$))
      IF (KLEN.LT.ANGBD2) GO TO 6
C
C     FAR SIDE OF KNOT REACHES FACE
C
      XR = BOARDW+DISTBD*TAN(RAD(360.-KTANG$+KNOT$))
      FULWID = (XR-XL)/2.
      GO TO 8
C
C     FAR SIDE OF KNOT DOES NOT REACH FACE
C
    6 XP = BOARDW+SOPT(KLEN**2-DISTBD**2)
      FULWID = DISTBD*TWN(RAD(360.-KTANG$+KNOT$))
     . -DISTBD*TAN(RAD(360.-KTANG$-KNOT$))
      GO TO 8
C
C
```

```
154 C   KNOT IS WITHIN ONE KNOT HALF-ANGLE OF 0 DEGREES                    KNOTO
155 C                                                                      KNOTO
156      7 XL = AMIN1(SQRT(KLEN**2-DISTBD**2),                             KNOTO
157       .  DISTBD*TAN(RAD(KTANG$+KNOT$-360.)))                           KNOTO
158        XL = BOARDW-XL                                                  KNOTO
159        XR = AMIN1(SQRT(KLEN**2-DISTBD**2),                             KNOTO
160       .  DISTBD*TAN(RAD(360.-KTANG$+KNOT$)))                           KNOTO
161        XR = BOARDW+XR                                                  KNOTO
162        FULWID = DY*2.                                                  KNOTO
163 C                                                                      KNOTO
164 C   IF A KNOT COMPLETELY INTERSECTS THE FACE, ITS LENGTH IS EXACTLY AS KNOTO
165 C   MEASURED ON THE FACE.  OTHERWISE, ITS LENGTH IS ESTIMATED AS A     KNOTO
166 C   PROPORTION OF ITS PROJECTED LENGTH                                 KNOTO
167 C                                                                      KNOTO
168      8 DY = AMIN1(DY,DY*((XR-XL)*2./FULWID))                           KNOTO
169        YB = KHIGH-DY                                                   KNOTO
170        YT = KHIGH+DY                                                   KNOTO
171 C                                                                      KNOTO
172 C   DEFECT LOCATIONS FOUND.  CHECK THAT THEY DO NOT EXTEND BEYOND THE  KNOTO
173 C   BOARD EDGES AND CONVERT TO QUARTER INCH UNITS FOR THE GRADING      KNOTO
174 C   PROGRAM                                                            KNOTO
175 C                                                                      KNOTO
176        XL = AMAX0(0,INT(XL*4.))                                        KNOTO
177        XR = AMIN0(INT(BOARDW*0.),INT(XR*4.+1.))                        KNOTO
178        YB = AMAX0(0,INT(YB*4.))                                        KNOTO
179        YT = AMIN0(INT(BDLEN*4.),INT(YT*4.+1.))                         KNOTO
180 C                                                                      KNOTO
181 C   LIMIT NUMBER OF DEFECTS TO MAXIMUM ALLOWED BY GRADING PROGRAM (22) KNOTO
182 C                                                                      KNOTO
183        NODEFC(1) = NODEFC(1)+1                                         KNOTO
184        IF (NODEFC(1).GT.22) RETURN                                     KNOTO
185 C                                                                      KNOTO
186 C   SAVE KNOT COORDINATES FOR GRADING PROGRAM                          KNOTO
187 C                                                                      KNOTO
188        ID(NODEFC(1),1) = 3                                             KNOTO
189        FLY(NODEFC(1),1) = XL                                           KNOTO
190        FLX(NODEFC(1),1) = YB                                           KNOTO
191        FUY(NODEFC(1),1) = XR                                           KNOTO
192        FUX(NODEFC(1),1) = YT                                           KNOTO
193        RETURN                                                          KNOTO
194        END                                                             KNOTO
```

```
           142   148   158   161   177
COS         65    66   104   120   134   135
D           11
DEG         23
DISTBD      11    17    23    24    33    65    66    74    78    84    85    91
            93   104   109   110   111   113   120   122   123   124   126   134
           135   137   142   148   149   156   159
DY         *24    99   162  *168   169   170
FLX         10    11  *190
FLY         10    11  *189
FULWID     *79   *85   *99  *111  *113  *124  *126  *143  *149  *162   168
FUX         10    11  *192
FUY         10    11  *191
ID           9    11  *188
INT        176   177   178   179
K           11
KHIGH        6    22   169   170
KLEN         6    17   *28    70    75    84    91    93   108   110   121   122
           136   138   148   156   159
KNOT$       11    24    37    38    45    49    61    65    66    74    78    85
            91    93   104   111   113   120   124   126   133   134   135   137
           142   149   156   159
KNOTO        6
KTANG$       6    37    38    45    49    54    61    65    66    74    78    85
            91    93   104   111   113   120   124   126   133   134   135   137
           142   149   156   159
L           11
LOG$        11    22
LRADUS     *22    28    33   111   113   124   126
NODEFC       9    11  *183   184   188   189   190   191   192
R           11
RAD         22    24    65    66    74    78    85    91    93   104   111   113
           120   124   126   134   135   137   142   149   156   159
RB          11    22
SQRT        84    91    93   109   110   122   123   148   156   159
T           11
TAN         22    24    74    78    85    91    93   111   113   124   126   137
           142   149   156   159
XL         *78    79   *84   *91   *92  *110  *123  *137   143  *156  *158   168
          *176   189
XR         *73    73   *93   *94  *109  *122  *142   143  *148  *159   161   168
          *177   191
YB        *169  *178   190
YT        *170  *179   192
```

*** STATEMENT NUMBERS ***

```
1      75   *84
2      61   *91
3      45  *104
4      49  *120
5      54  *133
6     138  *148
7     133  *156
8      80    87   100   116   129   144   151  *168
```

*** VARIABLES ***

```
AMAX0     176   178
AMIN0     177   179
AMIN1      91    93   113   126   156   159   168
ANGBD1    *65    70  *104   108   109  *120   121   123  *134   136
ANGBD2    *66    75  *135   138
ATAN       23
BDLEN      11   179
BDWID$    *23    37    38    45    49
BF         11
BOARDW     11    23    74    78    84    92    94   109   110   122   123   137
```

Subroutine CORE

```
C   SUBROUTINE 'CORE' LOCATES THE CORE DEFECT ON THE BOARD FACE          COREO
C                                                                        COREO
C                                                                        COREO
      SUBROUTINE COREO (CRADUS,COFSET,CANG$)                             COREO
C                                                                        COREO
      IMPLICIT REAL (K,L)                                                COREO
      REAL FLX(26,2),FLY(26,2),FUX(26,2),FUY(26,2)                       COREO
      INTEGER NODEFC(2),ID(22,2)                                         COREO
      COMMON /BOARD/ L,D,K,T,LOG$,RB,DISTBD,BDLEN,BOARDW,KNOT$,R,BF      COREO
             /DEFEC/ NODEFC,FLX,FLY,FUX,FUY,ID                           COREO
C                                                                        COREO
C                                                                        COREO
C   FIND DISTANCE TO THE BOARD FACE FROM THE CENTER OF THE CORE. EXIT    COREO
C   IF THE BOARD FACE IS BEYOND THE CORE RADIUS                          COREO
C                                                                        COREO
      TDIST = DISTBD-COFSET*COS(RAD(CANG$))                              COREO
      IF (CRADUS.LE.TDIST) RETURN                                        COREO
      NODEFC(1) = NODEFC(1)+1                                            COREO
C                                                                        COREO
C   LIMIT NUMBER OF DEFECTS TO MAXIMUM ALLOWED BY GRADING PROGRAM (22)   COREO
C                                                                        COREO
      IF (NODEFC(1).GT.22) RETURN                                        COREO
C                                                                        COREO
C   LOCATE THE CENTER POINT OF THE INTERSECTION OF THE BOARD FACE AND    COREO
C   THE CORE DEFECT.  CALCULATE THE DISTANCE ON THE BOARD FROM THE       COREO
C   CENTER POINT TO THE EDGES OF THE CORE DEFECT.  THE CORE DEFECT       COREO
C   IS NOT TAPERED, SO IT WILL EXTEND THE FULL LENGTH OF THE BOARD       COREO
C                                                                        COREO
      CONST = SQRT(CRADUS**2-TDIST**2)                                   COREO
      TBDCTR = COFSET*SIN(RAD(CANG$))                                    COREO
      XR = BOARDW+CONST-TBDCTR                                           COREO
      XL = BOARDW-CONST-TBDCTR                                           COREO
C                                                                        COREO
C   SAVE CORE DEFECT COORDINATES FOR GRADING PROGRAM                     COREO
C                                                                        COREO
      FUX(NODEFC(1),1) = INT(BDLEN*4.)                                   COREO
      FLX(NODEFC(1),1) = 0.0                                             COREO
      FUY(NODEFC(1),1) = AMIN0(INT(BOARDW*8.),INT(XR*4.+1.))             COREO
      FLY(NODEFC(1),1) = AMAX0(0,INT(XL*4.))                             COREO
      ID(NODEFC(1),1) = 7                                                COREO
      RETURN                                                             COREO
      END                                                                COREO
```

*** VARIABLES ***

AMAX0	39								
AMIN0	38								
BDLEN	9	36							
BF	9								
BOARDW	9	31	32	38					
CANG$	4	16	30						
COFSET	4	16	30						
CONST	*29	31	32						
COREO	4								
COS	16								
CRADUS	4	17	29						
D	9								
DISTBD	9	16							
FLX	7	9	*37						
FLY	7	9	*39						
FUX	7	9	*36						
FUY	7	9	*38						
ID	8	9	*40						
INT	36	38	39						
K	9								
KNOT$	9								
L	9								
LOG$	9								
NODEFC	8	9	*18	22	36	37	38	39	40
P	9								
RAD	16	30							
RB	9								
SIN	30								
SQRT	29								
T	9								
TBDCTR	*30	31	32						
TDIST	*16	17	29						
XL	*32	39							
XR	*31	38							

Subroutine GRADE

```
C     SUBROUTINE 'GRADE' CALLS THE U.S. FOREST PRODUCTS LABORATORY      GRADE
C     GRADING PROGRAM                                                   GRADE
C                                                                       GRADE
C                                                                       GRADE
        SUBROUTINE GRADE (NPG)                                          GRADE
C                                                                       GRADE
        INTEGER ID(22,2),NODEFC(2),IX(22)                               GRADE
        REAL FLX(26,2),FLY(26,2),FUX(26,2),FUY(26,2)                    GRADE
        COMMON /DEFEC/ NODEFC,FLX,FLY,FUX,FUY,ID                        GRADE
C                                                                       GRADE
C                                                                       GRADE
        M = NODEFC(1)                                                   GRADE
        IF (M.GT.22) GO TO 3                                            GRADE
        IF (M.EQ.0) GO TO 2                                             GRADE
        DO 1 J=1,M                                                      GRADE
          IX(J) = ID(J,1)                                               GRADE
      1 CONTINUE                                                        GRADE
      2 NPG = 1                                                         GRADE
        CALL GRD (M,IX,NPG,SM)                                          GRADE
        RETURN                                                          GRADE
C                                                                       GRADE
C     IF NUMBER OF DEFECTS IS MORE THAN GRADING PROGRAM CAN HANDLE,     GRADE
C     THE GRADE IS 3A 3B                                                GRADE
C                                                                       GRADE
      3 NPG = 5                                                         GRADE
        WRITE (6,100)                                                   GRADE
    100 FORMAT ('          DEFECT LIMIT EXCEEDED')                      GRADE
        RETURN                                                          GRADE
        END                                                             GRADE
```

*** 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				
GRD	19				
ID	7	9	16		
IX	7	*16	19		
J	*15	16			
M	*12	13	14	15	19
NODEFC	7	9	12		
NPG	5	*18	19	*25	
SM	19				

Subroutine PRICE

```
C     SUBROUTINE 'PRICE' DETERMINES THE BOARD GRADE BASED ON THE GRADES PRICED
C     OF BOTH SIDES (OLD,NPG) AND CALCULATES THE BOARD VALUE            PRICED
C                                                                       PRICED
C                                                                       PRICED
        SUBROUTINE PRICED (OLD,NPG,TVAL,PERC)                           PRICED
C                                                                       PRICED
        IMPLICIT REAL (K,L)                                             PRICED
        INTEGER OLD                                                     PRICED
        REAL PERC(5)                                                    PRICED
        COMMON /PRICE/ FAS,SEL,ONEC,TWOC,THRB                           PRICED
     *         /BOARD/ L,D,K,T,LOG$,PB,DISTBD,BDLEN,BOARDW,KNOT$,R,BF   PRICED
C                                                                       PRICED
C                                                                       PRICED
C     DETERMINE THE BOARD GRADE                                         PRICED
C                                                                       PRICED
        IF (OLD.EQ.5.OR.NPG.EQ.5) GO TO 3                               PRICED
        IF (OLD.EQ.2.OR.NPG.EQ.2) GO TO 1                               PRICED
        IF (OLD.EQ.NPG) GO TO 4                                         PRICED
        NPG = MAX0(OLD,NPG)                                             PRICED
        IF (NPG.EQ.4) GO TO 8                                           PRICED
        NPG = 2                                                         PRICED
        GO TO 4                                                         PRICED
      1 IF (OLD.EQ.4.OR.NPG.EQ.4) GO TO 2                               PRICED
        NPG = 3                                                         PRICED
        GO TO 4                                                         PRICED
      2 NPG = 4                                                         PRICED
        GO TO 4                                                         PRICED
      3 NPG = 5                                                         PRICED
      4 GO TO (5,6,7,8,9), NPG                                          PRICED
C                                                                       PRICED
C     AND CALCULATE THE BOARD VALUE                                     PRICED
C                                                                       PRICED
      5 V = BF*FAS                                                      PRICED
        GO TO 10                                                        PRICED
      6 V = BF*SEL                                                      PRICED
        GO TO 10                                                        PRICED
      7 V = BF*ONEC                                                     PRICED
        GO TO 10                                                        PRICED
      8 V = BF*TWOC                                                     PRICED
        GO TO 10                                                        PRICED
      9 V = BF*THRB                                                     PRICED
C                                                                       PRICED
C     INCREASE SURFACE MEASURE IN THIS GRADE AND THE TOTAL LOG VALUE    PRICED
C                                                                       PRICED
     10 PERC(NPG) = BF+PERC(NPG)                                        PRICED
        TVAL = TVAL+V                                                   PRICED
        RETURN                                                          PRICED
        END                                                             PRICED
```

*** STATEMENT NUMBERS ***

1	17	*23			
2	23	*26			
3	16	*28			
4	18	22	25	27	*29
5	29	*33			
6	29	*35			
7	29	*37			
8	20	29	*39		
9	29	*41			
10	34	36	38	40	*45

```
                    *** VARIABLES ***

BDLEN      10
BF         10   33   35   37   39   41   45
BOARDW     10
D          10
DISTBD     10
FAS        10   33
K          10
KNOTS      10
L          10
LOGS       10
MAXB       19
NPG         5   16   17   18  *19   20  *21   23  *24  *26  *28   29
           45
OLD         5    8   16   17   18   19   23
ONEC       10   37
PEF_        5    9  *45
PRICEQ      5
R          10
RB         10
SEL        10   35
T          10
THRB       10   41
TVAL        5  *46
TWOF       10   39
V         *33  *35  *37  *39  *41   46
```

Function DEG

```
C     FUNCTION 'DEG' CONVERTS ITS ARGUMENT (IN RADIANS) TO DEGREES       DEG
C                                                                        DEG
C                                                                        DEG
      FUNCTION DEG (A)                                                   DEG
C                                                                        DEG
C                                                                        DEG
      DEG = A*180.0/3.141592                                             DEG
      RETURN                                                             DEG
      END                                                                DEG
```

```
                    *** VARIABLES ***

A            4    7
DEG          4   *7
```

Function RAD

```
C   FUNCTION 'RAD' CONVERTS ITS ARGUMENT (IN DEGREES) TO RADIANS          RAD
C                                                                         RAD
C                                                                         RAD
      FUNCTION RAD (A)                                                    RAD
C                                                                         RAD
C                                                                         RAD
      RAD = A*3.141592/180.0                                              RAD
      RETURN                                                              RAD
      END                                                                 RAD
```

```
                    *** VARIABLES ***

A            4    7
RAD          4   *7
```

```
C     'CANT' CUTS FROM FACES 1 AND 3 LEAVING A CANT OF SPECIFIED
C     THICKNESS.  THIS CANT IS THEN TURNED 90 DEGREES AND CUT
C     INTO BOARDS
C
C
        IMPLICIT REAL (K.L)
        REAL FLX(26.2).FLY(26.2).FUX(26.2).FUY(26.2).PERC(5).ANGLE(50).
     .      HEIGHT(50).KTLEN(50).DAT*8
        INTEGER ID(22.2).NODEFC(2).OLD.F
        COMMON /DEFEC/ NODEFC.FLX.FLY.FUX.FUY.ID
     .         /BOARD/ L.D.K.T.LOGS.RB.DISTBD.BDLEN.BOARDW.KNOTS.CANT.R.BF
     .         /PRICF/ FAS.SEL.ONEC.TWOC.THRB
C
C
C     THIS PROGRAM ASSUMES INPUT FROM UNIT 5 AND OUTPUTS TO UNITS 6 AND 8
C
C     GET LOG DIAMETER (IN.). LOG LENGTH (IN.). KERF SIZE (IN.). AND
C     BOARD THICKNESS (IN.)
C
      1 READ (5.102.END=22) D
        READ (5.102) L
        READ (5.103) K
        READ (5.102) T
C
C     GET KNOT HALF-ANGLE (DEGREES) AND LOG HALF-TAPER (DEGREES)
C
        READ (5.102) KNOTS
        READ (5.102) LOGS
C
C     GET CORE DEFECT PARAMETERS: DIAMETER (IN.). LINEAR OFFSET (IN.).
C     AND ANGULAR OFFSET (DEGREES)
C
        READ (5.106) CORDIA
        READ (5.106) COFSET
        READ (5.106) CANGS
C
C     GET PRICE PER BOARD FOOT FOR EACH GRADE: NUMBER OF KNOTS
C
        READ (5.105) FAS.SEL.ONEC.TWOC.THRB
        READ (5.100) MAX
C
C     GET DATE AND CALCULATE RADIUS OF CORE DEFECT
C
        CALL DATE (DAT)
        CRADUS = CORDIA/2.
C
C     GET KNOT ANGLES. HEIGHTS. AND LENGTHS
C
        DO 2 J=1.MAX
          READ (5.104) ANGLE(J).HEIGHT(J)
      2 CONTINUE
        DO 3 J=1.MAX
          READ (5.106) KTLEN(J)
      3 CONTINUE
C
C     CALCULATE LOG RADIUS AT TOP AND BOTTOM: TOTAL LOG CUBIC FEET
C
        RI = D/2.
        RB = RI+L*TAN(RAD(LOGS))
        LVOL = 1/3.*3.141593*(RB**2+RB*RI+RI**2)*L/144.
C
C     CALCULATE CENTRAL CANT SIZE
C
        CANT = D/2.-2.
C
C     OUTPUT HEADER
C
        WRITE (6.116) DAT
        WRITE (6.117) L.D.LOGS.LVOL.CANT
        WRITE (6.101) MAX
        DO 4 J=1.MAX
          WRITE (6.105) ANGLE(J).HEIGHT(J).KTLEN(J)
      4 CONTINUE
        WRITE (6.118) KNOTS
        WRITE (6.110) CRADUS.COFSET.CANGS
        WRITE (6.119) K.T
        WRITE (6.120) FAS.SEL.ONEC.TWOC.THRB
        WRITE (8.200) L.D.MAX.CRADUS.COFSET.CANGS.LVOL.K.T.KNOTS.LOGS
        WRITE (8.201) (KTLEN(J).J=1.MAX)
        WRITE (8.201) (HEIGHT(J).J=1.MAX)
        WRITE (8.201) (ANGLE(J).J=1.MAX)
        WRITE (8.202) FAS.SEL.ONEC.TWOC.THRB
C
C     INITIALIZE VARIABLES FOR CALCULATING HIGH. LOW. AND AVERAGE
C     YIELDS FOR EACH LOG
C
        HIGHS = 0.
        SMALLS = 999.
        AVGS = 0.
C
C     SAW LOG IN 12 DIFFERENT ROTATIONAL POSITIONS. 15 DEGREES APART. TO
C     DETERMINE THE ORIENTATION GIVING THE HIGHEST YIELD
C
        DO 21 INX=1.12
C
C     INITIALIZE FOR EACH POSITION
C
          NBD = 1
          TVAL = 0.0
          BF = 0.0
C
C     CUTTING ORDER IS FACE 1. FACE 3. FACE 4. AND FACE 2
C
          DO 5 J=1.5
            PERC(J) = 0.0
      5   CONTINUE
C
C     FOR FOUR FACES OF THE LOG...
C
          DO 16 F=1.4
C
C     THE SLAB FACES ARE CUT TO THE DEFINED CANT
C
            IF (F.EQ.1.OR.F.EQ.2) R = RB-CANT/2.
C
C     THE CANT FACES ARE KERF CENTERED TO DIVIDE THE CANT
C
            IF (F.EQ.3.OR.F.EQ.4) R = RB+K/2.
C
C     CUT BOARDS FROM THIS FACE UNTIL FACE IS COMPLETELY CUT.  THE LOG IS
C     CUT FROM THE INSIDE OUT
C
      6     CALL KERFC (&11.F)
C
C     FOR BOTH SIDES OF THE BOARD...
C
            DO 10 I=1.2
C
C     INITIALIZE DEFECT ARRAYS. EXCEPT F..(26.1). WHICH HOLDS THE BOARD
C     DIMENSIONS
C
              DO 7 J=1.25
                FLX(J.1) = -9999.0
                FUX(J.1) = -9999.0
                FLY(J.1) = -9999.0
                FUY(J.1) = -9999.0
      7       CONTINUE
              DO 8 J=1.22
                ID(J.1) = 0
      8       CONTINUE
              NODEFC(1) = 0
C
C     PUT IN WANE DEFECTS
C
              CALL WANEC
C
C     PUT IN KNOTS
C
              DO 9 J=1.MAX
                CALL KNOTC(HEIGHT(J).KTLEN(J).ANGLE(J))
```

```
151     9       CONTINUE                                                 CANT
152 C                                                                    CANT
153 C   PUT IN CORE DEFECT                                               CANT
154 C                                                                    CANT
155             CALL COREC (CRADUS,COFSET,CANGS)                         CANT
156 C                                                                    CANT
157 C   AND GRADE THE FACE                                               CANT
158 C                                                                    CANT
159             CALL GRADE (NPG)                                         CANT
160 C                                                                    CANT
161 C   SAVE GRADE OF FIRST SIDE OF BOARD                                CANT
162 C                                                                    CANT
163             IF (I.EQ.1) OLD = NPG                                    CANT
164 C                                                                    CANT
165 C   SUBTRACT BOARD THICKNESS AND PROCESS THE INNER FACE              CANT
166 C                                                                    CANT
167             DISTBD = DISTBD-T                                        CANT
168     10      CONTINUE                                                 CANT
169 C                                                                    CANT
170 C   DETERMINE BOARD GRADE AND VALUE BASED ON GRADES OF BOTH FACES    CANT
171 C                                                                    CANT
172           CALL PRICEC (OLD,NPG,TVAL,PERC)                            CANT
173 C                                                                    CANT
174 C   INCREASE BOARD COUNT AND CUT ANOTHER BOARD                       CANT
175 C                                                                    CANT
176           NBD  =  NBD + 1                                            CANT
177           GO TO 6                                                    CANT
178 C                                                                    CANT
179 C   TURN LOG TO NEXT FACE                                            CANT
180 C                                                                    CANT
181     11    GO TO (12,14,12,16), F                                     CANT
182 C                                                                    CANT
183 C   TURN LOG BY 180 DEGREES TO CUT THE OPPOSITE FACE.                CANT
184 C                                                                    CANT
185     12    DO 13 J=1,MAX                                              CANT
186             ANGLE(J) = AMOD(ANGLE(J)+180.0,360.0)                    CANT
187     13    CONTINUE                                                   CANT
188           CANGS = AMOD(CANGS+180.0,360.0)                            CANT
189           GO TO 16                                                   CANT
190 C                                                                    CANT
191 C   OR TURN 90 DEGREES TO CUT A CANT FACE                            CANT
192 C                                                                    CANT
193     14    DO 15 J=1,MAX                                              CANT
194             ANGLE(J) = AMOD(ANGLE(J)+90.0,360.0)                     CANT
195     15    CONTINUE                                                   CANT
196           CANGS = AMOD(CANGS+90.0,360.0)                             CANT
197     16  CONTINUE                                                     CANT
198 C                                                                    CANT
199 C   LOG HAS BEEN COMPLETELY CUT...RETURN ANGLES TO ORIGINAL VALUES   CANT
200 C                                                                    CANT
201         DO 18 J=1,MAX                                                CANT
202           ANGLE(J) = AMOD(ANGLE(J)+270.0,360.0)                      CANT
203     18  CONTINUE                                                     CANT
204         CANGS = AMOD(CANGS+270.,360.)                                CANT
205 C                                                                    CANT
206 C   PRINT RESULTS FOR THIS POSITION                                  CANT
207 C                                                                    CANT
208         IROTAT = INX*15-15                                           CANT
209         WRITE (6,107) IROTAT                                         CANT
210         WRITE (6,111) (PERC(J),J=1,5)                                CANT
211         TOTS = PERC(1)+PERC(2)+PERC(3)+PERC(4)+PERC(5)               CANT
212         WRITE (6,112) TOTS                                           CANT
213         WRITE (8,203) IROTAT                                         CANT
214         WRITE (8,204) (PERC(J),J=1,5)                                CANT
215         WRITE (8,204) TOTS                                           CANT
216         DO 19 J=1,5                                                  CANT
217           PERC(J) = PERC(J)/LVOL*100.                                CANT
218     19  CONTINUE                                                     CANT
219         TOTS = TOTS*100./LVOL                                        CANT
220         WRITE (6,113) TOTS                                           CANT
221         WRITE (6,108) (ANGLE(J),J=1,MAX)                             CANT
222         WRITE (6,109) (PERC(J),J=1,5),TVAL                           CANT
223         WRITE (8,205) (PERC(J),J=1,5)                                CANT
224         WRITE (8,204) TVAL,TOTS                                      CANT
225 C                                                                    CANT
226 C   SAVE HIGH, LOW, AND AVERAGE YIELDS FOR THIS LOG                  CANT
227 C                                                                    CANT
228         AVGS = AVGS+TVAL                                             CANT
229           IF (TVAL.GT.HIGHS) HIGHS = TVAL                            CANT
230           IF (TVAL.LT.SMALLS) SMALLS = TVAL                          CANT
231 C                                                                    CANT
232 C   ROTATE LOG BY 15 DEGREES AND REPROCESS                           CANT
233 C                                                                    CANT
234           DO 20 J=1,MAX                                              CANT
235             ANGLE(J) = AMOD(ANGLE(J)+15.0,360.0)                     CANT
236     20    CONTINUE                                                   CANT
237           CANGS = AMOD(CANGS+15.0,360.0)                             CANT
238     21 CONTINUE                                                      CANT
239 C                                                                    CANT
240 C   PRINT HIGH, LOW, AND AVERAGE YIELDS FOR THIS LOG                 CANT
241 C                                                                    CANT
242         AVGS = AVGS/12.                                              CANT
243         WRITE (6,114) HIGHS,AVGS,SMALLS                              CANT
244 C                                                                    CANT
245 C   LOG COMPLETELY PROCESSED, READ PARAMETERS FOR NEXT LOG           CANT
246 C                                                                    CANT
247         GO TO 1                                                      CANT
248 C                                                                    CANT
249     22 STOP                                                          CANT
250 C                                                                    CANT
251 C   FORMATS FOR DEVICE 6 (PRINTER)                                   CANT
252 C                                                                    CANT
253   100 FORMAT (I2)                                                    CANT
254   101 FORMAT ('1THERE APE',I3,' KNOTS IN THIS LOG.  ',2(/),4X,       CANT
255      .  ' ANGLE  HEIGHT   LENGTH')                                   CANT
256   102 FORMAT (F5.1)                                                  CANT
257   103 FORMAT (F5.3)                                                  CANT
258   104 FORMAT (2F6.2)                                                 CANT
259   105 FORMAT (3(2X,F6.2))                                            CANT
260   106 FORMAT (F6.3)                                                  CANT
261   107 FORMAT (2(/),' ROTATION IS',I4)                                CANT
262   108 FORMAT (' KNOT ANGLES: ',15F7.2,2(/,13X,15F7.2))               CANT
263   109 FORMAT (' % AVAILABLE SURFACE MEASURE IN EACH GRADE ',5F5.1,   CANT
264      .  'LOG VALUE IS $',F6.2)                                       CANT
265   110 FORMAT (' CORE DEFECT RADIUS ',F6.3,'  LINEAR OFFSET ',F6.3,   CANT
266      .  '  ANGULAR OFFSET ',F6.3)                                    CANT
267   111 FORMAT (' SURFACE MEASURE/GRADE IS',5F7.2)                     CANT
268   112 FORMAT (' TOTAL SURFACE MEASURE IS ',F8.2)                     CANT
269   113 FORMAT (' TOTAL PERCENT YIELD IS',F8.2)                        CANT
270   114 FORMAT (' $ YIELD:  HIGH ',F6.2,'  AVERAGE ',F6.2,'  LOW ',F6.2)  CANT
271   115 FORMAT (5F10.4)                                                CANT
272   116 FORMAT (' CANT SAWING METHOD ',A8)                             CANT
273   117 FORMAT (' LOG PARAMETERS:  LENGTH',F6.1,'  DIA',F6.1,'  TAPER' CANT
274      .  F6.4,'DEGREES  LOG VOLUME',F8.2,'  CANT SIZE',F6.2)          CANT
275   118 FORMAT (' ANGLES MEASURED FROM ZERO DEGREES = EAST FOR A       CANT
276      .  'VERTICALLY CUTTING SAW.  KNOT TAPER',F6.2)                  CANT
277   119 FORMAT (' CUTTING PARAMETERS:  KERF',F6.4,'  BOARD THICKNESS'  CANT
278      .  F5.2)                                                        CANT
279   120 FORMAT (' PRICES PER BOARD FOOT: ',5F10.4)                     CANT
280 C                                                                    CANT
281 C   FORMATS FOR DEVICE 8 (STORAGE MEDIUM)                            CANT
282 C                                                                    CANT
283   200 FORMAT (F6.1,F5.1,I3,3F5.1,F8.2,F6.3,F4.1,F6.2,F5.3,F6.3)      CANT
284   201 FORMAT (3(10F7.2/))                                            CANT
285   202 FORMAT (5F7.4)                                                 CANT
286   203 FORMAT (I3)                                                    CANT
287   204 FORMAT (5F8.2)                                                 CANT
288   205 FORMAT (10F6.2)                                                CANT
289       END                                                            CANT
```

```
                *** STATEMENT NUMBERS ***

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

```
     5    104  *106
     6   *123   177
     7    132  *137
     8    138  *140
     9    149  *151
    10    127  *168
    11    123  *101
    12    101  *105
    13    105  *107
    14    101  *193
    15    193  *195
    16    110   101   109  *197
    18    201  *203
    19    216  *218
    20    234  *236
    21     94  *238
    22     20  *249
   100     40  *253
   101     70  *254
   102     20    21    23    27    28  *256
   103     22  *257
   104     50  *258
   105     72  *259
   106     33    34    35    53  *260
   107    209  *261
   108    221  *262
   109    222  *263
   110     75  *265
   111    210  *267
   112    212  *268
   113    220  *269
   114    243  *270
   115     39  *271
   116     63  *272
   117     69  *273
   118     74  *275
   119     76  *277
   120     77  *279
   200     78  *283
   201     79    80    81  *284
   202     82  *285
   203    213  *286
   204    214   215   224  *287
   205    223  *288
```

*** VARIABLES ***

```
AMOD     186   188   194   196   202   204   235   237
ANGLE      7   *50    72    81   150  *186  *194  *202   221  *235
AVGS     *89  *228  *242   243
BDLEN     10
BF        10  *100
BOARDW    10
CANGS    *35    75    78   155  *188  *196  *204  *237
CANT      10   *64    69   114
COFSET   *34    75    78   155
CORDIA   *33    45
COREC    155
CRADUS   *45    75    78   155
D         10   *20    58    64    69    78
DAT        7    44    60
DATE      44
DISTBD    10  *167
F          9  *110   114   118   123   101
FAS       10   *39    77    82
FLX        7    10  *133
FLY        7    10  *135
FUX        7    10  *134
FUY        7    10  *136
GRADE    159
HEIGHT     7   *50    72    80   150
HIGHS    *87  *229   243
I       *127   163
ID         9    10  *139
INX      *94   200
```

```
IROTAT  *200   209   213
J        *43   *50   *52   *53   *71    72    79    80    81  *104   105  *132
         133   134   135   136  *138   139  *149   150  *105   106  *193   194
        *201   202   210   214  *216   217   221   222   223  *234   235
K         10   *22    76    78   118
KERFC    123
KNOTS     10   *27    74    78
KNOTC    150
KTLEN      7   *53    72    79   150
L         10   *21    59    60    69    78
LOGS      10   *28    59    69    78
LVOL     *60    69    78   217   219
MAX      *40    49    52    70    71    78    79    80    81   149   105   193
         201   221   234
NBD      *98  *176
NODEFC     9    10  *141
NPG      159   163   172
OLD        9  *163   172
ONEC      10   *39    77    82
PERC       7  *105   172   210   211   214  *217   222   223
PRICEC   172
R         10  *114  *118
RAD       59
RB        10   *59    60   114   118
RI       *58    59    60
SEL       10   *39    77    82
SMALLS   *88  *230   243
T         10   *23    76    78   167
TAN       59
THRB      10   *39    77    82
TOTS    *211   212   215  *219   220   224
TVAL     *99   172   222   224   228   229   230
TWNC      10   *39    77    82
WDNEC    14?
```

Subroutine KERF

```
C   SUBROUTINE 'KERF' CUTS A BOARD FROM THE LOG.  THE BOARD WIDTH IS    KERFC
C   DEFINED AT THE CENTER, AND THE BOARD LENGTH FROM THE BUTTOM         KERFC
C   OF THE LOG                                                          KERFC
C                                                                       KERFC
C                                                                       KERFC
      SUBROUTINE KERFC (*,F)                                            KERFC
C                                                                       KERFC
      IMPLICIT REAL (K,L)                                               KERFC
      INTEGER ID(22,2),NODEFC(2),F                                      KERFC
      REAL FLX(26,2),FLY(26,2),FUX(26,2),FUY(26,2)                      KERFC
      COMMON /BOARD/ L,D,K,T,LOG$,RB,DISTBD,BDLEN,BOARDW,KNOTS,CANT,R,BF KERFC
             /DEFEC/ NODEFC,FLX,FLY,FUX,FUY,ID                          KERFC
C                                                                       KERFC
C                                                                       KERFC
C   ALL BOARDS MUST BE AT LEAST 48 INCHES LONG, 2.5 INCHES WIDE AT      KERFC
C   THE TOP, AND 3 INCHES WIDE AT THE CENTER.  IN ADDITION, THE TOTAL   KERFC
C   AMOUNT OF WANE FROM THE CENTER TO THE TOP IS LIMITED TO 4 INCHES.   KERFC
C   THE BOARD IS CUT BACK IN ONE FOOT INCREMENTS IF THE WIDTH OR WANE   KERFC
C   TESTS FAIL.  IF THE LENGTH TEST FAILS, ANOTHER BOARD IS CUT FROM    KERFC
C   THIS FACE                                                           KERFC
C                                                                       KERFC
C                                                                       KERFC
C   FIND OUTER FACE OF BOARD                                            KERFC
C                                                                       KERFC
      RI = D/2.                                                         KERFC
    1 R = R-T-K                                                         KERFC
      BDLEN = L                                                         KERFC
C                                                                       KERFC
C   IF THIS FACE OF THE LOG IS COMPLETELY CUT, EXIT                     KERFC
C                                                                       KERFC
      IF (R.LT.0.) RETURN1                                              KERFC
      DISTBD = RB-R                                                     KERFC
      POSLEN = R*TAN(RAD(90.-LOG$))                                     KERFC
C                                                                       KERFC
C                                                                       KERFC
C   CHECK FOR BOARDS SHORTER THAN THE LOG DUE TO LOG TAPER              KERFC
C                                                                       KERFC
      IF (POSLEN.LT.BDLEN) BDLEN = POSLEN                               KERFC
C                                                                       KERFC
C   BOARDS MUST BE AT LEAST 4 FEET LONG                                 KERFC
C                                                                       KERFC
    2 IF (BDLEN.LT.48.) GO TO 1                                         KERFC
C                                                                       KERFC
C   DETERMINE HALF-WIDTH AT MID-LENGTH OF BOARD                         KERFC
C                                                                       KERFC
      RM = D/2.+(L-BDLEN/2.)*TAN(RAD(LOG$))                             KERFC
      IF (DISTBD.GT.RM) GO TO 3                                         KERFC
      BOARDW = SQRT(RM**2-DISTBD**2)                                    KERFC
C                                                                       KERFC
C   FOR CANT CUTS, BOARD IS NO WIDER THAN CANT                          KERFC
C                                                                       KERFC
      IF (F.EQ.3.OR.F.EQ.4) BOARDW = AMIN1(CANT/2.,BOARDW)              KERFC
C                                                                       KERFC
C   BOARD MUST BE AT LEAST 3 INCHES WIDE AT MID-LENGTH                  KERFC
C                                                                       KERFC
      IF (BOARDW.LT.1.5) GO TO 3                                        KERFC
C                                                                       KERFC
C   DETERMINE HALF-WIDTH AT TOP OF BOARD                                KERFC
C                                                                       KERFC
      RT = D/2.+(L-BDLEN)*TAN(RAD(LOG$))                                KERFC
      IF (DISTBD.GT.RT) GO TO 3                                         KERFC
      WT = SQRT(RT**2-DISTBD**2)                                        KERFC
      IF (F.EQ.3.OR.F.EQ.4) WT = AMIN1(CANT/2.,WT)                      KERFC
C                                                                       KERFC
C   BOARD MUST BE AT LEAST 2.5 INCHES WIDE AT THE TOP                   KERFC
C                                                                       KERFC
      IF (WT.LT.1.25) GO TO 3                                           KERFC
C                                                                       KERFC
C   LIMIT THE TOTAL WANE TO AVOID GENERATING MORE DEFECTS THAN THE      KERFC
C   GRADING PROGRAM CAN HANDLE.  TOTAL WIDTH OF WANE MUST BE LESS       KERFC
C   THAN 4 INCHES                                                       KERFC
C                                                                       KERFC
      WAN = BOARDW-WT                                                   KERFC
      IF (WAN.GT.2.0) GO TO 3                                           KERFC
C                                                                       KERFC
C   CONVERT BOARD WIDTH AND LENGTH TO EVEN QUARTER INCHES               KERFC
C                                                                       KERFC
      BOARDW = INT(BOARDW*8.)/8.                                        KERFC
      BDLEN = INT(BDLEN*4.)/4.                                          KERFC
C                                                                       KERFC
C   DEFINE BOARD EDGES IN QUARTER INCH UNITS                            KERFC
C                                                                       KERFC
      FLX(26,1) = BDLEN*4.                                              KERFC
      FLY(26,1) = BOARDW*8.                                             KERFC
      FUX(26,1) = 0.0                                                   KERFC
      FUY(26,1) = 0.0                                                   KERFC
C                                                                       KERFC
C   CALCULATE UNROUNDED BOARD FEET                                      KERFC
C                                                                       KERFC
      BF = BDLEN*BOARDW*2.*T/144.                                       KERFC
      RETURN                                                            KERFC
C                                                                       KERFC
C   CUT BACK BOARD BY ONE FOOT                                          KERFC
C                                                                       KERFC
    3 BDLEN = BDLEN-12.                                                 KERFC
      GO TO 2                                                           KERFC
      END                                                               KERFC
```

*** STATEMENT NUMBERS ***

1	*26	42				
2	*42	96				
3	47	56	61	67	74	*95

*** VARIABLES ***

AMIN1	52	63								
BDLEN	11	*27	*38	42	46	60	*79	83	90	*95
BF	11	*90								
BOARDW	11	*48	*52	56	73	*78	84	90		
CANT	11	52	63							
D	11	25	46	60						
DISTBD	11	*32	47	48	61	62				
F	6	9	52	63						
FLX	10	11	*83							
FLY	10	11	*84							
FUX	10	11	*85							
FUY	10	11	*86							
ID	9	11								
INT	78	79								
K	11	26								
KERFC	6									
KNOTS	11									
L	11	27	46	60						
LOG$	11	33	46	60						
NODEFC	9	11								
POSLEN	*33	38								
R	11	*26	31	32	33					
RAD	33	46	60							
RB	11	32								
RI	*25									
RM	*46	47	48							
RT	*60	61	62							
SQRT	48	62								
T	11	26	90							
TAN	33	46	60							
WAN	*73	74								
WT	*62	*63	67	73						

Subroutine WANE

```
C     SUBROUTINE 'WANE' LOCATES WANE DEFECTS ON THE BOARD                WANEC
C                                                                        WANEC
C                                                                        WANEC
         SUBROUTINE WANEC                                                WANEC
C                                                                        WANEC
C     THE BOARD EDGE HAS A NEW WANE DEFECT IF THE WIDTH DROPS BY 1/2     WANEC
C     INCH (1/4 INCH FOR EACH SIDE) FROM THAT AT THE MIDDLE.  THE BOARD  WANEC
C     IS SYMMETRICAL SO THE WANE DEFECTS OCCUR AT THE SAME PLACE ON      WANEC
C     BOTH EDGES OF THE BOARD                                            WANEC
C                                                                        WANEC
         IMPLICIT REAL (K,L)                                             WANEC
         INTEGER ID(22,2),NODEFC(2)                                      WANEC
         REAL FLX(26,2),FLY(26,2),FUX(26,2),FUY(26,2)                    WANEC
         COMMON /BOARD/ L,D,K,T,LOG$,RB,DISTBD,BDLEN,BOARDW,KNOTS,CANT,R,BF WANEC
        1       /DEFEC/ NODEFC,FLX,FLY,FUX,FUY,ID                        WANEC
C                                                                        WANEC
C                                                                        WANEC
         CON1 = TAN(RAD(LOG$))                                           WANEC
         CON2 = DISTBD**2                                                WANEC
         IF ((RB-SORT(CON2+BOARDW**2))/CON1.GE.BDLEN) RETURN             WANEC
         OLDH = BDLEN*2.                                                 WANEC
         V = 0.                                                          WANEC
C                                                                        WANEC
C     FIND THE PLACE ALONG THE BOARD WHERE THE HALF-WIDTH HAS DECREASED  WANEC
C     BY 1/4 INCH.  THE DEFECT ENDS HERE AND EXTENDS FROM THE END OF THE WANEC
C     LAST WANE DEFECT (OR FROM THE MIDDLE OF THE BOARD)                 WANEC
C                                                                        WANEC
      1 V = V+0.25                                                       WANEC
         SQR = SORT(CON2+(BOARDW-V)**2)                                  WANEC
         H = INT(AMIN1((RB-SQR)*4./CON1,BDLEN*4.))                       WANEC
         NODEFC(1) = NODEFC(1)+1                                         WANEC
         M = NODEFC(1)                                                   WANEC
C                                                                        WANEC
C     LIMIT NUMBER OF DEFECTS TO MAXIMUM ALLOWED BY GRADING PROGRAM (22) WANEC
C                                                                        WANEC
         IF (M.GT.22) GO TO 2                                            WANEC
C                                                                        WANEC
C     SAVE WANE COORDINATES FOR GRADING PROGRAM                          WANEC
C                                                                        WANEC
         FLY(M,1) = 0.                                                   WANEC
         FUY(M,1) = V*4.+1.                                              WANEC
         FUX(M,1) = H                                                    WANEC
         FLX(M,1) = OLDH                                                 WANEC
         ID(M,1) = 5                                                     WANEC
C                                                                        WANEC
C     SINCE THE BOARD IS SYMMETRICAL, USE THE PREVIOUSLY GENERATED HEIGHT WANEC
C     FOR THE WANE DEFECTS ON THE OTHER EDGE OF THE BOARD                WANEC
C                                                                        WANEC
      2 NODEFC(1) = NODEFC(1)+1                                          WANEC
         M = NODEFC(1)                                                   WANEC
         IF (M.GT.22) GO TO 3                                            WANEC
         FLY(M,1) = BOARDW*8.-V*4.-1.                                    WANEC
         FUY(M,1) = BOARDW*8.                                            WANEC
         FUX(M,1) = H                                                    WANEC
         FLX(M,1) = OLDH                                                 WANEC
         ID(M,1) = 5                                                     WANEC
C                                                                        WANEC
C     WHEN THE DEFECT EXTENDS TO THE END OF THE BOARD, EXIT              WANEC
C                                                                        WANEC
      3 IF (H.GE.BDLEN*4.) RETURN                                        WANEC
C                                                                        WANEC
C     THE NEXT WANE DEFECT BEGINS WHERE THIS ONE ENDS                    WANEC
C                                                                        WANEC
         OLDH = H                                                        WANEC
         GO TO 1                                                         WANEC
         RETURN                                                          WANEC
         END                                                             WANEC
```

```
                 *** STATEMENT NUMBERS ***

      1       *28    65
      2        36   *49
      3        51   *60

                 *** VARIABLES ***

AMIN1    30
BDLEN    14    20    21    30    60
BF       14
BOARDW   14    20    29    52    53
CANT     14
CON1    *18    20    30
CON2    *19    20    29
D        14
DISTBD   14    19
FLX      13    14   *43   *55
FLY      13    14   *40   *52
FUX      13    14   *42   *54
FUY      13    14   *41   *53
H       *30    42    54    60    64
ID       12    14   *44   *56
INT      30
K        14
KNOTS    14
L        14
LOG$     14    18
M       *32    36    40    41    42    43    44   *50    51    52    53    54
         55    56
NODEFC   12    14   *31    32   *49    50
OLDH    *21    43    55   *64
R        14
RAD      18
RB       14    20    30
SQR     *29    30
SORT     20    29
T        14
TAN      18
V       *22   *28    29    41    52
WANEC     4
```

Subroutine KNOT

```
C     SUBROUTINE 'KNOT' LOCATES THE KNOT DEFECTS ON THE BOARD FACE.
C     ANGLES ARE MEASURED CLOCKWISE WITH 0 DEGREES DEFINED AS THE
C     LINE FROM THE CENTER OF THE LOG PERPENDICULAR TO THE BOARD FACE
C
C
      SUBROUTINE KNOTC (KHIGH,KLEN,KTANG$)
C
      IMPLICIT REAL (K,L)
      INTEGER ID(22,2),NODEFC(2)
      REAL FLX(26,2),FLY(26,2),FUX(26,2),FUY(26,2)
      COMMON /DEFEC/ NODEFC,FLX,FLY,FUX,FUY,ID
     .       /BOARD/ L,D,K,T,LOG$,RB,DISTBD,BDLEN,BOARDW,KNOT$,CANT,R,BF
C
C
C     SEE IF THE KNOT IS LONG ENOUGH TO REACH THE BOARD FACE
C
      IF (KLEN.LT.DISTBD) RETURN
C
C     CALCULATE KNOT DEFECT HALF-LENGTH AS PROJECTED ON THE FACE, LOG
C     RADIUS AT KNOT HEIGHT, AND BOARD HALF-WIDTH ANGLE
C
      DY = DISTBD*TAN(RAD(KNOT$))
      LRADUS = RB-KHIGH*TAN(RAD(LOG$))
      BDWID$ = DEG(ATAN(BOARDW/DISTBD))
C
C     THE KNOT EFFECTIVELY ENDS AT THE OUTER EDGE OF THE LOG
C
      IF (KLEN.GT.LRADUS) KLEN = LRADUS
C
C     SINCE THE LOG IS TAPERED, THE OUTER BOARDS MAY NOT BE THE FULL LOG
C     LENGTH, SO KNOTS ABOVE THE END OF THE BOARD ARE NOT CONSIDERED
C
      IF (LRADUS.LT.DISTBD) RETURN
C
C     ANGLE OF ROTATION PUTS KNOT COMPLETELY OUTSIDE BOARD
C
      IF ((KTANG$.GT.180.).AND.((KTANG$+KNOT$).LT.(360.-BDWID$))) RETURN
      IF ((KTANG$.LE.180.).AND.((KTANG$-KNOT$).GT.BDWID$)) RETURN
C
C
C     FIND THE QUADRANT IN WHICH THE KNOT LIES (ASSUME QUADRANT ONE)
C
C     QUADRANT TWO
C
      IF ((KTANG$.LT.180.).AND.((KTANG$+KNOT$).GT.BDWID$)) GO TO 3
C
C     QUADRANT THREE
C
      IF ((KTANG$.GE.180.).AND.((KTANG$-KNOT$).LT.(360.-BDWID$)))
     . GO TO 4
C
C     QUADRANT FOUR
C
      IF (KTANG$.GT.270.) GO TO 5
C
C
C     QUADRANT ONE
C
C     SEE IF KNOT CENTER IS WITHIN ONE KNOT HALF-ANGLE OF 0 DEGREES
C
      IF (KTANG$-KNOT$.LT.0.) GO TO 2
C
C     CALCULATE DISTANCE TO FACE ALONG BOTH SIDES OF KNOT
C
      ANGBD1 = DISTBD/COS(RAD(KTANG$-KNOT$))
      ANGBD2 = DISTBD/COS(RAD(KTANG$+KNOT$))
C
C     KNOT IS NOT LONG ENOUGH TO REACH FACE
C
      IF (KLEN.LT.ANGBD1) RETURN
C
C     FIND INTERSECTION OF NEAR SIDE OF KNOT AND FACE
C
      XR = BOARDW-DISTBD*TAN(RAD(KTANG$-KNOT$))
      IF (KLEN.LT.ANGBD2) GO TO 1
C     FAR SIDE OF KNOT REACHES FACE
C
      XL = BOARDW-DISTBD*TAN(RAD(KTANG$+KNOT$))
      FULWID = (XR-XL)/2.
      GO TO 8
C
C     FAR SIDE OF KNOT DOES NOT REACH FACE
C
    1 XL = BOARDW-SQRT(KLEN**2-DISTBD**2)
      FULWID =
     .  DISTBD*TAN(RAD(KTANG$+KNOT$))-DISTBD*TAN(RAD(KTANG$-KNOT$))
      GO TO 8
C
C     KNOT IS WITHIN ONE KNOT HALF-ANGLE OF 0 DEGREES
C
    2 XL = AMIN1(SQRT(KLEN**2-DISTBD**2),DISTBD*TAN(RAD(KNOT$+KTANG$)))
      XL = BOARDW-XL
      XR = AMIN1(SQRT(KLEN**2-DISTBD**2),DISTBD*TAN(RAD(KNOT$-KTANG$)))
      XR = BOARDW+XR
C
C     KNOTS NEARLY PERPENDICULAR HAVE THEIR WIDTH ESTIMATED BY THEIR
C     LENGTH
C
      FULWID = DY*2.
      GO TO 8
C
C     QUADRANT TWO
C
    3 ANGBD1 = DISTBD/COS(RAD(KTANG$-KNOT$))
C
C     TOO SHORT TO REACH FACE
C
      IF (KLEN.LT.ANGBD1) RETURN
      XR = BOARDW-SQRT(ANGBD1**2-DISTBD**2)
      XL = BOARDW-SQRT(KLEN**2-DISTBD**2)
      IF ((KTANG$+KNOT$).GE.89.5)
     . FULWID = LRADUS-DISTBD*TAN(RAD(KTANG$-KNOT$))
      IF ((KTANG$+KNOT$).LT.89.5)
     . FULWID = AMIN1(LRADUS,DISTBD*TAN(RAD(KTANG$+KNOT$)))
     .    -DISTBD*TAN(RAD(KTANG$-KNOT$))
      GO TO 8
C
C     QUADRANT THREE
C
    4 ANGBD1 = DISTBD/COS(RAD(360.-KTANG$-KNOT$))
      IF (KLEN.LT.ANGBD1) RETURN
      XR = BOARDW+SQRT(KLEN**2-DISTBD**2)
      XL = BOARDW+SQRT(ANGBD1**2-DISTBD**2)
      IF ((KTANG$-KNOT$).LE.270.5)
     . FULWID = LRADUS-DISTBD*TAN(RAD(360.-KTANG$-KNOT$))
      IF ((KTANG$-KNOT$).GT.270.5)
     . FULWID = AMIN1(LRADUS,DISTBD*TAN(RAD(360.-KTANG$+KNOT$)))
     .    -DISTBD*TAN(RAD(360.-KTANG$-KNOT$))
      GO TO 8
C
C     QUADRANT FOUR
C
    5 IF (360.-KTANG$-KNOT$.LT.0.) GO TO 7
      ANGBD1 = DISTBD/COS(RAD(360.-KTANG$-KNOT$))
      ANGBD2 = DISTBD/COS(RAD(360.-KTANG$+KNOT$))
      IF (KLEN.LT.ANGBD1) RETURN
      XL = BOARDW+DISTBD*TAN(RAD(360.-KTANG$-KNOT$))
      IF (KLEN.LT.ANGBD2) GO TO 6
C
C     FAR SIDE OF KNOT REACHES FACE
C
      XR = BOARDW+DISTBD*TAN(RAD(360.-KTANG$+KNOT$))
      FULWID = (XR-XL)/2.
      GO TO 8
C
C     FAR SIDE OF KNOT DOES NOT REACH FACE
C
    6 XR = BOARDW+SQRT(KLEN**2-DISTBD**2)
      FULWID = DISTBD*TAN(RAD(360.-KTANG$+KNOT$))
     .  -DISTBD*TAN(RAD(360.-KTANG$-KNOT$))
      GO TO 8
C
C
```

```
C     KNOT IS WITHIN ONE KNOT HALF-ANGLE OF 0 DEGREES
C
    7 XL = AMIN1(SQRT(KLEN**2-DISTBD**2),
     .  DISTBD*TAN(PHD(KTANGS+KNOTS-360.)))
      XL = BOARDW-XL
      XR = AMIN1(SQRT(KLEN**2-DISTBD**2),
     .  DISTBD*TAN(PAD(360.-KTANG$+KNOTS)))
      XR = BOARDW+XR
      FULWID = DY*2.
C
C     IF A KNOT COMPLETELY INTERSECTS THE FACE, ITS LENGTH IS EXACTLY AS
C     MEASURED ON THE FACE.  OTHERWISE, ITS LENGTH IS ESTIMATED AS A
C     PROPORTION OF ITS PROJECTED LENGTH
C
    8 DY = AMIN1(DY,DY*((XR-XL)*2, FULWID))
      YB = KHIGH-DY
      YT = KHIGH+DY
C
C     DEFECT LOCATIONS FOUND.  CHECK THAT THEY DO NOT EXTEND BEYOND THE
C     BOARD EDGES AND CONVERT TO QUARTER INCH UNITS FOR THE GRADING
C     PROGRAM
C
      XL = AMAX0(0,INT(XL*4.))
      XR = AMIN0(INT(BOARDW*8.),INT(XR*4.+1.))
      YB = AMAX0(0,INT(YB*4.))
      YT = AMIN0(INT(BDLEN*4.),INT(YT*4.+1.))
C
C     LIMIT NUMBER OF DEFECTS TO MAXIMUM ALLOWED BY GRADING PROGRAM (22)
C
      NODEFC(1) = NODEFC(1)+1
      IF (NODEFC(1).GT.22) RETURN
C
C     SAVE KNOT COORDINATES FOR GRADING PROGRAM
C
      ID(NODEFC(1),1) = 3
      FLY(NODEFC(1),1) = XL
      FLX(NODEFC(1),1) = YB
      FUY(NODEFC(1),1) = XR
      FUX(NODEFC(1),1) = YT
      RETURN
      END
```

```
           142   148   158   161   177
CANT        11
COS         65    66   104   120   134   135
D           11
DEG         24
DISTBD      11    17    22    24    33    65    66    74    78    84    85    91
            93   104   109   110   111   113   120   122   123   124   126   134
           135   137   142   148   149   156   159
DY         *22    99   162  *168   169   170
FLX         10    11  *190
FLY         10    11  *189
FULWID     *79   *85   *99  *111  *113  *124  *126  *143  *149  *162   168
FUX         10    11  *192
FUY         10    11  *191
ID           9    11  *186
INT        176   177   178   179
K           11
KHIGH        6    23   169   170
KLEN         6    17   *28    70    75    84    91    93   108   110   121   122
           136   138   148   156   159
KNOTS       11    22    37    38    45    49    61    65    66    74    78    85
            91    93   104   111   113   120   124   126   133   134   135   137
           142   149   156   159
KNOTC        6
KTANG$       6    37    38    45    49    54    61    65    66    74    78    85
            91    93   104   111   113   120   124   126   133   134   135   137
           142   149   156   159
L           11
LOG$        11    23
LPADUS     *23    28    33   111   113   124   126
NODEFC       9    11  *183   184   188   189   190   191   192
P           11
PAD         22    23    65    66    74    78    85    91    93   104   111   113
           120   124   126   134   135   137   142   149   156   159
PE          11    23
SQRT        84    91    93   109   110   122   123   143   156   159
T           11
TAN         22    23    74    78    85    91    93   111   113   124   126   137
           142   149   156   159
XL         *78    79   *84   *91   *92  *110  *123  *137   143  *156  *158   168
          *176   189
XR         *74    79   *93   *94  *109  *122  *142   143  *148  *159  *161   166
          *177   191
YB        *169  *178   190
YT        *170  *179   192
```

*** STATEMENT NUMBERS ***

```
1          75   *84
2          11   131
3          45  *104
4          43  *120
5          51  *133
6          13  *140
7         133  *156
8          80    87   100   116   129   144   151  *168
```

*** VARIABLES ***

```
AMAX0      176   178
AMIN0      177   179
AMIN1       91    93   113   126   156   159   168
ANGBD1     *65    70  *104   108   109  *120   121   123  *134   136
ANGBD2     *66    75  *135   138
ATAN        24
BDLEN       11   179
BDWID$     *24    37    39    45    49
BF          11
BOARDW      11    24    74    78    84    92    94   109   110   122   123   137
```

Subroutine CORE

```
 1 C   SUBROUTINE 'CORE' LOCATES THE CORE DEFECT ON THE BOARD FACE         COREC
 2 C                                                                       COREC
 3 C                                                                       COREC
 4       SUBROUTINE COREC (CRADUS.COFSET.CANG$)                            COREC
 5 C                                                                       COREC
 6       IMPLICIT REAL (K.L)                                               COREC
 7       REAL FLX(26.2).FLY(26.2).FUX(26.2).FUY(26.2)                      COREC
 8       INTEGER NODEFC(2).ID(22.2)                                        COREC
 9       COMMON /BOARD/ L.D.K.T.LOG$.RB.DISTBD.BDLEN.BOARDW.KNOT$.CANT.R.BF COREC
10              /DEFEC/ NODEFC.FLX.FLY.FUX.FUY.ID                          COREC
11 C                                                                       COREC
12 C                                                                       COREC
13 C   FIND DISTANCE TO THE BOARD FACE FROM THE CENTER OF THE CORE. EXIT   COREC
14 C   IF THE BOARD FACE IS BEYOND THE CORE RADIUS                         COREC
15 C                                                                       COREC
16       TDIST = DISTBD-COFSET*COS(RAD(CANG$))                             COREC
17       IF (CRADUS.LE.TDIST) RETURN                                       COREC
18       NODEFC(1) = NODEFC(1)+1                                           COREC
19 C                                                                       COREC
20 C   LIMIT NUMBER OF DEFECTS TO MAXIMUM ALLOWED BY GRADING PROGRAM (22)  COREC
21 C                                                                       COREC
22       IF (NODEFC(1).GT.22) RETURN                                       COREC
23 C                                                                       COREC
24 C   LOCATE THE CENTER POINT OF THE INTERSECTION OF THE BOARD FACE AND   COREC
25 C   THE CORE DEFECT.  CALCULATE THE DISTANCE ON THE BOARD FROM THE      COREC
26 C   CENTER POINT TO THE EDGES OF THE CORE DEFECT.  THE CORE DEFECT      COREC
27 C   IS NOT TAPERED. SO IT WILL EXTEND THE FULL LENGTH OF THE BOARD      COREC
28 C                                                                       COREC
29       CONST = SQRT(CRADUS**2-TDIST**2)                                  COREC
30       TBDCTR = COFSET*SIN(RAD(CANG$))                                   COREC
31       XR = BOARDW+CONST-TBDCTR                                          COREC
32       XL = BOARDW-CONST-TBDCTR                                          COREC
33 C                                                                       COREC
34 C   SAVE CORE DEFECT COORDINATES FOR GRADING PROGRAM                    COREC
35 C                                                                       COREC
36       FUX(NODEFC(1).1) = INT(BDLEN*4.)                                  COREC
37       FLX(NODEFC(1).1) = 0.0                                            COREC
38       FUY(NODEFC(1).1) = AMIN0(INT(BOARDW*8.).INT(XR*4.+1.))            COREC
39       FLY(NODEFC(1).1) = AMAX0(0.INT(XL*4.))                            COREC
40       ID(NODEFC(1).1) = 7                                               COREC
41       RETURN                                                            COREC
42       END                                                               COREC
```

```
K          9
KNOTS      9
L          9
LOG$       9
NODEFC     8    9   *18   22   36   37   38   39   40
R          9
RAD       16   30
RB         9
SIN       30
SQRT      29
T          9
TBDCTR   *30   31   32
TDIST    *16   17   29
XL       *32   39
XR       *31   38
```

```
*** VARIABLES ***

AMAX0     39
AMIN0     38
BDLEN      9   36
BF         9
BOARDW     9   31   32   38
CANG$      4   16   30
CANT       9
COFSET     4   16   30
CONST    *29   31   32
COREC      4
COS       16
CRADUS     4   17   29
D          9
DISTBD     9   16
FLX        7    9  *37
FLY        7    9  *39
FUX        7    9  *36
FUY        7    9  *38
ID         8    9  *40
INT       36   38   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           GRADE
 2 C    GRADING PROGRAM                                                       GRADE
 3 C                                                                          GRADE
 4 C                                                                          GRADE
 5        SUBROUTINE GRADE (NPG)                                              GRADE
 6 C                                                                          GRADE
 7        INTEGER ID(22,2),NODEFC(2),IX(22)                                   GRADE
 8        REAL FLX(26,2),FLY(26,2),FUX(26,2),FUY(26,2)                        GRADE
 9        COMMON /DEFEC/ NODEFC,FLX,FLY,FUX,FUY,ID                            GRADE
10 C                                                                          GRADE
11 C                                                                          GRADE
12        M = NODEFC(1)                                                       GRADE
13        IF (M.GT.22) GO TO 3                                                GRADE
14        IF (M.EQ.0) GO TO 2                                                 GRADE
15        DO 1 J=1,M                                                          GRADE
16          IX(J) = ID(J,1)                                                   GRADE
17      1 CONTINUE                                                            GRADE
18      2 NPG = 1                                                             GRADE
19        CALL GRD (M,IX,NPG,SM)                                              GRADE
20        RETURN                                                              GRADE
21 C                                                                          GRADE
22 C    IF NUMBER OF DEFECTS IS MORE THAN GRADING PROGRAM CAN HANDLE,         GRADE
23 C    THE GRADE IS 3A/3B                                                    GRADE
24 C                                                                          GRADE
25      3 NPG = 5                                                             GRADE
26        WRITE (6,100)                                                       GRADE
27    100 FORMAT ('     DEFECT LIMIT EXCEEDED')                                GRADE
28        RETURN                                                              GRADE
29        END                                                                 GRADE
```

```
                  *** 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
GRD        19
ID          7     9    16
IX          7   *16    19
J         *15    16
M         *12    13    14    15    19
NODEFC      7     9    12
NPG         5   *18    19   *25
SM         19
```

Subroutine PRICE

```
 1 C    SUBROUTINE 'PRICE' DETERMINES THE BOARD GRADE BASED ON THE GRADES      PRICEC
 2 C    OF BOTH SIDES (OLD,NPG) AND CALCULATES THE BOARD VALUE                 PRICEC
 3 C                                                                          PRICEC
 4 C                                                                          PRICEC
 5        SUBROUTINE PRICEC (OLD,NPG,TVAL,PERC)                               PRICEC
 6 C                                                                          PRICEC
 7        IMPLICIT REAL (K,L)                                                 PRICEC
 8        INTEGER OLD                                                         PRICEC
 9        REAL PERC(5)                                                        PRICEC
10        COMMON /PRICE/ FAS,SEL,ONEC,TWOC,THRB                               PRICEC
11              /BOARD/ L,D,K,T,LOGS,RB,DISTBD,BDLEN,BOARDW,KNOTS,CANT,R,BF   PRICEC
12 C                                                                          PRICEC
13 C                                                                          PRICEC
14 C    DETERMINE THE BOARD GRADE                                             PRICEC
15 C                                                                          PRICEC
16        IF (OLD.EQ.5.OR.NPG.EQ.5) GO TO 3                                   PRICEC
17        IF (OLD.EQ.2.OR.NPG.EQ.2) GO TO 1                                   PRICEC
18        IF (OLD.EQ.NPG) GO TO 4                                             PRICEC
19        NPG = MAX0(OLD,NPG)                                                 PRICEC
20        IF (NPG.EQ.4) GO TO 8                                               PRICEC
21        NPG = 2                                                             PRICEC
22        GO TO 4                                                             PRICEC
23      1 IF (OLD.EQ.4.OR.NPG.EQ.4) GO TO 2                                   PRICEC
24        NPG = 3                                                             PRICEC
25        GO TO 4                                                             PRICEC
26      2 NPG = 4                                                             PRICEC
27        GO TO 4                                                             PRICEC
28      3 NPG = 5                                                             PRICEC
29 C                                                                          PRICEC
30 C    AND CALCULATE THE BOARD VALUE                                         PRICEC
31 C                                                                          PRICEC
32      4 GO TO (5,6,7,8,9), NPG                                              PRICEC
33      5 V = BF*FAS                                                          PRICEC
34        GO TO 10                                                            PRICEC
35      6 V = BF*SEL                                                          PRICEC
36        GO TO 10                                                            PRICEC
37      7 V = BF*ONEC                                                         PRICEC
38        GO TO 10                                                            PRICEC
39      8 V = BF*TWOC                                                         PRICEC
40        GO TO 10                                                            PRICEC
41      9 V = BF*THRB                                                         PRICEC
42 C                                                                          PRICEC
43 C    INCREASE SURFACE MEASURE IN THIS GRADE AND THE TOTAL LOG VALUE        PRICEC
44 C                                                                          PRICEC
45     10 PERC(NPG) = BF+PERC(NPG)                                            PRICEC
46        TVAL = TVAL+V                                                       PRICEC
47        RETURN                                                              PRICEC
48        END                                                                 PRICEC
```

```
                  *** STATEMENT NUMBERS ***

  1     17   *23
  2     23   *26
  3     16   *28
  4     18    22    25    27   *32
  5     32   *33
  6     32   *35
  7     32   *37
  8     20    32   *39
  9     32   *41
 10     34    36    38    40   *45
```

*** VARIABLES ***

BDLEN	10											
BF	10	33	35	37	39	41	45					
BOARDW	10											
CANT	10											
D	10											
DISTBD	10											
FAS	10	33										
K	10											
KNOTS	10											
L	10											
LOGS	10											
MAXB	19											
NPG	5	16	17	18	*19	20	*21	23	*24	*26	*28	32
	45											
OLD	5	8	16	17	18	19	23					
ONEC	10	37										
PERC	5	9	*45									
PRICEC	5											
R	10											
RB	10											
SEL	10	35										
T	10											
THRB	10	41										
TVAL	5	*46										
TWDC	10	39										
V	*33	*35	*37	*39	*41	46						

Function DEG

```
C   FUNCTION 'DEG' CONVERTS ITS ARGUMENT (IN RADIANS) TO DEGREES      DEG
C                                                                     DEG
C                                                                     DEG
      FUNCTION DEG(A)                                                 DEG
C                                                                     DEG
C                                                                     DEG
      DEG = A*180.0/3.141592                                          DEG
      RETURN                                                          DEG
      END                                                             DEG
```

*** VARIABLES ***

A	4	7
DEG	4	*7

Function RAD

```
C   FUNCTION 'RAD' CONVERTS ITS ARGUMENT (IN DEGREES) TO RADIANS        RAD
C                                                                       RAD
C                                                                       RAD
      FUNCTION RAD(A)                                                   RAD
C                                                                       RAD
C                                                                       RAD
      RAD = A*3.141592/180.0                                            RAD
      RETURN                                                            RAD
      END                                                               RAD
```

```
                        *** VARIABLES ***

A              4    7
RAD            4   *7
```

Decision Sawing (DECID)

```
C     'DECID' SIMULATES THE DECISIONS OF A SKILLED HUMAN SAWYER.  AFTER
C     THE LOG IS SQUARED AT MID-LENGTH, BOARDS ARE CUT FROM THE 'BEST'
C     FACE UNTIL A DROP IN GRADE OCCURS.  A NEW 'BEST' FACE IS THEN FOUND
C     AND THE PROCESS REPEATED FOR THE REST OF THE LOG
C
C
      IMPLICIT REAL (K,L)
      REAL FLX(26,2),FLY(26,2),FUX(26,2),FUY(26,2),PERC(5),BOARDW(2),
     .     DISTBD(4),SMF(4),RCUT(4),ANGLE(40),HEIGHT(40),KTLEN(40),
     .     R(4),DAT*8
      INTEGER ID(22,2),NODEFC(2),CMPLET(4),F,GDORDR(50),CDUENS(50),
     .        GRDCOM(4),GDBEST,IX(22),OLD
      COMMON /BOARD/ L,D,K,T,LOGS,RB,DISTBD,BDLEN,BOARDW,KNOTS,R,RCUT,BF
     .       /PRICF/ FAS,SEL,ONEC,TWOC,THRB
     .       /DECID/ CMPLET,GDBEST,GRDCOM,SMBEST,SMF
     .       /DEFEC/ NODEFC,FLX,FLY,FUX,FUY,ID
C
C
C     THIS PROGRAM ASSUMES INPUT FROM UNIT 5 AND OUTPUTS TO UNITS 6 AND 8
C
C     GET LOG DIAMETER (IN.), LOG LENGTH (IN.), KERF SIZE (IN.), AND
C     BOARD THICKNESS (IN.)
C
    1 READ (5,116,END = 23) D
      READ (5,116) L
      READ (5,117) K
      READ (5,116) T
C
C     GET KNOT HALF-ANGLE (DEGREES) AND LOG HALF-TAPER (DEGREES)
C
      READ (5,116) KNOTS
      READ (5,116) LOGS
C
C     GET CORE DEFECT PARAMETERS: DIAMETER (IN.), LINEAR OFFSET (IN.),
C     AND ANGULAR OFFSET (DEGREES)
C
      READ (5,100) CORDIA
      READ (5,100) COFSET
      READ (5,100) CANGS
C
C     GET PRICE PER BOARD FOOT FOR EACH GRADE: NUMBER OF KNOTS
C
      READ (5,123) FAS,SEL,ONEC,TWOC,THRB
      READ (5,109) MAX
C
C     GET DATE AND CALCULATE RADIUS OF CORE DEFECT
C
      CALL DATE (DAT)
      CRADUS = CORDIA/2.
C
C     GET KNOT ANGLES, HEIGHTS, AND LENGTHS
C
      DO 2 J=1,MAX
        READ (5,102) ANGLE(J),HEIGHT(J)
    2 CONTINUE
      DO 3 J=1,MAX
        READ (5,100) KTLEN(J)
    3 CONTINUE
C
C     CALCULATE LOG RADIUS AT TOP AND BOTTOM: TOTAL LOG CUBIC FEET
C
      RI = D/2.
      RB = RI+L*TAN(RAD(LOGS))
      LVOL = 1/3.*3.141593*(RB**2+RB*RI+RI**2)*L/144.
C
C     OUTPUT HEADER
C
      WRITE (6,118) DAT
      WRITE (6,119) L,D,LOGS,LVOL
      WRITE (6,108) MAX
      DO 4 J=1,MAX
        WRITE (6,101) ANGLE(J),HEIGHT(J),KTLEN(J)
    4 CONTINUE
      WRITE (6,120) KNOTS
      WRITE (6,115) CRADUS,COFSET,CANGS
      WRITE (6,121) K,T
      WRITE (6,122) FAS,SEL,ONEC,TWOC,THRB
      WRITE (8,200) L,D,MAX,CRADUS,COFSET,CANGS,LVOL,K,T,KNOTS,LOGS
      WRITE (8,201) (KTLEN(J),J=1,MAX)
      WRITE (8,201) (HEIGHT(J),J=1,MAX)
      WRITE (8,201) (ANGLE(J),J=1,MAX)
      WRITE (8,202) FAS,SEL,ONEC,TWOC,THRB
C
C     INITIALIZE VARIABLES FOR CALCULATING HIGH, LOW, AND AVERAGE
C     YIELDS FOR EACH LOG
C
      HIGHS = 0.
      SMALLS = 999.
      AVGS = 0.
      HITOT = 0.
      SMTOT = 999.
      AVTOTS = 0.
C
C     SAW LOG IN 12 DIFFERENT ROTATIONAL POSITIONS, 15 DEGREES APART, TO
C     DETERMINE THE ORIENTATION GIVING THE HIGHEST YIELD
C
      DO 22 INX=1,12
C
C     INITIALIZE FOR EACH POSITION
C
        ISQUAR = 1
C
C     INITIALIZE PORTION OF LOG TO BE CUT...  THIS SETS THE DEPTH OF
C     THE FIRST CUT ON EACH OF THE FOUR FACES OF THE LOG.  THE LOG IS CUT
C     FROM THE OUTSIDE IN AND THE SLAB FACES MAY BE CUT TO THE CENTER
C     CANT WHICH CONTAINS EXACTLY FOUR BOARDS
C
        RTBCUT = RB-2.*T-1.5*K
        NBRDS = INT(RTBCUT/(T+K))
        EXCESS = RTBCUT-NBRDS*(T+K)
        RTBCUT = RTBCUT-EXCESS
        R(1) = RTBCUT+EXCESS-T-K
        R(2) = RTBCUT+2.*T+1.5*K+EXCESS
        R(3) = R(1)
        R(4) = R(2)
        DO 5 J=1,4
          DISTBD(J) = RB
          RCUT(J) = EXCESS-T-K
          GRDCOM(J) = 0
          SMF(J) = 0.0
          CMPLET(J) = 0
    5   CONTINUE
        SMBEST =  0.0
        BF = 0.0
        TVAL = 0.0
        GDBEST = 0
        NBD = 0
        DO 6 J=1,5
          PERC(J) = 0.0
    6   CONTINUE
        F = 0
C
C     CUT FACES SEQUENTIALLY UNTIL LOG IS SQUARED AT MID-LENGTH
C
    7   F = F+1
        IF (F.EQ.5) F = 1
C
C     CUT A BOARD FROM THIS FACE.  IF THE FACE IS COMPLETELY CUT, PICK
C     A NEW FACE TO CUT
C
    8   CALL KERFD (&18,ISQUAR,F,0)
        NBD = NBD+1
C
C     FOR BOTH SIDES OF THE BOARD...
C
        DO 12 I=1,2
C
C     INITIALIZE DEFECT ARRAYS, EXCEPT F..(26,1), WHICH HOLDS THE BOARD
C     DIMENSIONS
C
```

```
      DO 9 J=1,25
        FLX(J,1) = -9999.0
        FUX(J,1) = -9999.0
        FLY(J,1) = -9999.0
        FUY(J,1) = -9999.0
    9 CONTINUE
      DO 10 J=1,22
        ID(J,1) = 0
   10 CONTINUE
      NODEFC(1) = 0
C
C  PUT IN WANE DEFECTS
C
      CALL WANED (F)
C
C  PUT IN KNOTS
C
      DO 11 J=1,MA
        CALL KNOTID (HEIGHT(J),ANGLE(J),KTLEN(J),F)
   11 CONTINUE
C
C  PUT IN CORE DEFECT
C
      CALL CORED (CRADUS,COFSET,CANGS,F)
C
C  AND GRADE THE FACE
C
      CALL GRADE (NPG)
C
C  SAVE GRADE OF FIRST SIDE OF BOARD
C
      IF (I.EQ.1) OLD = NPG
C
C  SUBTRACT BOARD THICKNESS AND PROCESS THE INNER FACE
C
      DISTBD(F) = DISTBD(F)-T
   12 CONTINUE
C
C  IGNORE THE LAST BOARD TAKEN
C
      DISTBD(F) = DISTBD(F)+T
C
C  DETERMINE BOARD GRADE AND VALUE BASED ON GRADES OF BOTH FACES
C
      CALL FRICED (OLD,NPG,TVAL,PERC)
C
C  SAVE BOARD GRADE AND SURFACE MEASURE
C
      GRDCOM(F) = NPG
      SMF(F) = BF
C
C  SAVE SEQUENCE OF FACES CUT AND THEIR GRADES
C
      COUENS(NBD) = F
      GDORDR(NBD) = NPG
C
C  SEE IF LOG HAS BEEN SQUARED AT MID-LENGTH
C
      IF (ISQUAR.NE.0) GO TO 7
C
C  ONCE THE LOG HAS BEEN SQUARED, BOARDS ARE CUT FROM THE 'BEST' FACE
C  UNTIL A DROP IN GRADE OCCURS.  AFTER REGRADING THE AFFECTED FACES,
C  A NEW 'BEST' FACE IS FOUND.  NOTE THAT THE 'BEST' FACE IS INITIALLY
C  BETTER THAN FAS, FORCING A GRADE DROP ON THE FIRST BOARD
C
      IF (GRDCOM(F).GT.GDBEST) GO TO 13
C
C  OTHERWISE, SAVE BOARD GRADE AND SURFACE MEASURE AND CONTINUE CUTTING
C  ON THE SAME FACE
C
      GDBEST = GRDCOM(F)
      SMBEST = SMF(F)
      GO TO 8
C
C  REGRADE THE AFFECTED FACES, I.E., THE ONE JUST CUT AND THE ONES ON
C  EITHER SIDE OF IT
C
   13 F = F-2
      IF (F.EQ.-1) F = 3
      IF (F.EQ.0) F = 4
      DO 17 II=1,3
        F = F+1
        IF (F.EQ.5) F = 1
C
C  INITIALIZE DEFECT ARRAYS
C
          DO 14 JJ=1,26
            FUX(JJ,1) = -9999.0
            FLX(JJ,1) = -9999.0
            FUY(JJ,1) = -9999.0
            FLY(JJ,1) = -9999.0
   14     CONTINUE
          DO 15 JJ=1,22
            ID(JJ,1) = 0
   15     CONTINUE
          NODEFC(1) = 0
C
C  FIND EACH AFFECTED FACE
C
          CALL KERFD (&17,ISQUAR,F,1)
C
C  PUT IN WANE
C
          CALL WANED (F)
C
C  PUT IN KNOTS
C
          DO 16 J=1,MAX
            CALL KNOTID (HEIGHT(J),ANGLE(J),KTLEN(J),F)
   16     CONTINUE
C
C  PUT IN CORE DEFECT
C
          CALL CORED (CRADUS,COFSET,CANGS,F)
C
C  AND GRADE THE FACE
C
          CALL GRADE (NPG)
C
C  SAVE THE GRADE AND SURFACE MEASURE OF EACH AFFECTED FACE
C
          GRDCOM(F) = NPG
          SMF(F) = BF
C
C  IGNORE THE CUT JUST TAKEN
C
          RCUT(F) = RCUT(F)-T-K
          DISTBD(F) = DISTBD(F)+K
   17 CONTINUE
C
C  RETURN TO THE FACE ORIGINALLY BEING CUT
C
      F = F-1
      IF (F.EQ.0) F = 4
C
C  DECIDE WHICH FACE TO CUT NEXT
C
   18 CALL DECIDE (F,&19)
      GO TO 8
C
C  PRINT RESULTS FOR THIS POSITION
C
   19 IROTAT = INX*15-15
      WRITE (6,111) IROTAT
      WRITE (6,113) (PERC(J),J=1,5)
      TOTS = PERC(1)+PERC(2)+PERC(3)+PERC(4)+PERC(5)
      WRITE (6,112) TOTS
      WRITE (8,203) IROTAT
      WRITE (8,204) (PERC(J),J=1,5)
      WRITE (8,204) TOTS
      DO 20 J=1,5
        PERC(J) = (PERC(J)/LVOL)*100.
   20 CONTINUE
      TOTS = (TOTS/LVOL)*100.
      WRITE (6,114) TOTS
      WRITE (6,103) (ANGLE(J),J=1,MAX)
```

```
307         WRITE (6.104) (PERC(J).J=1.5).TVAL                          DECID
308         WRITE (6.110)                                               DECID
309         WRITE (6.107) (COUENS(J).J=1.NBD)                           DECID
310         WRITE (6.107) (GDORDR(J).J=1.NBD)                           DECID
311         WRITE (8.205) (PERC(J).J=1.5)                               DECID
312         WRITE (8.204) TVAL.TOTS                                     DECID
313 C                                                                   DECID
314 C   SAVE HIGH. LOW. AND AVERAGE YIELDS FOR THIS LOG                 DECID
315 C                                                                   DECID
316         AVTOTS = AVTOTS+TOTS                                        DECID
317         AVG$ = AVG$+TVAL                                            DECID
318         IF (TVAL.GT.HIGH$) HITOT = TOTS                             DECID
319         IF (TVAL.LT.SMALL$) SMTOT = TOTS                            DECID
320         IF (TVAL.GT.HIGH$) HIGH$ = TVAL                             DECID
321         IF (TVAL.LT.SMALL$) SMALL$ = TVAL                           DECID
322 C                                                                   DECID
323 C   ROTATE LOG BY 15 DEGREES AND REPROCESS                          DECID
324 C                                                                   DECID
325         DO 21 J=1.MAX                                               DECID
326           ANGLE(J) = AMOD(ANGLE(J)+15.0.360.0)                      DECID
327     21  CONTINUE                                                    DECID
328         CANG$ = AMOD(CANG$+15..360.)                                DECID
329     22 CONTINUE                                                     DECID
330 C                                                                   DECID
331 C   PRINT HIGH. LOW. AND AVERAGE YIELDS FOR THIS LOG                DECID
332 C                                                                   DECID
333       AVTOTS = AVTOTS/12.                                           DECID
334       AVG$ = AVG$/12.                                               DECID
335       WRITE (6.105) HIGH$.AVG$.SMALL$                               DECID
336       WRITE (6.106) HITOT.AVTOTS.SMTOT                              DECID
337 C                                                                   DECID
338 C   LOG COMPLETELY PROCESSED. READ PARAMETERS FOR NEXT LOG          DECID
339 C                                                                   DECID
340       GO TO 1                                                       DECID
341 C                                                                   DECID
342     23 STOP                                                         DECID
343 C                                                                   DECID
344 C   FORMATS FOR DEVICE 6 (PRINTER)                                  DECID
345 C                                                                   DECID
346    100 FORMAT (F6.3)                                                DECID
347    101 FORMAT (3(2X.F6.2))                                          DECID
348    102 FORMAT (2F6.2)                                               DECID
349    103 FORMAT (' KNOT ANGLES: '.15F7.2./.13X.15F7.2)                DECID
350    104 FORMAT (' % OF AVAILABLE SURFACE MEASURE IN EACH GRADE: '.5F5.1.  DECID
351      .  ' LOG VALUE IS $'.F6.2)                                     DECID
352    105 FORMAT (' $ YIELD:  HIGH '.F6.2.'  AVERAGE '.F6.2.'  LOW '.F6.2)  DECID
353    106 FORMAT (' % YIELD:  HIGH $ FACE '.F6.2.'  AVERAGE '.F6.2.    DECID
354      .  '  LOW $ FACE '.F6.2)                                       DECID
355    107 FORMAT (' '.40I3)                                            DECID
356    108 FORMAT ('1THERE ARE'.I3.'KNOTS IN THIS LOG. THEY ARE'/       DECID
357      .  4X.'ANGLE  HEIGHT  LENGTH')                                 DECID
358    109 FORMAT (I2)                                                  DECID
359    110 FORMAT (' SEQUENCE OF LOG FACES CUT AND THEIR BOARD GRADES') DECID
360    111 FORMAT (2(/).' ROTATION IS'.I4)                              DECID
361    112 FORMAT (' TOTAL SURFACE MEASURE IS'.F8.2)                    DECID
362    113 FORMAT (' SURFACE MEASURE PER GRADE IS'.5F7.2)               DECID
363    114 FORMAT (' TOTAL PERCENT YIELD IS'.F8.2)                      DECID
364    115 FORMAT (' CORE DEFECT RADIUS '.F6.3.'  LINEAR OFFSET '.F6.3. DECID
365      .  '  ANGULAR OFFSET '.F6.3)                                   DECID
366    116 FORMAT (F5.1)                                                DECID
367    117 FORMAT (F5.3)                                                DECID
368    118 FORMAT (' DECISION SAWING METHOD '.A8)                       DECID
369    119 FORMAT (' LOG PARAMETERS:  LENGTH'.F6.1.'  DIA'.F6.1.'  TAPER'.  DECID
370      .  F6.4.' DEGREES.  LOG VOLUME: '.F8.2.' CUBIC FEET')          DECID
371    120 FORMAT (' ANGLES MEASURED FROM ZERO DEGREES = EAST FOR A '.  DECID
372      .  'VERTICALLY CUTTING SAW.  KNOT TAPER'.F6.2)                 DECID
373    121 FORMAT (' CUTTING PARAMETERS:  KERF'.F6.4.'  BOARD THICKNESS'  DECID
374      .  F5.2)                                                       DECID
375    122 FORMAT (' PRICES PER BOARD FOOT: '.5F10.4)                   DECID
376    123 FORMAT (5F10.4)                                              DECID
377 C                                                                   DECID
378 C   FORMATS FOR DEVICE 8 (STORAGE MEDIUM)                           DECID
379 C                                                                   DECID
380    200 FORMAT (F6.1.F5.1.I3.3F5.1.F8.2.F6.3.F4.1.F6.2.F5.3.F6.3)    DECID
381    201 FORMAT (3(10F7.2/))                                          DECID
382    202 FORMAT (5F7.4)                                               DECID
383    203 FORMAT (I3)                                                  DECID
384    204 FORMAT (5F8.2)                                               DECID
385    205 FORMAT (10F6.2)                                              DECID
386        END                                                          DECID
```

```
                    *** STATEMENT NUMBERS ***

     1    *24   340
     2     53   *55
     3     56   *58
     4     71   *73
     5    116  *122
     6    128  *130
     7   *135   289
     8   *141   223   289
     9    151  *156
    10    157  *159
    11    168  *170
    12    146  *187
    13    216  *228
    14    237  *242
    15    243  *245
    16    253  *260
    17    231   250  *279
    18    141  *288
    19    288  *293
    20    301  *303
    21    325  *327
    22     97  *329
    23     24  *342
   100     37    38    39    57  *346
   101     72  *347
   102     54  *348
   103    306  *349
   104    307  *350
   105    335  *352
   106    336  *353
   107    309   310  *355
   108     70  *356
   109     44  *358
   110    308  *359
   111    294  *360
   112    297  *361
   113    295  *362
   114    305  *363
   115     75  *364
   116     24    25    27    31    32  *366
   117     26  *367
   118     68  *368
   119     69  *369
   120     74  *371
   121     76  *373
   122     77  *375
   123     43  *376
   200     78  *380
   201     79    80    81  *381
   202     82  *382
   203    298  *383
   204    299   300   312  *384
   205    311  *385

                    *** VARIABLES ***

AMOD      326   328
ANGLE       6   *54    72    81   169   259   306  *326
AVG$      *89  *317  *334   335
AVTOTS    *92  *316  *333   336
```

```
BDLEN       13
BF          13  *124   200   273
BOARDW       8    13
CANGS      *33    75    78   174   264  *328
CMPLET      11    13  *121
COFSET     *38    75    78   174   264
COPDIA     *37    49
COPED      174   264
COMEMS      11  *204   309
CRADUS     *40    75    78   174   264
D           13   *24    62    69    78
DAT          8    48    68
DATE        48
DECIDE     288
DISTBD       3    13  *117  *186  *191  *278
EXCESS    *110   111   112   113   118
F           11  *131  *135  *136   141   164   169   174   186   191   199   200
           204   216   221   222  *228  *229  *230  *232  *233   250   254   259
           264   272   273   277   278  *283  *284   288
FAS         13   *43    77    82
FLX          9    13  *152  *239
FLY          8    13  *154  *241
FUX          8    13  *153  *238
FUY          8    13  *155  *240
GDBEST      11    13  *126   216  *221
GDORDR      11  *205   310
GRADE      178   268
GRDCOM      11    13  *119  *199   216   221  *272
HEIGHT       8   *54    72    80   169   259
HIGHS      *87   318  *320   335
HITOT      *90  *318   336
I         *145   182
ID          11    13  *158  *244
II        *231
INT        109
INX        *97   293
IPOTAT    *293   294   298
ISQUAR    *101   141   209   250
IX          11
J          *53   *54   *56   *57   *71    72    79    80    81  *116   117   118
           119   120   121  *128   129  *151   152   153   154   155  *157   158
          *168   169  *258   259   295   299  *301   302   306   307   309   310
           311  *325   326
JJ        *237   238   239   240   241  *243   244
K           17   *26    76    78   108   109   110   112   113   118   277   278
KEPFD      141   250
KNOTS       13   *31    74    78
KNOTD      169   259
KTLEN        8   *57    72    79   169   259
L           13   *25    63    64    69    78
LOGS        13   *32    63    69    78
LVOL       *64    69    78   302   304
MAX        *44    53    56    70    71    78    79    80    81   160   258   306
           325
NBD       *127  *142   204   205   309   310
NBPDS     *109   110
NODFFC      11    13  *160  *246
NPC        179   182   195   199   205   268   272
OLD         11  *182   195
ONEC        13   *43    77    82
PERC         8  *129   195   295   296   299  *302   307   311
PRICED     195
P            8    13  *112  *113  *114  *115
RAD         63
RB          13   *63    64   108   117
RCUT         8    13  *118  *277
RI         *62    63    64
RTBCUT    *108   109   110  *111   112   113
SEL         13   *43    77    82
SMALLS     *88   319  *321   335
SMBEST      13  *123  *222
SMF          8    13  *120  *208   222  *273
SMTOT      *91  *319   336
T           13   *27    76    78   108   109   110   112   113   118   186   191
           277
TAN         63
THRB        13   *43    77    82
TOTS      *296   297   300  *304   305   312   316   318   319

TVAL      *125   195   307   312   317   318   319   320   321
TUDC        13   *43    77    82
WANED      164   254
```

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
  6       SUBROUTINE KERFD (*,ISQUAR,F,IREGRD)
  7 C
  8       IMPLICIT REAL (K,L)
  9       INTEGER F,FRT,FLFT,ID(22,2),CMPLET(4),NODEFC(2),GRDCOM(4)
 10       REAL FLX(26,2),FLY(26,2),FUX(26,2),FUY(26,2),BOARDW(2),WT(2),
 11      .     DISTBD(4),RCUT(4),SMF(4),R(4)
 12       COMMON /BOARD/ L,D,K,T,LOG$,RB,DISTBD,BDLEN,BOARDW,KNOT$,R,RCUT,BF
 13      .       /DECID/ CMPLET,GDBEST,GRDCOM,SMBEST,SMF
 14      .       /DEFEC/ NODEFC,FLX,FLY,FUX,FUY,ID
 15 C
 16 C
 17 C   ALL BOARDS MUST BE AT LEAST 48 INCHES LONG, 2.5 INCHES WIDE AT
 18 C   THE TOP, AND 3 INCHES WIDE AT THE CENTER.  IN ADDITION, THE TOTAL
 19 C   AMOUNT OF WANE FROM THE CENTER TO THE TOP IS LIMITED TO 4 INCHES.
 20 C   THE BOARD IS CUT BACK IN ONE FOOT INCREMENTS IF THE WIDTH OR WANE
 21 C   TESTS FAIL.  IF THE LENGTH TEST FAILS, ANOTHER BOARD IS CUT FROM
 22 C   THIS FACE
 23 C
 24       RI = D/2.
 25 C
 26 C   FIND THE TWO ADJACENT FACES
 27 C
 28       FLFT = F-1
 29       IF (FLFT.EQ.0) FLFT = 4
 30       FRT = F+1
 31       IF (FRT.EQ.5) FRT = 1
 32 C
 33 C   FIND OUTER FACE OF BOARD
 34 C
 35     1 RCUT(F) = RCUT(F)+T+K
 36       DISTBD(F) = RB-RCUT(F)
 37       BDLEN = L
 38 C
 39 C   IF THIS FACE OF THE LOG IS COMPLETELY CUT, EXIT
 40 C
 41       IF (RCUT(F).GT.R(F)) GO TO 6
 42       POSLEN = RCUT(F)*TAN(RAD(90.-LOG$))
 43 C
 44 C   CHECK FOR BOARDS SHORTER THAN THE LOG DUE TO LOG TAPER
 45 C
 46       IF (POSLEN.LT.BDLEN) BDLEN = POSLEN
 47 C
 48 C   BOARDS MUST BE AT LEAST 4 FEET LONG
 49 C
 50     2 IF (BDLEN.LT.48.) GO TO 5
 51 C
 52 C   DETERMINE HALF-WIDTHS AT MID-LENGTH OF BOARD.  SINCE THE BOARD IS
 53 C   NOT NECESSARILY SYMMETRICAL ABOUT A LINE FROM THE CENTER OF THE
 54 C   LOG PERPENDICULAR TO THE BOARD FACE, THE WIDTH OF EACH SIDE MUST
 55 C   BE DETERMINED SEPARATELY
 56 C
 57       RM = RI+(L-BDLEN/2.)*TAN(RAD(LOG$))
 58       IF (DISTBD(F).GT.RM) GO TO 4
 59       BOARDW(1) = SQRT(RM**2-DISTAD(F)**2)
 60       BOARDW(2) = BOARDW(1)
 61 C
 62 C   THE BOARD WIDTH MAY BE LIMITED BY THE ADJACENT FACES
 63 C
 64       DSTLFT = DISTBD(FLFT)-K
 65       DSTRGT = DISTBD(FRT)-K
 66       BOARDW(1) = AMIN1(DSTLFT,BOARDW(1))
 67       BOARDW(2) = AMIN1(DSTRGT,BOARDW(2))
 68 C
 69 C   IS LOG SQUARED AT MID-LENGTH?
 70 C
 71       IF (BOARDW(2).EQ.DSTRGT) ISQUAR = 0
 72 C
 73 C   BOARD MUST BE AT LEAST 3 INCHES WIDE AT MID-LENGTH
 74 C
 75       IF ((BOARDW(1)+BOARDW(2)).LT.3.) GO TO 4
 76 C   DETERMINE HALF-WIDTHS AT TOP OF BOARD
 77 C
 78       RT = RI+(L-BDLEN)*TAN(RAD(LOG$))
 79       IF (DISTBD(F).GT.RT) GO TO 4
 80       WT(1) = SQRT(RT**2-DISTBD(F)**2)
 81       WT(2) = WT(1)
 82       WT(1) = AMIN1(DSTLFT,WT(1))
 83       WT(2) = AMIN1(DSTRGT,WT(2))
 84 C
 85 C   BOARD MUST BE AT LEAST 2.5 INCHES WIDE AT THE TOP
 86 C
 87       IF ((WT(1)+WT(2)).LT.2.5) GO TO 4
 88 C
 89 C   LIMIT THE TOTAL WANE TO AVOID GENERATING MORE DEFECTS THAN THE
 90 C   GRADING PROGRAM CAN HANDLE.  TOTAL WIDTH OF WANE MUST BE LESS
 91 C   THAN 4 INCHES
 92 C
 93       WAN = BOARDW(1)+BOARDW(2)-WT(1)-WT(2)
 94       IF (WAN.GT.4.) GO TO 4
 95 C
 96 C   CONVERT BOARD HALF-WIDTHS AND LENGTH TO EVEN QUARTER INCHES
 97 C
 98       DO 3 J=1,2
 99         BOARDW(J) = INT(BOARDW(J)*4.)/4.
100     3 CONTINUE
101       BDLEN = INT(BDLEN*4.)/4.
102 C
103 C   DEFINE BOARD EDGES IN QUARTER INCH UNITS
104 C
105       FLX(26,1) = BDLEN*4.
106       FUX(26,1) =  0.0
107       FLY(26,1) = (BOARDW(1)+BOARDW(2))*4.
108       FUY(26,1) =  0.0
109 C
110 C   CALCULATE UNROUNDED BOARD FEET
111 C
112       BF = BDLEN*(BOARDW(1)+BOARDW(2))*T/144.
113       PFTURN
114 C
115 C   CUT BACK BOARD BY ONE FOOT
116 C
117     4 BDLEN = BDLEN-12.
118       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     5 IF (IREGRD.EQ.1) GO TO 6
124       GO TO 1
125 C
126 C   FACE HAS BEEN COMPLETELY CUT.  DO NOT CUT ANY MORE BOARDS FROM IT
127 C
128     6 CMPLET(F) = 1
129       GRDCOM(F) = 7
130 C
131 C   IGNORE THE CUT JUST ATTEMPTED
132 C
133       RCUT(F) = RCUT(F)-T-K
134       DISTBD(F) = RB-RCUT(F)-T
135       SMF(F) = 0.0
136       RETURN1
137       END
```

```
                    ***  STATEMENT NUMBERS  ***

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

```
     4        58    75    79    87    94  *117
     5        58  *123
     6        41   123  *128


                              ***  VARIABLES  ***


AMIN1         66    67    82    83
BDLEN         12   *37   *46    58    57    78  *101   105   112  *117
BF            12  *112
BOARDW        10    12   *59   *60   *66   *67    71    75    93   *99   107   112
CMPLET         9    12  *128
D             12    24
DISTBD        10    12   *36    58    59    64    65    79    80  *134
DSTLFT       *64    66    82
DSTRGT       *65    67    71    83
F              6     9    28    38    35    36    41    42    58    59    79    80
             128   129   133   134   135
FLFT           9   *28   *29    64
FLX           10    12  *105
FLY           10    12  *107
FRT            9   *30   *31    65
FUX           10    12  *106
FUY           10    12  *108
GDBEST        12
GRDCOM         9    12  *129
ID             9    12
INT           99   101
IREGRD         6   123
ISQUAR         6   *71
J            *98    99
K             12    35    64    65   133
KERFD          6
KNOTS         12
L             12    37    57    78
LOGS          12    42    57    78
NODEFC         9    12
POSLEN       *42    46
R             10    12    41
RAD           42    57    78
RB            12    36   134
RCUT          10    12   *35    36    41    42  *133   134
RI           *24    57    78
RM           *57    58    59
RT           *78    79    80
SMBEST        12
SMF           10    12  *135
SQRT          59    90
T             12    35   112   133   134
TAN           42    57    78
WAN          *93    94
WT            10   *80   *81   *82   *83    87    93
```

Subroutine WANE

```
C     SUBROUTINE 'WANE' LOCATES WANE DEFECTS ON THE BOARD                WANED
C                                                                        WANED
C                                                                        WANED
        SUBROUTINE WANED (F)                                             WANED
C                                                                        WANED
C     THE BOAPD EDGE HAS A NEW WANE DEFECT IF THE HALF-WIDTH ON THAT SIDE WANED
C     DROPS BY 1/4 INCH.  SINCE THE BOARD IS NOT NECESSARILY SYMMETRICAL WANED
C     THE WIDTH OF THE WANE DEFECTS MUST BE CALCULATED SEPARATELY FOR EACH WANED
C     SIDE                                                               WANED
C                                                                        WANED
        IMPLICIT REAL (K,L)                                              WANED
        INTEGER ID(22,2),NODEFC(2),F                                     WANED
        REAL BOARDW(2),DISTBD(4),FLY(26,2),FLX(26,2),FUY(26,2),FUX(26,2), WANED
       .     RCUT(4),R(4)                                                WANED
        COMMON /BOARD/ L,D,K,T,LOGS,RB,DISTBD,BDLEN,BOARDW,KNOTS,R,RCUT,BF WANED
       .       /DEFEC/ NODEFC,FLX,FLY,FUX,FUY,ID                         WANED
C                                                                        WANED
C                                                                        WANED
        CON1 = TAN(RAD(LOGS))                                            WANED
        CON2 = DISTBD(F)**2                                              WANED
        IF ((RB-SQRT(CON2+BOARDW(2)**2))/CON1.GE.BDLEN) GO TO 3          WANED
        OLDH = BDLEN*2.                                                  WANED
        V = 0.                                                           WANED
C                                                                        WANED
C                                                                        WANED
C     FIND THE PLACE ALONG THE BOARD WHERE THE HALF-WIDTH HAS DECREASED  WANED
C     BY 1/4 INCH.  THE DEFECT ENDS HERE AND EXTENDS FROM THE END OF THE WANED
C     LAST WANE DEFECT (OR FROM THE MIDDLE OF THE BOARD)                 WANED
C                                                                        WANED
      1 V = V+0.25                                                       WANED
        SQR = SQRT(CON2+(BOARDW(2)-V)**2)                                WANED
        H = INT(AMIN1((RB-SQR)*4./CON1,BDLEN*4.))                        WANED
        NODEFC(1) = NODEFC(1)+1                                          WANED
        M = NODEFC(1)                                                    WANED
C                                                                        WANED
C     LIMIT NUMBEP OF DEFECTS TO MAXIMUM ALLOWED BY GRADING PROGRAM (22) WANED
C                                                                        WANED
        IF (M.GT.22) GO TO 2                                             WANED
C                                                                        WANED
C     SAVE WANE COORDINATES FOR GRADING PROGRAM                          WANED
C                                                                        WANED
        FLY(M,1) = 0.                                                    WANED
        FUY(M,1) = V*4.+1.                                               WANED
        FUX(M,1) = H                                                     WANED
        FLX(M,1) = OLDH                                                  WANED
        ID(M,1) = 5                                                      WANED
C                                                                        WANED
C     WHEN WANE EXTENDS TO END OF BOARD, THIS EDGE IS DONE               WANED
C                                                                        WANED
      2 IF (H.GE.BDLEN*4.) GO TO 3                                       WANED
C                                                                        WANED
C     THE NEXT WANE DEFECT BEGINS WHERE THIS ONE ENDS                    WANED
C                                                                        WANED
        OLDH = H                                                         WANED
        GO TO 1                                                          WANED
C                                                                        WANED
C                                                                        WANED
C     REPEAT FOR OTHER EDGE OF BOARD                                     WANED
C                                                                        WANED
      3 IF ((RB-SQRT(CON2+BOARDW(1)**2))/CON1.GE.BDLEN) RETURN           WANED
        OLDH = BDLEN*2.                                                  WANED
        V = 0.                                                           WANED
C                                                                        WANED
C     FIND END OF WANE DEFECT                                            WANED
C                                                                        WANED
      4 V = V+0.25                                                       WANED
        SQR = SQRT(CON2+(BOARDW(1)-V)**2)                                WANED
        H = INT(AMIN1((RB-SQR)*4./CON1,BDLEN*4.))                        WANED
        NODEFC(1) = NODEFC(1)+1                                          WANED
        M = NODEFC(1)                                                    WANED
C                                                                        WANED
C     LIMIT NUMBER OF DEFECTS TO MAXIMUM ALLOWED BY GRADING PROGRAM (22) WANED
C                                                                        WANED
        IF (M.GT.22) GO TO 5                                             WANED
C                                                                        WANED
```

```
C     SAVE WANE COORDINATES FOR GRADING PROGRAM                         WANED
C                                                                       WANED
        FUY(M,1) = (BOARDW(1)+BOARDW(2))*4.                             WANED
        FLY(M,1) = (BOARDW(1)+BOARDW(2))*4.-V*4.-1.                     WANED
        FUX(M,1) = H                                                    WANED
        FLX(M,1) = OLDH                                                 WANED
        ID(M,1) = 5                                                     WANED
C                                                                       WANED
C     WHEN THE DEFECT EXTENDS TO THE END OF THE BOARD, EXIT             WANED
C                                                                       WANED
      5 IF (H.GE.BDLEN*4.) RETURN                                       WANED
C                                                                       WANED
C     THE NEXT WANE DEFECT BEGINS WHERE THIS ONE ENDS                   WANED
C                                                                       WANED
        OLDH = H                                                        WANED
        GO TO 4                                                         WANED
        RETURN                                                          WANED
        END                                                             WANED
```

*** STATEMENT NUMBERS ***

1	*50	55	
2	39	*50	
3	21	50	*60
4	*66	91	
5	74	*86	

*** VARIABLES ***

AMIN1	32	68									
BDLEN	15	21	22	32	50	60	61	68	86		
BF	15										
BOARDW	13	15	21	31	60	67	78	79			
CON1	*19	21	32	60	68						
CON2	*20	21	31	60	67						
D	15										
DISTBD	13	15	20								
F	4	12	20								
FLX	13	15	*45	*81							
FLY	13	15	*42	*79							
FUX	13	15	*44	*80							
FUY	13	15	*43	*78							
H	*32	44	50	54	*60	80	86	90			
ID	12	15	*46	*82							
INT	32	68									
K	15										
KNOTS	15										
L	15										
LOGS	15	19									
M	*34	38	42	43	44	45	46	*70	74	78	79	80
	81	82									
NODEFC	12	15	*33	34	*69	70					
OLDH	*22	45	*54	*61	81	*90					
R	13	15									
RAD	19										
RB	15	21	32	60	68						
RCUT	13	15									
SOR	*31	32	*67	68							
SOPT	21	31	60	67							
T	15										
TAN	19										
V	*23	*30	31	43	*62	*66	67	79			
WANED	4										

Subroutine KNOT

```
C     SUBROUTINE 'KNOT' LOCATES THE KNOT DEFECTS ON THE BOARD FACE.     KNOTD
C     ANGLES ARE MEASURED CLOCKWISE WITH 0 DEGREES DEFINED AS THE       KNOTD
C     LINE FROM THE CENTER OF THE LOG PERPENDICULAR TO THE BOARD FACE   KNOTD
C                                                                       KNOTD
C                                                                       KNOTD
        SUBROUTINE KNOTD (KHIGH,KTANGS,KLEN,F)                          KNOTD
C                                                                       KNOTD
        IMPLICIT REAL (K,L)                                             KNOTD
        INTEGER ID(22,2),NODEFC(2),F                                    KNOTD
        REAL BOARDW(2),DISTBD(4),RCUT(4),BDWIDS(2),R(4),FLX(26,2),      KNOTD
             FLY(26,2),FUX(26,2),FUY(26,2)                              KNOTD
        COMMON /DEFEC/ NODEFC,FLX,FLY,FUX,FUY,ID                        KNOTD
               /BOARD/ L,D,K,T,LOGS,RB,DISTBD,BDLEN,BOARDW,KNOTS,R,RCUT,BF KNOTD
C                                                                       KNOTD
C                                                                       KNOTD
C     SEE IF THE KNOT IS LONG ENOUGH TO REACH THE BOARD FACE            KNOTD
C                                                                       KNOTD
        IF (KLEN.LT.DISTBD(F)) RETURN                                   KNOTD
C                                                                       KNOTD
C     CALCULATE BOARD HALF-WIDTH ANGLES, KNOT DEFECT HALF-LENGTH AS     KNOTD
C     PROJECTED ON THE FACE, AND LOG RADIUS AT KNOT HEIGHT              KNOTD
C                                                                       KNOTD
        BDWIDS(1) = DEG(ATAN(BOARDW(1)/DISTBD(F)))                      KNOTD
        BDWIDS(2) = DEG(ATAN(BOARDW(2)/DISTBD(F)))                      KNOTD
        DY = DISTBD(F)*TAN(RAD(KNOTS))                                  KNOTD
        LRADUS = RB-KHIGH*TAN(RAD(LOGS))                                KNOTD
C                                                                       KNOTD
C     THE KNOT EFFECTIVELY ENDS AT THE OUTER EDGE OF THE LOG            KNOTD
C                                                                       KNOTD
        IF (KLEN.GT.LRADUS) KLEN = LRADUS                               KNOTD
C                                                                       KNOTD
C     SINCE THE LOG IS TAPERED, THE OUTER BOARDS MAY NOT BE THE FULL LOG KNOTD
C     LENGTH, SO KNOTS ABOVE THE END OF THE BOARD ARE NOT CONSIDERED    KNOTD
C                                                                       KNOTD
        IF (LRADUS.LT.DISTBD(F)) RETURN                                 KNOTD
C                                                                       KNOTD
C     ROTATE THE LOG TO THE CURRENT FACE                                KNOTD
C                                                                       KNOTD
        IF (F.EQ.2) KTANGS = AMOD(KTANGS+270.,360.)                     KNOTD
        IF (F.EQ.3) KTANGS = AMOD(KTANGS+180.,360.)                     KNOTD
        IF (F.EQ.4) KTANGS = AMOD(KTANGS+90.,360.)                      KNOTD
C                                                                       KNOTD
C     ANGLE OF ROTATION PUTS KNOT COMPLETELY OUTSIDE BOARD              KNOTD
C                                                                       KNOTD
        IF ((KTANGS.GT.180.).AND.((KTANGS+KNOTS).LT.(360.-BDWIDS(1))))  KNOTD
         GO TO 9                                                        KNOTD
        IF ((KTANGS.LE.180.).AND.((KTANGS-KNOTS).GT.BDWIDS(2))) GO TO 9 KNOTD
C                                                                       KNOTD
C                                                                       KNOTD
C     FIND THE QUADRANT IN WHICH THE KNOT LIES (ASSUME QUADRANT ONE)    KNOTD
C                                                                       KNOTD
C     QUADRANT TWO                                                      KNOTD
C                                                                       KNOTD
        IF ((KTANGS.LT.180.).AND.((KTANGS+KNOTS).GT.BDWIDS(2))) GO TO 3 KNOTD
C                                                                       KNOTD
C     QUADRANT THREE                                                    KNOTD
C                                                                       KNOTD
        IF ((KTANGS.GE.180.).AND.((KTANGS-KNOTS).LT.(360.-BDWIDS(1))))  KNOTD
         GO TO 4                                                        KNOTD
C                                                                       KNOTD
C     QUADRANT FOUR                                                     KNOTD
C                                                                       KNOTD
        IF (KTANGS.GT.270.) GO TO 5                                     KNOTD
C                                                                       KNOTD
C                                                                       KNOTD
C     QUADRANT ONE                                                      KNOTD
C                                                                       KNOTD
C     SEE IF KNOT CENTER IS WITHIN ONE KNOT HALF-ANGLE OF 0 DEGREES     KNOTD
C                                                                       KNOTD
        IF (KTANGS-KNOTS.LT.0.) GO TO 2                                 KNOTD
C                                                                       KNOTD
C     CALCULATE DISTANCE TO FACE ALONG BOTH SIDES OF KNOT               KNOTD
C                                                                       KNOTD
        ANGBD1 = DISTBD(F)/COS(RAD(KTANGS-KNOTS))                       KNOTD
        ANGBD2 = DISTBD(F)/COS(RAD(KTANGS+KNOTS))                       KNOTD
```

```
C     KNOT IS NOT LONG ENOUGH TO REACH FACE
C
        IF (KLEN.LT.ANGBD1) GO TO 9
C
C     FIND INTERSECTION OF NEAR SIDE OF KNOT AND FACE
C
        XR = BOARDW(2)-DISTBD(F)*TAN(RAD(KTANG$-KNOT$))
        IF (KLEN.LT.ANGBD2) GO TO 1
C
C     FAR SIDE OF KNOT REACHES FACE
C
        XL = BOARDW(2)-DISTBD(F)*TAN(RAD(KTANG$+KNOT$))
        FULWID = (XR-XL)/2.
        GO TO 8
C
C     FAR SIDE OF KNOT DOES NOT REACH FACE
C
      1 XL = BOARDW(2)-SQRT(KLEN**2-DISTBD(F)**2)
        FULWID = DISTBD(F)*TAN(RAD(KTANG$+KNOT$))
     .  -DISTBD(F)*TAN(RAD(KTANG$-KNOT$))
        GO TO 9
C
C     KNOT IS WITHIN ONE KNOT HALF-ANGLE OF 0 DEGREES
C
      2 XL = AMIN1(SQRT(KLEN**2-DISTBD(F)**2),
     .  DISTBD(F)*TAN(RAD(KTANG$+KNOT$)))
        XL = BOARDW(2)-XL
        XR = AMIN1(SQRT(KLEN**2-DISTBD(F)**2),
     .  DISTBD(F)*TAN(RAD(KNOT$-KTANG$)))
        XR = BOARDW(2)+XR
C
C     KNOTS NEARLY PERPENDICULAR HAVE THEIR WIDTH ESTIMATED BY THEIR
C     LENGTH
C
        FULWID = DY*2.
        GO TO 8
C
C     QUADRANT TWO
C
      3 ANGBD1 = DISTBD(F)/COS(RAD(KTANG$-KNOT$))
C
C     TOO SHORT TO REACH FACE
C
        IF (KLEN.LT.ANGBD1) GO TO 9
        XR = BOARDW(2)-SQRT(ANGBD1**2-DISTBD(F)**2)
        XL = BOARDW(2)-SQRT(KLEN**2-DISTBD(F)**2)
        IF ((KTANG$+KNOT$).GE.89.5)
     .  FULWID = LRADUS-DISTBD(F)*TAN(RAD(KTANG$-KNOT$))
        IF ((KTANG$+KNOT$).LT.89.5)
     .  FULWID = AMIN1(LRADUS,DISTBD(F)*TAN(RAD(KTANG$+KNOT$)))
     .     -DISTBD(F)*TAN(RAD(KTANG$-KNOT$))
        GO TO 8
C
C     QUADRANT THREE
C
      4 ANGBD1 = DISTBD(F)/COS(RAD(360.-KTANG$-KNOT$))
        IF (KLEN.LT.ANGBD1) GO TO 9
        XR = BOARDW(2)+SQRT(KLEN**2-DISTBD(F)**2)
        XL = BOARDW(2)+SQRT(ANGBD1**2-DISTBD(F)**2)
        IF ((KTANG$-KNOT$).LE.270.5)
     .  FULWID = LRADUS-DISTBD(F)*TAN(RAD(360.-KTANG$-KNOT$))
        IF ((KTANG$-KNOT$).GT.270.5)
     .  FULWID = AMIN1(LRADUS,DISTBD(F)*TAN(RAD(360.-KTANG$+KNOT$)))
     .     -DISTBD(F)*TAN(RAD(360.-KTANG$-KNOT$))
        GO TO 8
C
C     QUADRANT FOUR
C
      5 IF (360.-KTANG$-KNOT$.LT.0.) GO TO 7
        ANGBD1 = DISTBD(F)/COS(RAD(360.-KTANG$-KNOT$))
        ANGBD2 = DISTBD(F)/COS(RAD(360.-KTANG$+KNOT$))
        IF (KLEN.LT.ANGBD1) GO TO 9
        XL = BOARDW(2)+DISTBD(F)*TAN(RAD(360.-KTANG$-KNOT$))
        IF (KLEN.LT.ANGBD2) GO TO 6
C
C     FAR SIDE OF KNOT REACHES FACE
C
        XR = BOARDW(2)+DISTBD(F)*TAN(RAD(360.-KTANG$+KNOT$))
        FULWID = (XR-XL)/2.
        GO TO 8
C
C     FAR SIDE OF KNOT DOES NOT REACH FACE
C
      6 XR = BOARDW(2)+SQRT(KLEN**2-DISTBD(F)**2)
        FULWID = DISTBD(F)*TAN(RAD(360.-KTANG$+KNOT$))
     .  -DISTBD(F)*TAN(RAD(360.-KTANG$-KNOT$))
        GO TO 8
C
C     KNOT IS WITHIN ONE KNOT HALF-ANGLE OF 0 DEGREES
C
      7 XL = AMIN1(SQRT(KLEN**2-DISTBD(F)**2),
     .  DISTBD(F)*TAN(RAD(KTANG$+KNOT$-360.)))
        XL = BOARDW(2)-XL
        XR = AMIN1(SQRT(KLEN**2-DISTBD(F)**2),
     .  DISTBD(F)*TAN(RAD(360.-KTANG$+KNOT$)))
        XR = BOARDW(2)+XR
        FULWID = DY*2.
C
C     IF A KNOT COMPLETELY INTERSECTS THE FACE, ITS LENGTH IS EXACTLY AS
C     MEASURED ON THE FACE.  OTHERWISE, ITS LENGTH IS ESTIMATED AS A
C     PROPORTION OF ITS PROJECTED LENGTH
C
      8 DY = AMIN1(DY,DY*(XR-XL)*2./FULWID))
        YB = KHIGH-DY
        YT = KHIGH+DY
C
C     DEFECT LOCATIONS FOUND.  CHECK THAT THEY DO NOT EXTEND BEYOND THE
C     BOARD EDGES AND CONVERT TO QUARTER INCH UNITS FOR THE GRADING
C     PROGRAM
C
        XL = AMAX0(0,INT(XL*4.))
        XR = AMIN0(INT((BOARDW(1)+BOARDW(2))*4.),INT(XR*4.+1.))
        YB = AMAX0(0,INT(YB*4.))
        YT = AMIN0(INT(BDLEN*4.),INT(YT*4.+1.))
C
C     LIMIT NUMBER OF DEFECTS TO MAXIMUM ALLOWED BY GRADING PROGRAM (22)
C
        NODEFC(1) = NODEFC(1) +1
        IF (NODEFC(1).GT.22) RETURN
C
C     SAVE KNOT COORDINATES FOR GRADING PROGRAM
C
        FLX(NODEFC(1),1) = XL
        FUX(NODEFC(1),1) = XR
        FLY(NODEFC(1),1) = YB
        FUY(NODEFC(1),1) = YT
        IDNODEFC(1),1) = 3
C
C     RETURN KNOT ANGLES TO ORIGINAL VALUES
C
      9 IF (F.EQ.2) KTANG$ = AMOD(KTANG$+90.,360.)
        IF (F.EQ.3) KTANG$ = AMOD(KTANG$+180.,360.)
        IF (F.EQ.4) KTANG$ = AMOD(KTANG$+270.,360.)
        RETURN
        END
```

*** STATEMENT NUMBERS ***

1	83	*93					
2	70	*100					
3	54	*115					
4	58	*131					
5	63	*144					
6	149	*159					
7	144	*166					
8	89	96	111	127	140	155	162	*178

```
      9         45    47    78   119   132   147  *206

                        ***  VARIABLES  ***

AMAXØ      186   188
AMINØ      187   189
AMIN1      100   124   137   166   169   178
AMINI      103
AMOD        39    40    41   206   207   208
ANGBD1     *74    78  *115   119   120  *131   132   134  *145   147
ANGBD2     *75    83  *146   149
ATAN        23    24
BDLEN       12   189
BDWTS       10   *23   *24    45    47    54    58
BF          12
BOARDW      10    12    23    24    82    87    93   102   105   120   121   133
           134   140   153   159   160   171   187
COS         74    75   115   131   145   146
D           12
DEG         23    24
DISTBD      10    12    18    23    24    25    35    74    75    82    87    93
            94   100   103   115   120   121   122   124   131   133   134   135
           137   145   146   148   153   159   160   166   169
DY         *25   110   172  *178   179   180
F            6     9    18    23    24    25    35    39    40    41    74    75
            82    87    93    94   100   103   115   120   121   122   124   131
           133   134   135   137   145   146   148   153   159   160   166   169
           206   207   208
FLX         10    12  *200
FLY         10    12  *198
FULWID     *83   *94  *110  *122  *124  *135  *137  *154  *160  *172   178
FUX         10    12  *201
FUY         10    12  *199
ID           9    12  *202
INT        186   187   188   189
K           12
KHIGH        6    26   179   180
KLEN         6    13   *30    78    83    93   100   103   119   121   132   133
           147   149   159   166   169
KNOTS       12    25    45    47    54    58    70    74    75    82    87    94
           100   103   115   122   124   131   135   137   144   145   146   148
           153   160   166   169
KNOTD        6
KTANGS       6   *39   *40   *41    45    47    54    58    63    70    74    75
            82    87    94   100   103   115   122   124   131   135   137   144
           145   146   148   153   160   166   169  *206  *207  *208
L           12
LOGS        12    26
LRADUS     *26    30    35   122   124   135   137
NODEFC       9    12  *193   194   198   199   200   201   202
R           10    12
RAD         25    26    74    75    82    87    94   100   103   115   122   124
           131   135   137   145   146   148   153   160   166   169
RB          12    26
RCUT        10    12
SQRT        93   100   103   120   121   133   134   159   166   169
T           12
TAN         25    26    82    87    94   100   103   122   124   135   137   148
           153   160   166   169
XL         *87    88   *93  *100  *102  *121  *134  *148   154  *166  *168   178
          *186   198
XR         *82    88  *103  *105  *120  *133  *153   154  *159  *169  *171   178
          *107   199
YB        *179  *188   200
YT        *180  *189   201
```

Subroutine CORE

```
C     SUBROUTINE 'CORE' LOCATES THE CORE DEFECT ON THE BOARD FACE        CORED
C                                                                        CORED
C                                                                        CORED
      SUBROUTINE CORED (CRADUS,COFSET,CANGS,F)                           CORED
C                                                                        CORED
      IMPLICIT REAL(K,L)                                                 CORED
      INTEGER ID(22,2),NODEFC(2),F                                       CORED
      REAL FLX(26,2),FLY(26,2),FUX(26,2),FUY(26,2),DISTBD(4),BOARDW(2),  CORED
     .     RCUT(4),R(4)                                                  CORED
      COMMON /BOARD/ L,D,K,T,LOGS,RB,DISTBD,BDLEN,BOARDW,KNOTS,R,RCUT,BF CORED
     .       /DEFEC/ NODEFC,FLX,FLY,FUX,FUY,ID                           CORED
C                                                                        CORED
C                                                                        CORED
C     ROTATE LOG TO THE FACE BEING CUT                                   CORED
C                                                                        CORED
      TCANGS = CANGS                                                     CORED
      IF (F.EQ.2) TCANGS = AMOD(CANGS+270.,360.)                         CORED
      IF (F.EQ.3) TCANGS = AMOD(CANGS+180.,360.)                         CORED
      IF (F.EQ.4) TCANGS = AMOD(CANGS+90.,360.)                          CORED
C                                                                        CORED
C     FIND DISTANCE TO THE BOARD FACE FROM THE CENTER OF THE CORE. EXIT  CORED
C     IF THE BOARD FACE IS BEYOND THE CORE RADIUS                        CORED
C                                                                        CORED
      TDIST = DISTBD(F)-COFSET*COS(RAD(TCANGS))                          CORED
      IF (CRADUS.LE.TDIST) RETURN                                        CORED
      NODEFC(1) = NODEFC(1)+1                                            CORED
C                                                                        CORED
C     LIMIT NUMBER OF DEFECTS TO MAXIMUM ALLOWED BY GRADING PROGRAM (22) CORED
C                                                                        CORED
      IF (NODEFC(1).GT.22) RETURN                                        CORED
C                                                                        CORED
C     LOCATE THE CENTER POINT OF THE INTERSECTION OF THE BOARD FACE AND  CORED
C     THE CORE DEFECT.  CALCULATE THE DISTANCE ON THE BOARD FROM THE     CORED
C     CENTER POINT TO THE EDGES OF THE CORE DEFECT.  THE CORE DEFECT     CORED
C     IS NOT TAPERED. SO IT WILL EXTEND THE FULL LENGTH OF THE BOARD     CORED
C                                                                        CORED
      CONST = SQRT(CRADUS**2-TDIST**2)                                   CORED
      TBDCTR = COFSET*SIN(RAD(TCANGS))                                   CORED
      XP = BOARDW(2)+CONST-TBDCTP                                        CORED
      XL = BOARDW(2)-CONST-TBDCTP                                        CORED
C                                                                        CORED
C     SAVE CORE DEFECT COORDINATES FOR GRADING PROGRAM                   CORED
C                                                                        CORED
      FUX(NODEFC(1),1) = INT(BDLEN*4.)                                   CORED
      FLX(NODEFC(1),1) = 0.0                                             CORED
      FUY(NODEFC(1),1) = AMINO(INT((BOARDW(1)+BOARDW(2))*4.),            CORED
     .   INT(XP*4.+1.))                                                  CORED
      FLY(NODEFC(1),1) = AMAXØ(0,INT(XL*4.))                             CORED
      ID(NODEFC(1),1) = 7                                                CORED
      RETURN                                                             CORED
      END                                                                CORED


                        ***  VARIABLES  ***

AMAXØ       48
AMINØ       46
AMOD        17    18    19
BDLEN       10    44
BF          10
BOARDW       8    10    39    40    46
CANGS        4    16    17    18    19
COFSET       4    24    38
CONST      *37    39    40
CORED        4
COS         24
```

```
CRADUS      4    25    37
D          10
DISTBD      8    10    24
F           4     7    17    10    15    24
FLX         8    10   *45
FLY         8    10   *48
FUX         8    10   *44
FUY         8    10   *46
ID          7    10   *49
INT        44    46    48
K          10
KNOTS      10
L          10
LOGS       10
NODEFC      7    10   *26    30    44    45    46    48    49
R           8    10
RAD        24    38
RB         10
RCUT        8    10
SIN        38
SORT       37
T          10
TBDCTR    *38    39    40
TCANGS    *16   *17   *18   *19    24    38
TDIST     *24    25    37
XL        *40    48
XR        *33    46
```

Subroutine GRADE

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

*** 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
GRD       19
ID         7     9    16
IX         7   *16    19
J        *15    16
M        *12    13    14    15    19
NODEFC     7     9    12
NPG        5   *18    19   *25
SM        19
```

Subroutine PRICE

```
 1 C    SUBROUTINE 'PRICE' DETERMINES THE BOARD GRADE BASED ON THE GRADES    PRICED
 2 C    OF BOTH SIDES (OLD,NPG) AND CALCULATES THE BOARD VALUE               PRICED
 3 C                                                                         PRICED
 4 C                                                                         PRICED
 5        SUBROUTINE PRICED (OLD,NPG,TVAL,PERC)                              PRICED
 6 C                                                                         PRICED
 7        IMPLICIT REAL (K,L)                                                PRICED
 8        INTEGER OLD                                                        PRICED
 9        REAL PERC(5),BOARDW(2),R(4),DISTBD(4),RCUT(4)                      PRICED
10        COMMON /BOARD/ L,D,K,T,LOG$,RB,DISTBD,BDLEN,BOARDW,KNOT$,R,RCUT,BF PRICED
11       &       /PRICF/ FAS,SEL,ONEC,TWOC,THRB                              PRICED
12 C                                                                         PRICED
13 C                                                                         PRICED
14 C    DETERMINE THE BOARD GRADE                                            PRICED
15 C                                                                         PRICED
16        IF (OLD.EQ.5.OR.NPG.EQ.5) GO TO 3                                  PRICED
17        IF (OLD.EQ.2.OR.NPG.EQ.2) GO TO 1                                  PRICED
18        IF (OLD.EQ.NPG) GO TO 4                                            PRICED
19        NPG = MAX0(OLD,NPG)                                                PRICED
20        IF (NPG.EQ.4) GO TO 8                                              PRICED
21        NPG = 2                                                            PRICED
22        GO TO 6                                                            PRICED
23      1 IF (OLD.EQ.4.OR.NPG.EQ.4) GO TO 2                                  PRICED
24        NPG = 3                                                            PRICED
25        GO TO 4                                                            PRICED
26      2 NPG = 4                                                            PRICED
27        GO TO 4                                                            PRICED
28      3 NPG = 5                                                            PRICED
29 C                                                                         PRICED
30 C    AND CALCULATE THE BOARD VALUE                                        PRICED
31 C                                                                         PRICED
32      4 GO TO (5,6,7,8,9), NPG                                             PRICED
33      5 V = BF*FAS                                                         PRICED
34        GO TO 10                                                           PRICED
35      6 V = BF*SEL                                                         PRICED
36        GO TO 10                                                           PRICED
37      7 V = BF*ONEC                                                        PRICED
38        GO TO 10                                                           PRICED
39      8 V = BF*TWOC                                                        PRICED
40        GO TO 10                                                           PRICED
41      9 V = BF*THRB                                                        PRICED
42 C                                                                         PRICED
43 C    INCREASE SURFACE MEASURE IN THIS GRADE AND THE TOTAL LOG VALUE       PRICED
44 C                                                                         PRICED
45     10 PERC(NPG) = BF+PERC(NPG)                                           PRICED
46        TVAL = TVAL+V                                                      PRICED
47        RETURN                                                             PRICED
48        END                                                                PRICED
```

*** VARIABLES ***

```
BDLEN    10
BF       10   33   35   37   39   41   45
BOARDW    9   10
D        10
DISTBD    9   10
FAS      10   33
K        10
KNOT$    10
L        10
LOG$     10
MAX0     19
NPG       5   16   17   18  *19   20  *21   23  *24  *26  *28   32
         45
OLD       5    8   16   17   18   19   23
ONEC     10   37
PERC      5    9  *45
PRICED    5
R         9   10
RB       10
RCUT      9   10
SEL      10   35
T        10
THRB     10   41
TVAL      5  *46
TWOC     10   39
V       *33  *35  *37  *39  *41   46
```

*** STATEMENT NUMBERS ***

```
 1    17  *23
 2    23  *26
 3    16  *28
 4    18   25   27  *32
 5    32  *33
 6    22   32  *35
 7    32  *37
 8    20   32  *39
 9    32  *41
10    34   36   38   40  *45
```

Subroutine DECIDE

```
 1 C    SUBROUTINE 'DECIDE' FINDS THE LOG FACE THAT SHOULD BE CUT NEXT.    DECIDE
 2 C    USUALLY THIS IS THE FACE WITH THE HIGHEST GRADE. BUT, IF TWO OR    DECIDE
 3 C    MORE FACES HAVE THE SAME GRADE, THE FACE WITH THE LARGEST SURFACE  DECIDE
 4 C    MEASURE IS USED.  IF TWO OR MORE FACES HAVE THE SAME GRADE AND     DECIDE
 5 C    SURFACE MEASURE, THE LAST FACE CONSIDERED IS USED                  DECIDE
 6 C                                                                       DECIDE
 7 C                                                                       DECIDE
 8         SUBROUTINE DECIDE (F,*)                                         DECIDE
 9 C                                                                       DECIDE
10         IMPLICIT INTEGER (C,F,G)                                        DECIDE
11         DIMENSION CMPLET(4),GRDCOM(4),SMF(4)                            DECIDE
12         COMMON /DECID/ CMPLET,GDBEST,GRDCOM,SMBEST,SMF                  DECIDE
13 C                                                                       DECIDE
14 C                                                                       DECIDE
15         JCMPLT = 0                                                      DECIDE
16         GDBEST = 6                                                      DECIDE
17 C                                                                       DECIDE
18 C    FOR EACH FACE OF THE LOG...                                        DECIDE
19 C                                                                       DECIDE
20         DO 5 J=1,4                                                      DECIDE
21 C                                                                       DECIDE
22 C    IS FACE COMPLETELY CUT?                                            DECIDE
23 C                                                                       DECIDE
24           IF (CMPLET(J).EQ.1) GO TO 2                                   DECIDE
25 C                                                                       DECIDE
26 C    IF NOT, IS THIS FACE BETTER THAN...                                DECIDE
27 C                                                                       DECIDE
28           IF (GRDCOM(J).LT.GDBEST) GO TO 3                              DECIDE
29 C                                                                       DECIDE
30 C    OR EQUAL TO THE 'BEST' FACE?                                       DECIDE
31 C                                                                       DECIDE
32           IF (GRDCOM(J).EQ.GDBEST) GO TO 4                              DECIDE
33           GO TO 5                                                       DECIDE
34 C                                                                       DECIDE
35 C    IF COMPLETELY CUT, INCREASE COUNT AND CHECK NEXT FACE              DECIDE
36 C                                                                       DECIDE
37      2    JCMPLT = JCMPLT+1                                             DECIDE
38           GO TO 5                                                       DECIDE
39 C                                                                       DECIDE
40 C    THIS FACE IS BETTER.  SAVE GRADE AND SURFACE MEASURE; CHECK        DECIDE
41 C    NEXT FACE                                                          DECIDE
42 C                                                                       DECIDE
43      3    F = J                                                         DECIDE
44           GDBEST = GRDCOM(J)                                            DECIDE
45           SMBEST = SMF(J)                                               DECIDE
46           GO TO 5                                                       DECIDE
47 C                                                                       DECIDE
48 C    USE SURFACE MEASURE TO BREAK TIES                                  DECIDE
49 C                                                                       DECIDE
50      4    IF (SMF(J).LT.SMBEST) GO TO 5                                 DECIDE
51           F = J                                                         DECIDE
52           SMBEST = SMF(J)                                               DECIDE
53      5 CONTINUE                                                         DECIDE
54 C                                                                       DECIDE
55 C    WHEN ALL FOUR FACES ARE COMPLETELY CUT, ROTATE LOG 15 DEGREES      DECIDE
56 C    AND REPROCESS                                                      DECIDE
57 C                                                                       DECIDE
58         IF (JCMPLT.EQ.4) RETURN1                                        DECIDE
59         RETURN                                                          DECIDE
60         END                                                             DECIDE
```

*** STATEMENT NUMBERS ***

2	24	*37				
3	28	*43				
4	32	*50				
5	20	33	38	46	50	*53

*** VARIABLES ***

CMPLET	11	12	24							
DECIDE	8									
F	8	*43	*51							
GDBEST	12	*16	28	32	*44					
GRDCOM	11	12	28	32	44					
J	*20	24	28	32	43	44	45	50	51	52
JCMPLT	*15	*37	58							
SMBEST	12	*45	50	*52						
SMF	11	12	45	50	52					

Function DEG

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

```
                    *** VARIABLES ***

A             5    7
DEG           5   *7
```

Function RAD

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

```
                    *** VARIABLES ***

A             5    7
RAD           5   *7
```

Live Sawing (LIVE)

```
C     'LIVE' SAWS THE LOG THROUGH ITS CENTRAL AXIS.  THE PLANE OF EACH    LIVE
C     SUBSEQUENT SAW CUT (AND HENCE EACH BOARD FACE) IS PARALLEL TO THIS  LIVE
C     FIRST CUT.  IN ADDITION, THE CORE DEFECT IS RIPPED OUT OF ANY BOARDS LIVE
C     IN WHICH IT APPEARS AND THE RESULTING BOARDS REGRADED               LIVE
C                                                                         LIVE
C                                                                         LIVE
      IMPLICIT REAL (K,L)                                                 LIVE
      REAL FLX(26,2),FLY(26,2),FUX(26,2),FUY(26,2),PERC(5),PERCRP(5),     LIVE
     .     ANGLE(40),HEIGHT(40),KTLEN(40),DAT*8                           LIVE
      INTEGER ID(22,2),NODEFC(2),CORFLG,OLD,GRDRGT,GRDLFT,GRDMID,F        LIVE
      COMMON /BOARD/ L,D,K,T,LOGS,RB,DISTBD,BDLEN,BOARDW,KNOTS,R          LIVE
     .       /DEFEC/ NODEFC,FLX,FLY,FUX,FUY,ID                            LIVE
     .       /FORRIP/ RPLOSS,RPKERF,LFTLOC,RGTLOC,GRDRGT,GRDLFT,GRDMID,   LIVE
     .                TLOSS                                               LIVE
     .       /PRICF/ FAS,SEL,ONEC,TWOC,THRB                               LIVE
C                                                                         LIVE
C                                                                         LIVE
C     THIS PROGRAM ASSUMES INPUT FROM UNIT 5 AND OUTPUTS TO UNITS 6 AND 8 LIVE
C                                                                         LIVE
C     GET LOG DIAMETER (IN.), LOG LENGTH (IN.), KERF SIZES (IN.), AND     LIVE
C     BOARD THICKNESS (IN.)                                               LIVE
C                                                                         LIVE
    1 READ (5,109,END=17) D                                               LIVE
      READ (5,109) L                                                      LIVE
      READ (5,110) K,RPKERF                                               LIVE
      READ (5,109) T                                                      LIVE
C                                                                         LIVE
C     GET KNOT HALF-ANGLE (DEGREES) AND LOG HALF-TAPER (DEGREES)          LIVE
C                                                                         LIVE
      READ (5,109) KNOTS                                                  LIVE
      READ (5,109) LOGS                                                   LIVE
C                                                                         LIVE
C     GET CORE DEFECT PARAMETERS: DIAMETER (IN.), LINEAR OFFSET (IN.),    LIVE
C     AND ANGULAR OFFSET (DEGREES)                                        LIVE
C                                                                         LIVE
      READ (5,105) CORDIA                                                 LIVE
      READ (5,105) COFSET                                                 LIVE
      READ (5,105) CANGS                                                  LIVE
C                                                                         LIVE
C     GET PRICE PER BOARD FOOT FOR EACH GRADE; NUMBER OF KNOTS            LIVE
C                                                                         LIVE
      READ (5,120) FAS,SEL,ONEC,TWOC,THRB                                 LIVE
      READ (5,101) MAX                                                    LIVE
C                                                                         LIVE
C     GET DATE AND CALCULATE RADIUS OF CORE DEFECT                        LIVE
C                                                                         LIVE
      CALL DATE (DAT)                                                     LIVE
      CRADUS=CORDIA/2.                                                    LIVE
C                                                                         LIVE
C     GET KNOT ANGLES, HEIGHTS, AND LENGTHS                               LIVE
C                                                                         LIVE
      DO 2 J=1,MAX                                                        LIVE
        READ (5,104) ANGLE(J),HEIGHT(J)                                   LIVE
    2 CONTINUE                                                            LIVE
      DO 3 J=1,MAX                                                        LIVE
        READ (5,105) KTLEN(J)                                             LIVE
    3 CONTINUE                                                            LIVE
C                                                                         LIVE
C     CALCULATE LOG RADIUS AT TOP AND BOTTOM; TOTAL LOG CUBIC FEET        LIVE
C                                                                         LIVE
      RI = D/2.                                                           LIVE
      RB = RI+L*TAN(RAD(LOGS))                                            LIVE
      LVOL = 1/3.*3.141593*(RB**2+RB*RI+RI**2)*L/144.                     LIVE
C                                                                         LIVE
C     OUTPUT HEADER                                                       LIVE
C                                                                         LIVE
      WRITE (6,121) DAT                                                   LIVE
      WRITE (6,122) L,D,LOGS,LVOL                                         LIVE
      WRITE (6,103) MAX                                                   LIVE
      DO 4 J=1,MAX                                                        LIVE
        WRITE (6,106) ANGLE(J),HEIGHT(J),KTLEN(J)                         LIVE
    4 CONTINUE                                                            LIVE
      WRITE (6,123) KNOTS                                                 LIVE
      WRITE (6,114) CRADUS,COFSET,CANGS                                   LIVE
      WRITE (6,124) K,RPKERF,T                                            LIVE
      WRITE (6,125) FAS,SEL,ONEC,TWOC,THRB                                LIVE
      WRITE (8,200) L,D,MAX,CRADUS,COFSET,CANGS,LVOL,K,T,KNOTS,LOGS,      LIVE
     .               RPKERF                                               LIVE
      WRITE (8,201) (KTLEN(J),J=1,MAX)                                    LIVE
      WRITE (8,201) (HEIGHT(J),J=1,MAX)                                   LIVE
      WRITE (8,201) (ANGLE(J),J=1,MAX)                                    LIVE
      WRITE (8,202) FAS,SEL,ONEC,TWOC,THRB                                LIVE
C                                                                         LIVE
C     INITIALIZE VARIABLES FOR CALCULATING HIGH, LOW, AND AVERAGE YIELDS  LIVE
C     FOR EACH LOG                                                        LIVE
C                                                                         LIVE
      HIGHS = 0.                                                          LIVE
      SMALLS = 999.                                                       LIVE
      AVGS = 0.                                                           LIVE
      HIRIPS = 0.                                                         LIVE
      SMRIPS = 999.                                                       LIVE
      AVRIPS = 0.                                                         LIVE
      AVRIP = 0.                                                          LIVE
      HIRIP = 0.                                                          LIVE
      LORIP = 999.                                                        LIVE
C                                                                         LIVE
C                                                                         LIVE
C     SAW LOG IN 12 DIFFERENT ROTATIONAL POSITIONS, 15 DEGREES APART, TO  LIVE
C     DETERMINE THE ORIENTATION GIVING THE HIGHEST YIELD                  LIVE
C                                                                         LIVE
      DO 16 IN=1,12                                                       LIVE
C                                                                         LIVE
C     INITIALIZE FOR EACH POSITION                                        LIVE
C                                                                         LIVE
        BF = 0.0                                                          LIVE
        NBD = 1                                                           LIVE
        TOTS = 0.                                                         LIVE
        RPTOTS = 0.                                                       LIVE
        TVAL = 0.0                                                        LIVE
        TVALRP = 0.0                                                      LIVE
        DO 5 J=1,5                                                        LIVE
          PERC(J) = 0.0                                                   LIVE
          PERCRP(J) = 0.0                                                 LIVE
    5   CONTINUE                                                          LIVE
        RPLOSS = 0.                                                       LIVE
        CORFLG = 0                                                        LIVE
C                                                                         LIVE
C                                                                         LIVE
C     FOR BOTH SIDES OF THE LOG...                                        LIVE
C                                                                         LIVE
        DO 13 F=1,2                                                       LIVE
C                                                                         LIVE
C     INITIALIZE PORTION OF LOG TO BE CUT... THIS SETS THE DEPTH OF THE   LIVE
C     FIRST CUT                                                           LIVE
C                                                                         LIVE
          R = RB+K/2.                                                     LIVE
C                                                                         LIVE
C     CUT BOARDS FROM THIS FACE UNTIL FACE IS COMPLETELY CUT.  THE LOG IS LIVE
C     CUT FROM THE INSIDE OUT                                             LIVE
C                                                                         LIVE
    6     CALL KERFL (&11,BF)                                             LIVE
C                                                                         LIVE
C     FOR BOTH SIDES OF THE BOARD...                                      LIVE
C                                                                         LIVE
          DO 10 I=1,2                                                     LIVE
C                                                                         LIVE
C     INITIALIZE DEFECT ARRAYS, EXCEPT F..(26,1), WHICH HOLDS THE BOARD   LIVE
C     DIMENSIONS                                                          LIVE
C                                                                         LIVE
            DO 7 J=1,25                                                   LIVE
              FLX(J,1) = -9999.0                                          LIVE
              FUX(J,1) = -9999.0                                          LIVE
              FLY(J,1) = -9999.0                                          LIVE
              FUY(J,1) = -9999.0                                          LIVE
    7       CONTINUE                                                      LIVE
            DO 8 I=1,22                                                   LIVE
              ID(J,1) = 0                                                 LIVE
    8       CONTINUE                                                      LIVE
            NODEFC(1) = 0                                                 LIVE
C                                                                         LIVE
```

```
C                                                                       LIVE
C   PUT IN CORE DEFECT                                                  LIVE
C                                                                       LIVE
          CALL COREL (CRADUS,COFSET,CANG$,CORFLG,I)                     LIVE
C                                                                       LIVE
C   PUT IN KNOTS                                                        LIVE
C                                                                       LIVE
          DO 9 J=1,MAX                                                  LIVE
            CALL KNOTL (HEIGHT(J),KTLEN(J),ANGLE(J))                    LIVE
    9     CONTINUE                                                      LIVE
C                                                                       LIVE
C   PUT IN WANE DEFECTS                                                 LIVE
C                                                                       LIVE
          CALL WANEL                                                    LIVE
C                                                                       LIVE
C   AND GRADE THE FACE                                                  LIVE
C                                                                       LIVE
          CALL GRADE (NPG)                                              LIVE
C                                                                       LIVE
C   SAVE GRADE OF FIRST SIDE OF BOARD                                   LIVE
C                                                                       LIVE
          IF (I.EQ.1) OLD = NPG                                         LIVE
C                                                                       LIVE
C   DETERMINE BOARD GRADE AND VALUE BASED ON GRADES OF BOTH FACES       LIVE
C                                                                       LIVE
          IF (I.EQ.2) CALL PRICEL (BF,OLD,NPG,TVAL,PERC,V)              LIVE
C                                                                       LIVE
C   RIP CORE DEFECT OUT OF BOARD                                        LIVE
C                                                                       LIVE
          CALL RERIP (TVALRP,PERCRP,NPG,V,CORFLG,I,BF)                  LIVE
C                                                                       LIVE
C   SUBTRACT BOARD THICKNESS AND PROCESS THE INNER FACE                 LIVE
C                                                                       LIVE
          DISTBD = DISTBD-T                                             LIVE
   10   CONTINUE                                                        LIVE
C                                                                       LIVE
C   INCREASE BOARD COUNT AND CUT ANOTHER BOARD                          LIVE
C                                                                       LIVE
        NBD = NBD+1                                                     LIVE
        GO TO 6                                                         LIVE
C                                                                       LIVE
C   TURN LOG 180 DEGREES TO CUT OTHER SIDE                              LIVE
C                                                                       LIVE
   11   DO 12 J=1,MAX                                                   LIVE
          ANGLE(J) = AMOD(ANGLE(J)+180.0,360.0)                         LIVE
   12   CONTINUE                                                        LIVE
        CANG$ = AMOD(CANG$+180.0,360.0)                                 LIVE
   13 CONTINUE                                                          LIVE
C                                                                       LIVE
C   PRINT RESULTS FOR THIS POSITION                                     LIVE
C                                                                       LIVE
      IROTAT = INX*15-15                                                LIVE
      WRITE (6,100) IROTAT                                              LIVE
      WRITE (6,112) (PERC(J),J=1,5)                                     LIVE
      TOTS = PERC(1)+PERC(2)+PERC(3)+PERC(4)+PERC(5)                    LIVE
      WRITE (6,111) TOTS                                                LIVE
      WRITE (8,203) IROTAT                                              LIVE
      WRITE (8,204) (PERC(J),J=1,5)                                     LIVE
      WRITE (8,204) TOTS                                                LIVE
      DO 14 J=1,5                                                       LIVE
        PERC(J) = (PERC(J)/LVOL)*100.                                   LIVE
        PERCRP(J) = (PERCRP(J)/LVOL)*100.                               LIVE
        RPTOTS = RPTOTS+PERCRP(J)                                       LIVE
   14 CONTINUE                                                          LIVE
      TOTS = TOTS*100./LVOL                                             LIVE
      WRITE (6,107) (ANGLE(J),J=1,MAX)                                  LIVE
      WRITE (6,108) (PERC(J),J=1,5),TVAL,(PERCRP(J),J=1,5),TVALRP       LIVE
      WRITE (6,113) TOTS                                                LIVE
      WRITE (6,118) RPLOSS                                              LIVE
      WRITE (8,205) (PERC(J),J=1,5),(PERCRP(J),J=1,5)                   LIVE
      WRITE (8,204) TVAL,TOTS,TVALRP,RPLOSS,RPTOTS                      LIVE
C                                                                       LIVE
C   SAVE HIGH, LOW, AND AVERAGE YIELDS FOR THIS LOG                     LIVE
C                                                                       LIVE
      AVG$ = AVG$+TVAL                                                  LIVE
      AVRIP$ = AVRIP$+TVALRP                                            LIVE
      IF (TVAL.GT.HIGH$) HIGH$ = TVAL                                   LIVE
      IF (TVAL.LT.SMALL$) SMALL$ = TVAL                                 LIVE
      IF (TVALRP.GT.HIRIP$) HIRIP = RPTOTS                              LIVF
      IF (TVALRP.LT.SMRIP$) LORIP = RPTOTS                              LIVE
      IF (TVALRP.GT.HIRIP$) HIRIP$ = TVALRP                             LIVE
      IF (TVALRP.LT.SMRIP$) SMRIP$ = TVALRP                             LIVE
      AVRIP = AVRIP+RPTOTS                                              LIVE
C                                                                       LIVE
C   ROTATE LOG BY 15 DEGREES AND REPROCESS                              LIVE
C                                                                       LIVF
      DO 15 J=1,MAX                                                     LIVE
        ANGLE(J) = AMOD(ANGLE(J)+15.,360.0)                             LIVE
   15 CONTINUE                                                          LIVE
      CANG$ = AMOD(CANG$+15.0,360.0)                                    LIVE
   16 CONTINUE                                                          LIVE
C                                                                       LIVE
C   PRINT HIGH, LOW, AND AVERAGE YIELDS FOR THIS LOG                    LIVE
C                                                                       LIVE
      AVRIP = AVRIP/12.                                                 LIVE
      AVG$ = AVG$/12.                                                   LIVE
      AVRIP$ = AVRIP$/12.                                               LIVE
      WRITE (6,115) HIGH$,AVG$,SMALL$                                   LIVE
      WRITE (6,116) HIRIP$,AVRIP$,SMRIP$                                LIVE
      WRITE (6,119) HIRIP,AVRIP,LORIP                                   LIVE
C                                                                       LIVE
C   LOG COMPLETELY PROCESSED, READ PARAMETERS FOR NEXT LOG              LIVE
C                                                                       LIVE
      GO TO 1                                                           LIVE
C                                                                       LIVE
   17 STOP                                                              LIVE
C                                                                       LIVE
C   FORMATS FOR DEVICE 6 (PRINTER)                                      LIVE
C                                                                       LIVE
  100 FORMAT (2(/),' ROTATION IS',I4)                                   LIVE
  101 FORMAT (I2)                                                       LIVE
  103 FORMAT (' THERE ARE',I3,' KNOTS IN THIS LOG',/,4X,'ANGLE  ',      LIVE
     . 'HEIGHT  LENGTH')                                                LIVE
  104 FORMAT (2F6.2)                                                    LIVE
  105 FORMAT (F6.3)                                                     LIVE
  106 FORMAT (3(2X,F6.2))                                               LIVE
  107 FORMAT (' KNOT ANGLES: ',15F7.2,2(/,13X,15F7.2))                  LIVE
  108 FORMAT (' % AVAILABLE SURFACE MEASURE IN EACH GRADE WITHOUT ',    LIVE
     . 'RERIP: ',5F5.1,' LOG VALUE IS $',F6.2,/,46X,'WITH RERIP: ',     LIVE
     . 5F5.1,' LOG VALUE IS $',F6.2)                                    LIVE
  109 FORMAT (F5.1)                                                     LIVE
  110 FORMAT (2F5.3)                                                    LIVE
  111 FORMAT (' TOTAL SURFACE MEASURE IS',F8.2)                         LIVE
  112 FORMAT (' SURFACE MEASURE PER GRADE IS',5F8.2)                    LIVE
  113 FORMAT (' TOTAL PERCENT YIELD IS',F7.2)                           LIVE
  114 FORMAT (' CORE DEFECT RADIUS ',F6.3,'  LINEAR OFFSET ',F6.3,      LIVE
     . '  ANGULAR OFFSET ',F6.3)                                        LIVE
  115 FORMAT (' $ YIELD:  HIGH ',F6.2,'  AVERAGE ',F6.2,'  LOW ',F6.2)  LIVE
  116 FORMAT (' RERIP $ YIELD HIGH ',F6.2,' AVERAGE ',F6.2,' LOW ',     LIVE
     . F6.2)                                                            LIVE
  118 FORMAT (' RERIP SURFACE MEASURE LOSS IS',F6.2,' BOARD FEET')      LIVE
  119 FORMAT (' RERIP % YIELD:  HIGH ',F6.2,'   AVERAGE ',F6.2,         LIVE
     . '   LOW ',F6.2)                                                  LIVE
  120 FORMAT (5F10.4)                                                   LIVE
  121 FORMAT (' LIVE SAWING WITH RERIP ',A8)                            LIVE
  122 FORMAT (' LOG PARAMETERS:  LENGTH',F6.1,'  DIA',F6.1,'  TAPER',   LIVE
     . F6.4,' DEGREES,  LOG VOLUME:',F8.2,' CUBIC FEET')                LIVE
  123 FORMAT (' ANGLES MEASURED FROM ZERO DEGREES = EAST FOR A ',       LIVE
     . 'VERTICALLY CUTTING SAW.  KNOT TAPER',F6.2)                      LIVE
  124 FORMAT (' CUTTING PARAMETERS:  KERF',F7.4,'  RERIP KERF:',F7.4,   LIVE
     . '  BOARD THICKNESS:',F5.2)                                       LIVE
  125 FORMAT (' PRICES PER BOARD FOOT  ',5F10.4)                        LIVE
C                                                                       LIVE
C   FORMATS FOR DEVICE 8 (STORAGE MEDIUM)                               LIVE
C                                                                       LIVE
  200 FORMAT (F6.1,F5.1,I3,3F5.1,F8.2,F6.3,F4.1,F6.2,F5.3,F6.3)         LIVE
  201 FORMAT (3(10F7.2/))                                               LIVE
  202 FORMAT (5F7.4)                                                    LIVE
  203 FORMAT (I3)                                                       LIVE
  204 FORMAT (5F8.2)                                                    LIVE
  205 FORMAT (10F6.2)                                                   LIVE
      END                                                               LIVE
```

```
               *** STATEMENT NUMBERS ***

   1      *23   253
   2       52   *54
   3       55   *57
   4       70   *72
   5      111  *114
   6     *131   189
   7      140  *145
   8      146  *148
   9      157  *159
  10      135  *184
  11      131  *193
  12      193  *195
  13      121  *197
  14      209  *213
  15      236  *238
  16      181  *240
  17       23  *255
 100      202  *259
 101       43  *260
 103       69  *261
 104       53  *263
 105       36    37    38    56  *264
 106       71  *265
 107      215  *266
 108      216  *267
 109       23    24    26    30    31  *270
 110       25  *271
 111      205  *272
 112      203  *273
 113      217  *274
 114       74  *275
 115      247  *277
 116      248  *278
 118      218  *280
 119      243  *281
 120       42  *283
 121       67  *284
 122       68  *285
 123       73  *287
 124       75  *289
 125       76  *291
 200       77  *295
 201       79    80    81  *296
 202       82  *297
 203      206  *298
 204      207   208   220  *299
 205      213  *300

                    *** VARIABLES ***

AMOD      194   196   237   239
ANGLE       8   *53    71    81   158  *194   215  *237
AVG$      *89  *224  *245   247
AVRIP     *93  *232  *244   249
AVRIP$    *92  *225  *246   248
BDLEN      11
BF       *105   131   175   179
BOARDW     11
CANG$     *38    74    77   153  *196  *239
COFSET    *37    74    77   153
CORDIA    *36    48
COREL     153
CORFLG     10  *116   153   179
CRADUS    *48    74    77   153
D          11   *23    61    68    77
DAT         8    47    67
DATE       47
DISTBD     11  *183
F          11  *121
FAS        11   *42    76    82
FLX         8    11  *141
FLY         8    11  *143
FUX         8    11  *142
FUY         8    11  *144
GRADE     167
GRDLFT     10    11
GRDMID     10    11
GRDRGT     10    11
HEIGHT      9   *53    71    80   158
HIGH$     *87  *226   247
HIPIP     *94  *228   249
HIPIP$    *90   228  *230   248
I        *135   153   171   175   179
ID         10    11  *147
INX      *101   201
IROTAT   *201   202   206
J         *52   *53   *55   *56   *70    71    79    80    81  *111   112   113
         *140   141   142   143   144  *146   147  *157   158  *193   194   203
          207  *209   210   211   212   215   216   219  *236   237
K          11   *25    75    77   126
KEPFL     131
KNOT$      11   *30    73    77
KNOTL     158
KTLEN       8   *56    71    79   158
L          11   *24    62    63    68    77
LFTLOC     11
LOG$       11   *31    62    68    77
LORIP     *95  *229   249
LVOL      *63    68    77   210   211   214
MAX       *43    52    55    69    70    77    79    80    81   157   193   215
          236
NBD      *106  *188
NODEFC     10    11  *149
NPG       167   171   175   179
OLP        10  *171   175
ONEC       11   *42    76    82
PEPC        8  *112   175   203   204   207  *210   216   219
PEPCPP      8  *113   179  *211   212   216   219
PPICEL    175
P          11  *126
PAD        62
PB         11   *62    63   126
PEPIP     179
PGTLOC     11
PI        *61    62    63
PPKEFF     11   *25    75    77
RPLOSS     11  *115   218   220
RPTOTS   *108  *212   220   220   229   232
SEL        11   *42    76    82
SMALL$    *88  *227   247
SMPIP$    *91   229  *231   248
T          11   *26    75    77   183
TAN        62
THRB       11   *42    76    82
TLOSS      11
TOTS     *107  *204   205   208  *214   217   220
TVAL     *109   175   216   220   224   226   227
TVALRP   *110   179   216   220   225   228   229   230   231
TWOC       11   *42    76    82
V         175   179
WANEL     163
```

Subroutine KERF

```
C     SUBROUTINE 'KERF' CUTS A BOARD FROM THE LOG.  THE BOARD WIDTH IS
C     DEFINED AT THE CENTER. AND THE BOARD LENGTH FROM THE BOTTOM
C     OF THE LOG
C
C
      SUBROUTINE KERFL (*.BF)
C
      IMPLICIT REAL (K.L)
      REAL FLX(26.2).FLY(26.2).FUX(26.2).FUY(26.2)
      INTEGFR ID(22.2).NODEFC(2)
      COMMON /BOARD / L.D.K.T.LOG$.RB.DISTBD.BDLEN.BOARDW.KNOTS.R
     /      /DEFEC/ NODEFC.FLX.FLY.FUX.FUY.ID
C
C
C     ALL BOARDS MUST BE AT LEAST 48 INCHES LONG. 2.5 INCHES WIDE AT
C     THE TOP. AND 3 INCHES WIDE AT THE CENTER.  IN ADDITION. THE TOTAL
C     AMOUNT OF WANE FROM THE CENTER TO THE TOP IS LIMITED TO 4 INCHES.
C     THE BOARD IS CUT BACK IN ONE FOOT INCREMENTS IF THE WIDTH OR WANE
C     TESTS FAIL.  IF THE LENGTH TEST FAILS. ANOTHER BOARD IS CUT FROM
C     THIS FACE
C
C
C      FIND OUTER FACE OF BOARD
C
      RI = D/2.
    1 R = R-T-K
      BDLEN = L
C
C     IF THIS FACE OF THE LOG IS COMPLETELY CUT. EXIT
C
      IF (R.LT.0.) RETURN1
      DISTBD = RB-R
      POSLEN = R*TAN(RAD(90.-LOG$))
C
C
C     CHECK FOR BOARDS SHORTER THAN THE LOG DUE TO LOG TAPER
C
      IF (POSLEN.LT.BDLEN) BDLEN = POSLEN
C
C     BOARDS MUST BE AT LEAST 4 FEET LONG
C
    2 IF (BDLEN.LT.48.) GO TO 1
C
C     DETERMINE HALF-WIDTH AT MID-LENGTH OF BOARD
C
      RM = D/2.+(L-BDLEN/2.)*TAN(RAD(LOG$))
      IF (DISTBD.GT.RM) GO TO 3
      BOARDW = SQRT(RM**2-DISTBD**2)
C
C     BOARD MUST BE AT LEAST 3 INCHES WIDE AT MID-LENGTH
C
      IF (BOARDW.LT.1.5) GO TO 3
C
C     DETERMINE HALF-WIDTH AT TOP OF BOARD
C
      RT = RI+(L-BDLEN)*TAN(RAD(LOG$))
      IF (DISTBD.GT.RT) GO TO 3
      WT = SQRT(RT**2-DISTBD**2)
C
C     BOARD MUST BE AT LEAST 2.5 INCHES WIDE AT THE TOP
C
      IF (WT.LT.1.25) GO TO 3
C
C     LIMIT THE TOTAL WANE TO AVOID GENERATING MORE DEFECTS THAN THE
C     GRADING PROGRAM CAN HANDLE.  TOTAL WIDTH OF WANE MUST BE LESS
C     THAN 4 INCHES
C
      WAN = BOARDW-WT
      IF (WAN.GT.2.0) GO TO 3
C
C     CONVERT BOARD WIDTH AND LENGTH TO EVEN QUARTER INCHES
C
      BOARDW = INT(BOARDW*8.)/8.
      BDLEN = INT(BDLEN*4.)/4.
C
C     DEFINE BOARD EDGES IN QUARTER INCH UNITS
C
      FLY(26.1) = BOARDW*8.
      FLX(26.1) = BDLEN*4.
      FUX(26.1) = 0.0
      FUY(26.1) = 0.0
C
C     CALCULATE UNROUNDED BOARD FEET
C
      BF = BDLEN*BOARDW*2.*T/144.
      RETURN
C
C     CUT BACK BOARD BY ONE FOOT
C
    3 BDLEN = BDLEN-12.
      GO TO 2
      END
```

```
                 ***  STATEMENT NUMBERS  ***

  1      *26    42
  2      *42    91
  3       47    52    57    62    69   *90

                 ***  VARIABLES  ***

BDLEN     11   *27   *38    42    46    56   *74    79    85   *90
BF         6   *85
BOARDW    11   *48    52    68   *73    78    85
D         11    25    46
DISTBD    11   *32    47    48    57    58
FLX        9    11   *79
FLY        9    11   *78
FUX        9    11   *80
FUY        9    11   *81
ID        10    11
INT       73    74
K         11    26
KERFL      6
KNOTS     11
L         11    27    46    56
LOG$      11    33    46    56
NODEFC    10    11
POSLEN   *33    38
R         11   *26    31    32    33
RAD       33    46    56
RB        11    32
RI       *25    56
RM       *46    47    48
RT       *56    57    58
SQRT      48    58
T         11    26    85
TAN       33    46    56
WAN      *68    69
WT       *58    62    68
```

Subroutine WANE

```
 1 C   SUBROUTINE 'WANE' LOCATES WANE DEFECTS ON THE BOARD                   WANEL
 2 C                                                                         WANEL
 3 C                                                                         WANEL
 4       SUBROUTINE WANEL                                                    WANEL
 5 C                                                                         WANEL
 6 C   THE BOARD EDGE HAS A NEW WANE DEFECT IF THE WIDTH DROPS BY 1/2        WANEL
 7 C   INCH (1/4 INCH FOR EACH SIDE) FROM THAT AT THE MIDDLE.  THE BOARD     WANEL
 8 C   IS SYMMETRICAL SO THE WANE DEFECTS OCCUR AT THE SAME PLACE ON         WANEL
 9 C   BOTH EDGES OF THE BOARD                                               WANEL
10 C                                                                         WANEL
11       IMPLICIT REAL (K,L)                                                 WANEL
12       INTEGER ID(22,2),NODEFC(2)                                          WANEL
13       REAL FLX(26,2),FLY(26,2),FUX(26,2),FUY(26,2)                        WANEL
14       COMMON /BOARD/ L,D,K,T,LOGS,RB,DISTBD,BDLEN,BOARDW,KNOTS,R          WANEL
15            /DEFEC/ NODEFC,FLX,FLY,FUX,FUY,ID                              WANEL
16 C                                                                         WANEL
17 C                                                                         WANEL
18       CON1 = TAN(RAD(LOGS))                                               WANEL
19       CON2 = DISTBD**2                                                    WANEL
20       IF ((RB-SQRT(CON2+BOARDW**2))/CON1.GE.BDLEN) RETURN                 WANEL
21       OLDH = BDLEN*2.                                                     WANEL
22       V = 0.                                                              WANEL
23 C                                                                         WANEL
24 C   FIND THE PLACE ALONG THE BOARD WHERE THE HALF-WIDTH HAS DECREASED     WANEL
25 C   BY 1/4 INCH.  THE DEFECT ENDS HERE AND EXTENDS FROM THE END OF THE    WANEL
26 C   LAST WANE DEFECT (OR FROM THE MIDDLE OF THE BOARD).                   WANEL
27 C                                                                         WANEL
28   1 V = V+0.25                                                            WANEL
29       SQR = SQRT(CON2+(BOARDW-V)**2)                                      WANEL
30       H = INT(AMIN1((RB-SQR)*4./CON1,BDLEN*4.))                           WANEL
31       NODEFC(1) = NODEFC(1)+1                                             WANEL
32       M = NODEFC(1)                                                       WANEL
33 C                                                                         WANEL
34 C   LIMIT NUMBER OF DEFECTS TO MAXIMUM ALLOWED BY GRADING PROGRAM (22)    WANEL
35 C                                                                         WANEL
36       IF (M.GT.22) GO TO 2                                                WANEL
37 C                                                                         WANEL
38 C   SAVE WANE COORDINATES FOR GRADING PROGRAM                             WANEL
39 C                                                                         WANEL
40       FUY(M,1) = V*4.+1                                                   WANEL
41       FLY(M,1) = 0.                                                       WANEL
42       FUX(M,1) = H                                                        WANEL
43       FLX(M,1) = OLDH                                                     WANEL
44       ID(M,1) = 5                                                         WANEL
45 C                                                                         WANEL
46 C   SINCE THE BOARD IS SYMMETRICAL, USE THE PREVIOUSLY GENERATED HEIGHT   WANEL
47 C   FOR THE WANE DEFECTS ON THE OTHER EDGE OF THE BOARD                   WANEL
48 C                                                                         WANEL
49   2 NODEFC(1) = NODEFC(1)+1                                               WANEL
50       M = NODEFC(1)                                                       WANEL
51       IF (M.GT.22) GO TO 3                                                WANEL
52       FUY(M,1) = BOARDW*8.                                                WANEL
53       FLY(M,1) = BOARDW*8.-V*4.-1.                                        WANEL
54       FUX(M,1) = H                                                        WANEL
55       FLX(M,1) = OLDH                                                     WANEL
56       ID(M,1) = 5                                                         WANEL
57 C                                                                         WANEL
58 C   WHEN THE DEFECT EXTENDS TO THE END OF THE BOARD, EXIT                 WANEL
59 C                                                                         WANEL
60   3 IF (H.GE.BDLEN*4.) RETURN                                             WANEL
61 C                                                                         WANEL
62 C   THE NEXT WANE DEFECT BEGINS WHERE THIS ONE ENDS                       WANEL
63 C                                                                         WANEL
64       OLDH = H                                                            WANEL
65       GO TO 1                                                             WANEL
66       RETURN                                                              WANEL
67       END                                                                 WANEL
```

*** STATEMENT NUMBERS ***

```
      1     *28   65
      2      36  *49
      3      51  *60
```

*** VARIABLES ***

```
AMIN1    30
BDLEN    14   20   21   30   60
BOARDW   14   20   29   52   53
CON1    *18   20   30
CON2    *19   20   29
D        14
DISTBD   14   19
FLX      13   14  *43  *55
FLY      13   14  *41  *53
FUX      13   14  *42  *54
FUY      13   14  *40  *52
H       *30   42   54   60   64
ID       12   14  *44  *56
INT      30
K        14
KNOTS    14
L        14
LOGS     14   18
M       *32   36   40   41   42   43   44  *50   51   52   53   54
         55   56
NODEFC   12   14  *31   32  *49   50
OLDH    *21   43   55  *64
R        14
RAD      18
RB       14   20   30
SQR     *29   30
SQRT     20   29
T        14
TAN      18
V       *22  *28   29   40   53
WANEL     4
```

Subroutine KNOT

```
C     SUBROUTINE 'KNOT' LOCATES THE KNOT DEFECTS ON THE BOARD FACE.
C     ANGLES ARE MEASURED CLOCKWISE WITH 0 DEGREES DEFINED AS THE
C     LINE FROM THE CENTER OF THE LOG PERPENDICULAR TO THE BOARD FACE
C
C
      SUBROUTINE KNOTL (KHIGH,KLEN,KTANG$)
C
      IMPLICIT REAL (K,L)
      INTEGER ID(22,2),NODEFC(2)
      REAL FLX(26,2),FLY(26,2),FUX(26,2),FUY(26,2)
      COMMON /DEFEC/ NODEFC,FLX,FLY,FUX,FUY,ID
             /BOARD/ L,D,K,T,LOG$,RB,DISTBD,BDLEN,BOARDW,KNOT$,R
C
C
C     SEE IF THE KNOT IS LONG ENOUGH TO REACH THE BOARD FACE
C
      IF (KLEN.LT.DISTBD) RETURN
C
C     CALCULATE KNOT DEFECT HALF-LENGTH AS PROJECTED ON THE FACE, LOG
C     RADIUS AT KNOT HEIGHT, AND BOARD HALF-WIDTH ANGLE
C
      DY = DISTBD*TAN(RAD(KNOT$))
      LRADUS = RB-KHIGH*TAN(RAD(LOG$))
      BDWID$ = DEG(ATAN(BOARDW/DISTBD))
C
C     THE KNOT EFFECTIVELY ENDS AT THE OUTER EDGE OF THE LOG
C
      IF (KLEN.GT.LRADUS) KLEN = LRADUS
C
C     SINCE THE LOG IS TAPERED, THE OUTER BOARDS MAY NOT BE THE FULL LOG
C     LENGTH, SO KNOTS ABOVE THE END OF THE BOARD ARE NOT CONSIDERED
C
      IF (LRADUS.LT.DISTBD) RETURN
C
C     ANGLE OF ROTATION PUTS KNOT COMPLETELY OUTSIDE BOARD
C
      IF ((KTANG$.GT.180.).AND.((KTANG$+KNOT$).LT.(360.-BDWID$))) RETURN
      IF ((KTANG$.LE.180.).AND.((KTANG$-KNOT$).GT.BDWID$)) RETURN
C
C
C     FIND THE QUADRANT IN WHICH THE KNOT LIES (ASSUME QUADRANT ONE)
C
C     QUADRANT TWO
C
      IF ((KTANG$.LT.180.).AND.((KTANG$+KNOT$).GT.BDWID$)) GO TO 3
C
C     QUADRANT THREE
C
      IF ((KTANG$.GE.180.).AND.((KTANG$-KNOT$).LT.(360.-BDWID$)))
       GO TO 4
C
C     QUADRANT FOUR
C
      IF (KTANG$.GT.270.) GO TO 5
C
C
C     QUADRANT ONE
C
C     SEE IF KNOT CENTER IS WITHIN ONE KNOT HALF-ANGLE OF 0 DEGREES
C
      IF (KTANG$-KNOT$.LT.0.) GO TO 2
C
C     CALCULATE DISTANCE TO FACE ALONG BOTH SIDES OF KNOT
C
      ANGBD1 = DISTBD/COS(RAD(KTANG$-KNOT$))
      ANGBD2 = DISTBD/COS(RAD(KTANG$+KNOT$))
C
C     KNOT IS NOT LONG ENOUGH TO REACH FACE
C
      IF (KLEN.LT.ANGBD1) RETURN
C
C     FIND INTERSECTION OF NEAR SIDE OF KNOT AND FACE
C
      XR = BOARDW-DISTBD*TAN(RAD(KTANG$-KNOT$))
      IF (KLEN.LT.ANGBD2) GO TO 1
C     FAR SIDE OF KNOT REACHES FACE
C
      XL = BOARDW-DISTBD*TAN(RAD(KTANG$+KNOT$))
      FULWID = (XR-XL)/2.
      GO TO 8
C
C     FAR SIDE OF KNOT DOES NOT REACH FACE
C
    1 XL = BOARDW-SQRT(KLEN**2-DISTBD**2)
      FULWID =
     . DISTBD*TAN(RAD(KTANG$+KNOT$))-DISTBD*TAN(RAD(KTANG$-KNOT$))
      GO TO 8
C
C     KNOT IS WITHIN ONE KNOT HALF-ANGLE OF 0 DEGREES
C
    2 XL = AMIN1(SQRT(KLEN**2-DISTBD**2),DISTBD*TAN(RAD(KNOT$+KTANG$)))
      XL = BOARDW-XL
      XR = AMIN1(SQRT(KLEN**2-DISTBD**2),DISTBD*TAN(RAD(KNOT$-KTANG$)))
      XR = BOARDW+XR
C
C     KNOTS NEARLY PERPENDICULAR HAVE THEIR WIDTH ESTIMATED BY THEIR
C     LENGTH
C
      FULWID = DY*2.
      GO TO 8
C
C     QUADRANT TWO
C
    3 ANGBD1 = DISTBD/COS(RAD(KTANG$-KNOT$))
C
C     TOO SHORT TO REACH FACE
C
      IF (KLEN.LT.ANGBD1) RETURN
      XR = BOARDW-SQRT(ANGBD1**2-DISTBD**2)
      XL = BOARDW-SQRT(KLEN**2-DISTBD**2)
      IF ((KTANG$+KNOT$).GE.89.5)
     . FULWID = LRADUS-DISTBD*TAN(RAD(KTANG$-KNOT$))
      IF ((KTANG$+KNOT$).LT.89.5)
     . FULWID = AMIN1(LRADUS,DISTBD*TAN(RAD(KTANG$+KNOT$)))
     .     -DISTBD*TAN(RAD(KTANG$-KNOT$))
      GO TO 8
C
C     QUADRANT THREE
C
    4 ANGBD1 = DISTBD/COS(RAD(360.-KTANG$-KNOT$))
      IF (KLEN.LT.ANGBD1) RETURN
      XR = BOARDW+SQRT(KLEN**2-DISTBD**2)
      XL = BOARDW+SQRT(ANGBD1**2-DISTBD**2)
      IF ((KTANG$-KNOT$).LE.270.5)
     . FULWID = LRADUS-DISTBD*TAN(RAD(360.-KTANG$-KNOT$))
      IF ((KTANG$-KNOT$).GT.270.5)
     . FULWID = AMIN1(LRADUS,DISTBD*TAN(RAD(360.-KTANG$+KNOT$)))
     .     -DISTBD*TAN(RAD(360.-KTANG$-KNOT$))
      GO TO 8
C
C     QUADRANT FOUR
C
    5 IF (360.-KTANG$-KNOT$.LT.0.) GO TO 7
      ANGBD1 = DISTBD/COS(RAD(360.-KTANG$-KNOT$))
      ANGBD2 = DISTBD/COS(RAD(360.-KTANG$+KNOT$))
      IF (KLEN.LT.ANGBD1) RETURN
      XL = BOARDW+DISTBD*TAN(RAD(360.-KTANG$-KNOT$))
      IF (KLEN.LT.ANGBD2) GO TO 6
C
C     FAR SIDE OF KNOT REACHES FACE
C
      XR = BOARDW+DISTBD*TAN(RAD(360.-KTANG$+KNOT$))
      FULWID = (XR-XL)/2.
      GO TO 8
C
C     FAR SIDE OF KNOT DOES NOT REACH FACE
C
    6 XR = BOARDW+SQRT(KLEN**2-DISTBD**2)
      FULWID = DISTBD*TAN(RAD(360.-KTANG$+KNOT$))
     .  -DISTBD*TAN(RAD(360.-KTANG$-KNOT$))
      GO TO 8
C
C
```

```
154 C     THE KNOT IS WITHIN ONE KNOT HALF-ANGLE OF 0 DEGREES                    KNOTL
155 C                                                                            KNOTL
156       7 XL = AMIN1(SQRT(KLEN**2-DISTBD**2),                                  KNOTL
157      .  DISTBD*TAN(RAD(KTANG$+KNOT$-360.)))                                  KNOTL
158         XL = BOARDW-XL                                                       KNOTL
159         XR = AMIN1(SQRT(KLEN**2-DISTBD**2),                                  KNOTL
160      .  DISTBD*TAN(RAD(360.-KTANG$+KNOT$)))                                  KNOTL
161         XR = BOARDW+XR                                                       KNOTL
162         FULWID = D/2.                                                        KNOTL
163 C                                                                            KNOTL
164 C     IF A KNOT COMPLETELY INTERSECTS THE FACE, ITS LENGTH IS EXACTLY AS     KNOTL
165 C     MEASURED ON THE FACE.  OTHERWISE, ITS LENGTH IS ESTIMATED AS A         KNOTL
166 C     PROPORTION OF ITS PROJECTED LENGTH                                     KNOTL
167 C                                                                            KNOTL
168         DY = AMIN1(DY,DY*(XR-XL)*2./FULWID)                                  KNOTL
169         YB = KHIGH-DY                                                        KNOTL
170         YT = KHIGH+DY                                                        KNOTL
171 C                                                                            KNOTL
172 C     DEFECT LOCATIONS FOUND.  CHECK THAT THEY DO NOT EXTEND BEYOND THE      KNOTL
173 C     BOARD EDGES AND CONVERT TO QUARTER INCH UNITS FOR THE GRADING          KNOTL
174 C     PROGRAM                                                                KNOTL
175 C                                                                            KNOTL
176         XL = AMAX0(0,INT(XL*4.))                                             KNOTL
177         XR = AMIN0(INT(BOARDW*8.),INT(XR*4.+1.))                             KNOTL
178         YB = AMAX0(0,INT(YB*4.))                                             KNOTL
179         YT = AMIN0(INT(BDLEN*4.),INT(YT*4.+1.))                              KNOTL
180 C                                                                            KNOTL
181 C     LIMIT NUMBER OF DEFECTS TO MAXIMUM ALLOWED BY GRADING PROGRAM (22)     KNOTL
182 C                                                                            KNOTL
183         NODEFC(1) = NODEFC(1)+1                                              KNOTL
184         IF (NODEFC(1).GT.22) RETURN                                          KNOTL
185 C                                                                            KNOTL
186 C     SAVE KNOT COORDINATES FOR GRADING PROGRAM                              KNOTL
187 C                                                                            KNOTL
188         ID(NODEFC(1),1) = 3                                                  KNOTL
189         FL(NODEFC(1),1) = XL                                                 KNOTL
190         FL(NODEFC(1),1) = YB                                                 KNOTL
191         FU(NODEFC(1),1) = XR                                                 KNOTL
192         FU(NODEFC(1),1) = YT                                                 KNOTL
193         RETURN                                                               KNOTL
194         END                                                                  KNOTL
```

```
COS       65    66   104   120   134   135
D         11
DEG       24
DISTBD    11    17    22    24    33    65    66    74    78    84    85    91
          97   104   109   110   111   113   120   122   123   126   134   135
         137   142   148   149   156   159
DY       *22    99   162  *168   169   170
FLX       10    11  *190
FLY       10    11  *189
FULWID   *79   *85   *99  *111  *113  *126  *143  *149  *162   168
FUX       10    11  *192
FUY       10    11  *191
ID         9    11  *188
INT      176   177   178   179
K         11
KHIGH      9    23   169   170
KLEN       6    17   *26    70    75    84    91    93   108   110   121   122
         136   138   148   156   159
KNOT$     11    22    37    38    45    49    61    65    66    74    78    85
          91    93   104   111   113   120   126   133   134   135   137   142
         149   156   159
KNOTL      1
KTANG$     6    37    38    45    49    54    61    65    66    74    78    85
          91    93   104   111   113   120   126   133   134   135   137   142
         149   156   159
L         11
LOG$      11    23
LRADUS   *23    28    37   111   113   126
NODEFC     9    11  *183   184   188   189   190   191   192
P         11
RAD       22    23    65    66    74    78    85    91    93   104   111   113
         120   126   134   135   137   142   149   156   157
RB        11    23
SQRT      51    51    92   109   110   122   123   148   156   159
T         11
TAN       22    23    74    78    85    91    93   111   113   126   137   [illegible]
         149   156   159
XL       *75    79   *84   *91   *92  *110  *123  *137   143  *156  *158   168
        *176   189
XR       *21    79   *97   *94  *109  *122  *142   143  *149  *159  *161   163
        *177   191
YB       *69  *178   190
YT       *70  *179   192
```

```
                  ***  STATEMENT NUMBERS  ***

     1      75   *84
     2      61   *91
     3      45  *104
     4      49  *120
     5      54  *133
     6     138  *148
     7     177  *156
     8      80    87   100   116   129   144   151  *163
```

```
                  ***  VARIABLES  ***

AMAX0     176   178
AMIN0     177   179
AMIN1      91    93   113   126   156   159   168
ANGBD1    *55    70  *104   108   139  *120   121   123  *134   136
ANGBD2    *66    75  *135   138
ATAN       [illegible]
BDLEN      11   179
BDWID$    *24    37    38    45    49
BOARDW     11    24    74    78    84    92    94   103   110   122   123   137
          141   148   158   161   177
```

Subroutine CORE


```
C     SUBROUTINE 'CORE' LOCATES THE CORE DEFECT ON THE BOARD FACE        COREL
C                                                                        COREL
C                                                                        COREL
      SUBROUTINE COREL (CRADUS,COFSET,CANGS,CORFLG,I)                    COREL
C                                                                        COREL
      IMPLICIT REAL (K,L)                                                COREL
      INTEGER ID(22,2),NODEFC(2),CORFLG                                  COREL
      REAL FLX(26,2),FLY(26,2),FUX(26,2),FUY(26,2)                       COREL
      COMMON /BOARD/ L,D,K,T,LOG$,RB,DISTBD,BDLEN,BOARDW,KNOTS,R         COREL
     *       /DEFEC/ NODEFC,FLX,FLY,FUX,FUY,ID                           COREL
C                                                                        COREL
C                                                                        COREL
C     FIND DISTANCE TO THE BOARD FACE FROM THE CENTER OF THE CORE. EXIT  COREL
C     IF THE BOARD FACE IS BEYOND THE CORE RADIUS                        COREL
C                                                                        COREL
      TDIST = DISTBD-COFSET*COS(RAD(CANG$))                              COREL
      IF ((RADUS.LE.TDIST) RETURN                                        COREL
      NODEFC(1) = NODEFC(1)+1                                            COREL
C                                                                        COREL
C     LIMIT NUMBER OF DEFECTS TO MAXIMUM ALLOWED BY GRADING PROGRAM (22) COREL
C                                                                        COREL
      IF (NODEFC(1).GT.22) RETURN                                        COREL
C                                                                        COREL
C     LOCATE THE CENTER POINT OF THE INTERSECTION OF THE BOARD FACE AND  COREL
C     THE CORE DEFECT.  CALCULATE THE DISTANCE ON THE BOARD FROM THE     COREL
C     CENTER POINT TO THE EDGES OF THE CORE DEFECT.  THE CORE DEFECT     COREL
C     IS NOT TAPERED, SO IT WILL EXTEND THE FULL LENGTH OF THE BOARD     COREL
C                                                                        COREL
      CONST = SQRT(CRADUS**2-TDIST**2)                                   COREL
      TBDCTR = COFSET*SIN(RAD(CANG$))                                    COREL
      XR = BOARDW+CONST-TBDCTR                                           COREL
      XL = BOARDW-CONST-TBDCTR                                           COREL
C                                                                        COREL
C     SAVE CORE DEFECT COORDINATES FOR GRADING PROGRAM                   COREL
C                                                                        COREL
      FUX(NODEFC(1),1) = INT(BDLEN*4.)                                   COREL
      FLX(NODEFC(1),1) = 0.0                                             COREL
      FUY(NODEFC(1),1) = AMIN0(INT(BOARDW*8.),INT(XR*4.+1.))             COREL
      FLY(NODEFC(1),1) = AMAX0(0,INT(XL*4.))                             COREL
      ID(NODEFC(1),1) = 7                                                COREL
C                                                                        COREL
C     IF A CORE DEFECT IS FOUND ON THE OUTER FACE OF A BOARD, THE BOARD  COREL
C     WILL BE RERIPPED IN AN ATTEMPT TO INCREASE ITS GRADE AND VALUE     COREL
C                                                                        COREL
      IF (I.EQ.1) CORFLG = 1                                             COREL
      RETURN                                                             COREL
      END                                                                COREL
```

*** VARIABLES ***

```
AMAX0       39
AMIN0       38
RADLEN       9    36
BOARDW       9    31    32    38
CANG$        4    16    30
COFSET       4    16    30
CONST      *29    31    32
COREL        1
CORFLG       1     7   *45
COS         16
CRADUS       1    1[illegible]  [illegible]
D            [illegible]
DISTBD       [illegible]    16
FLX          8     9   *37
FLY          8     9   *39
FUX          8     9   *36
FUY          8     9   *38
I            4    45
ID           7     9   *40
INT         36    38    39
K            9
KNOTS        9
L            9
LOG$         9
NODEFC       7     9   *18    22    36    37    38    39    40
R            [illegible]
RAD         16    30
RB           9
SIN         30
SQRT        29
T            9
TBDCTR     *30    31    32
TDIST      *16    17    29
XL         *32    39
XR         *31    38
```

Subroutine GRADE

```
C     SUBROUTINE 'GRADE' CALLS THE U.S. FOREST PRODUCTS LABORATORY          GRADE
C     GRADING PROGRAM                                                       GRADE
C                                                                           GRADE
C                                                                           GRADE
        SUBROUTINE GRADE (NPG)                                              GRADE
C                                                                           GRADE
        INTEGER ID(22,2),NODEFC(2),IX(22)                                   GRADE
        REAL FLX(26,2),FLY(26,2),FUX(26,2),FUY(26,2)                        GRADE
        COMMON /DEFEC/ NODEFC,FLX,FLY,FUX,FUY,ID                            GRADE
C                                                                           GRADE
C                                                                           GRADE
        M = NODEFC(1)                                                       GRADE
        IF (M.EQ.0) GO TO 2                                                 GRADE
        IF (M.GT.22) GO TO 3                                                GRADE
        DO 1 J=1,M                                                          GRADE
          IX(J) = ID(J,1)                                                   GRADE
      1 CONTINUE                                                            GRADE
      2 NPG = 1                                                             GRADE
        CALL GRD (M,IX,NPG,SM)                                              GRADE
        RETURN                                                              GRADE
C                                                                           GRADE
C     IF NUMBER OF DEFECTS IS MORE THAN GRADING PROGRAM CAN HANDLE.        GRADE
C     THE GRADE IS 3A/3B                                                    GRADE
C                                                                           GRADE
      3 WRITE (6,100)                                                       GRADE
    100 FORMAT (' DEFECT LIMIT EXCEEDED')                                   GRADE
        NPG = 5                                                             GRADE
        RETURN                                                              GRADE
        END                                                                 GRADE
```

*** STATEMENT NUMBERS ***

1	15	*17
2	13	*18
3	14	*25
100	[illegible]	*26

*** VARIABLES ***

FLX	8	9			
FLY	8	9			
FUX	8	9			
FUY	8	9			
GRADE	5				
GRD	19				
ID	7	9	16		
IX	7	*16	19		
J	*15	16			
M	*12	13	14	15	19
NODEFC	7	9	12		
NPG	5	*18	19	*27	
SM	19				

Subroutine PRICE

```
C     SUBROUTINE 'PRICE' DETERMINES THE BOARD GRADE BASED ON THE GRADES    PRICEL
C     OF BOTH SIDES (OLD,NPG) AND CALCULATES THE BOARD VALUE               PRICEL
C                                                                          PRICEL
C                                                                          PRICEL
        SUBROUTINE PRICEL (BF,OLD,NPG,TVAL,PERC,V)                         PRICEL
C                                                                          PRICEL
        INTEGER OLD                                                        PRICEL
        REAL PERC(5)                                                       PRICEL
        COMMON /PRICE/ FAS,SEL,ONEC,TWOC,THRB                              PRICEL
C                                                                          PRICEL
C                                                                          PRICEL
C     IS EITHER SIDE BELOW GRADE?                                          PRICEL
C                                                                          PRICEL
        IF (OLD.EQ.6.OR.NPG.EQ.6) RETURN                                   PRICEL
C                                                                          PRICEL
C     DETERMINE THE BOARD GRADE                                            PRICEL
C                                                                          PRICEL
        IF (OLD.EQ.5.OR.NPG.EQ.5) GO TO 3                                  PRICEL
        IF (OLD.EQ.2.OR.NPG.EQ.2) GO TO 1                                  PRICEL
        IF (OLD.EQ.NPG) GO TO 4                                            PRICEL
        NPG = MAX0(OLD,NPG)                                                PRICEL
        IF (NPG.EQ.4) GO TO 8                                              PRICEL
        NPG = 2                                                            PRICEL
        GO TO 6                                                            PRICEL
      1 IF (OLD.EQ.4.OR.NPG.EQ.4) GO TO 2                                  PRICEL
        NPG = 3                                                            PRICEL
        GO TO 4                                                            PRICEL
      2 NPG = 4                                                            PRICEL
        GO TO 4                                                            PRICEL
      3 NPG = 5                                                            PRICEL
      4 GO TO (5,6,7,8,9), NPG                                             PRICEL
C                                                                          PRICEL
C     AND CALCULATE THE BOARD VALUE                                        PRICEL
C                                                                          PRICEL
      5 V = BF*FAS                                                         PRICEL
        GO TO 10                                                           PRICEL
      6 V = BF*SEL                                                         PRICEL
        GO TO 10                                                           PRICEL
      7 V = BF*ONEC                                                        PRICEL
        GO TO 10                                                           PRICEL
      8 V = BF*TWOC                                                        PRICEL
        GO TO 10                                                           PRICEL
      9 V = BF*THRB                                                        PRICEL
C                                                                          PRICEL
C     INCREASE SURFACE MEASURE IN THIS GRADE AND THE TOTAL LOG VALUE      PRICEL
C                                                                          PRICEL
     10 PERC(NPG) = BF+PERC(NPG)                                           PRICEL
        TVAL = TVAL+V                                                      PRICEL
        RETURN                                                             PRICEL
        END                                                                PRICEL
```

*** 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	22	31	*41		
9	31	*43			
10	36	38	40	42	*47

Subroutine RERIP

```
C   SUBROUTINE 'RERIP' DETERMINES WHETHER RIPPING A BOARD INTO TWO OR         RERIP
C   THREE PIECES WILL IMPROVE ITS VALUE WHEN A CORE DEFECT IS FOUND ON        RERIP
C   THE OUTER FACE (CORFLG = 1) AND THE GRADE IS BELOW SELECTS.  THE          RERIP
C   CORE DEFECT BOUNDRIES ON THE OUTER FACE DETERMINE THE RIP LINES           RERIP
C                                                                             RERIP
C                                                                             RERIP
      SUBROUTINE RERIP (TVALRP,PERCRP,NPG,V,CORFLG,I,BF)                      RERIP
C                                                                             RERIP
      IMPLICIT REAL (K,L)                                                     RERIP
      INTEGER CORFLG,ID(22,2),NODEFC(2),GRDRGT,GRDMID,RGRADE,GRDLFT,          RERIP
     .        LGRADE                                                          RERIP
      REAL FLX(26,2),FLY(26,2),FUX(26,2),FUY(26,2),PERCRP(5),DUMYRP(5)        RERIP
      COMMON /DEFEC/ NODEFC,FLX,FLY,FUX,FUY,ID                                RERIP
     .       /FORRIP/ RPLOSS,RPKERF,LFTLOC,RGTLOC,GRDRGT,GRDLFT,GRDMID,       RERIP
     .                TLOSS                                                   RERIP
     .       /BOARD/ L,D,K,T,LOGS,RB,DISTBD,BDLEN,BOARDW,KNOTS,R              RERIP
C                                                                             RERIP
C                                                                             RERIP
C   CONVERT BOARD WIDTH TO INCHES                                             RERIP
C                                                                             RERIP
      WIDTH = FLY(26,1)/4.                                                    RERIP
      IF (I.EQ.2) GO TO 8                                                     RERIP
C                                                                             RERIP
C                                                                             RERIP
C   OUTER FACE OF BOARD TO BE RIPPED...                                       RERIP
C                                                                             RERIP
C   IF NO CORE DEFECT, DO NOTHING                                             RERIP
C                                                                             RERIP
      TLOSS = 0.0                                                             RERIP
      IF (CORFLG.EQ.0) RETURN                                                 RERIP
C                                                                             RERIP
C  FIND THE CORE DEFECT                                                       RERIP
C                                                                             RERIP
      M = NODEFC(1)                                                           RERIP
      INDEX = 1                                                               RERIP
      DO 1 J=1,M                                                              RERIP
        IF (ID(J,1).EQ.7) INDEX = J                                           RERIP
    1 CONTINUE                                                                RERIP
      IF (ID(INDEX,1).NE.7) GO TO 12                                          RERIP
C                                                                             RERIP
C   DEFINE RIP LOCATIONS IN QUARTER INCH UNITS                                RERIP
C                                                                             RERIP
      LFTLOC = FLY(INDEX,1)                                                   RERIP
      RGTLOC = FUY(INDEX,1)                                                   RERIP
C                                                                             RERIP
C   GRADE OF MIDDLE PIECE ASSUMED 3A/3B. OTHERS FAS                           RERIP
C                                                                             RERIP
      GRDMID = 5                                                              RERIP
      RGRADE = 1                                                              RERIP
      LGRADE = 1                                                              RERIP
C                                                                             RERIP
C   CALCULATE PIECE WIDTHS.  IF EITHER SIDE IS TOO SMALL TO MAKE LUMBER       RERIP
C   (<3"), ADD IT TO THE MIDDLE PIECE                                         RERIP
C                                                                             RERIP
      WIDRGT = WIDTH-RGTLOC/4.                                                RERIP
      IF (WIDRGT-3.) 2,3,3                                                    RERIP
    2 RGTLOC = WIDTH*4.+RPKERF*4.                                             RERIP
      RGRADE = 6                                                              RERIP
    3 WIDLFT = LFTLOC/4.                                                      RERIP
      IF (WIDLFT-3.) 4,5,5                                                    RERIP
    4 LFTLOC = -RPKERF*4.                                                     RERIP
      LGRADE = 6                                                              RERIP
    5 WIDMID = (RGTLOC-LFTLOC)/4.-2.*RPKERF                                   RERIP
      IF (WIDMID-3.) 6,7,7                                                    RERIP
C                                                                             RERIP
C   IF MIDDLE IS TOO SMALL TO MAKE LUMBER, ADD ITS SURFACE MEASURE TO         RERIP
C   THE RIP LOSS                                                              RERIP
C                                                                             RERIP
    6 TLOSS = AMAX1(0.,(WIDMID*BDLEN*T/144.))                                 RERIP
      GRDMID = 6                                                              RERIP
C                                                                             RERIP
C   GET GRADE OF OUTER FACES                                                  RERIP
C                                                                             RERIP
    7 CALL RIP (RGRADE,LGRADE)                                                RERIP
C                                                                             RERIP
```

**** VARIABLES ****

BF	5	35	37	39	41	43	47					
FAS	9	35										
MAXB	21											
NPG	5	14	18	19	20	*21	22	*23	25	*26	*28	*30
	31	47										
OLD	5	7	14	18	19	20	21	25				
ONEC	9	39										
PERC	5	8	*47									
PRICEL	5											
SEL	9	37										
THRB	9	43										
TVAL	5	*48										
TWOC	9	41										
V	5	*35	*37	*39	*41	*43	48					

```
C    SAVE OUTER FACE GRADES                                          RERIP
C                                                                    RERIP
       GRDRGT = RGRADE                                               RERIP
       GRDLFT = LGRADE                                               RERIP
       RETURN                                                        RERIP
C                                                                    RERIP
C                                                                    RERIP
C    INNER FACE OF BOARD TO BE RIPPED...                             RERIP
C                                                                    RERIP
     8 RIPVAL = 0.                                                   RERIP
       OLDVAL = V                                                    RERIP
       DO 9 J=1,5                                                    RERIP
         DUMYRP(J) = 0.                                              RERIP
     9 CONTINUE                                                      RERIP
C                                                                    RERIP
C    IF BOARD IS FAS OR SELECTS.                                     RERIP
C                                                                    RERIP
       IF (NPG.EQ.1.OR.NPG.EQ.2) GO TO 11                            RERIP
C                                                                    RERIP
C    OR IF NO OUTER FACE CORE DEFECT FOUND, DO NOT RIP               RERIP
C                                                                    RERIP
       IF (CORFLG.EQ.0) GO TO 11                                     RERIP
C                                                                    RERIP
C    ANY PIECE THAT IS TOO SMALL TO MAKE LUMBER IS BELOW GRADE       RERIP
C                                                                    RERIP
       IF (GRDRGT.EQ.6) RGRADE = 6                                   RERIP
       IF (GRDLFT.EQ.6) LGRADE = 6                                   RERIP
C                                                                    RERIP
C    GET GRADE OF INNER FACES                                        RERIP
C                                                                    RERIP
       CALL RIP (RGRADE,LGRADE)                                      RERIP
C                                                                    RERIP
C    CALCULATE SURFACE MEASURE OF EACH PIECE                         RERIP
C                                                                    RERIP
       WIDMID = (RGTLOC-LFTLOC)/4.-2.*RPKERF                         RERIP
       SMMID = WIDMID*BDLEN*T/144.                                   RERIP
       SMRGT = (WIDTH-RGTLOC/4.)*BDLEN*T/144.                        RERIP
       SMLFT = LFTLOC/4.*BDLEN*T/144.                                RERIP
C                                                                    RERIP
C    DETERMINE GRADE AND VALUE OF EACH PIECE                         RERIP
C                                                                    RERIP
       IOLDRT = GRDRGT                                               RERIP
       IOLDLT = GRDLFT                                               RERIP
       IMGRD = GRDMID                                                RERIP
       CALL PRICEL (SMRGT,IOLDRT,RGRADE,RIPVAL,DUMYRP,V)             RERIP
       CALL PRICEL (SMLFT,IOLDLT,LGRADE,RIPVAL,DUMYRP,V)             RERIP
       CALL PRICEL (SMMID,IMGRD,IMGRD,RIPVAL,DUMYRP,V)               RERIP
C                                                                    RERIP
C    IF RERIPPING DOES NOT IMPROVE THE VALUE, DO NOT DO IT           RERIP
C                                                                    RERIP
       IF (RIPVAL.LE.OLDVAL) GO TO 11                                RERIP
C                                                                    RERIP
C    OTHERWISE, INCREASE RERIP VALUES                                RERIP
C                                                                    RERIP
       TVALRP = TVALRP+RIPVAL                                        RERIP
       DO 10 J=1,5                                                   RERIP
         PERCRP(J) = PERCRP(J)+DUMYRP(J)                             RERIP
    10 CONTINUE                                                      RERIP
C                                                                    RERIP
C    INCREASE LOSS DUE TO RERIPPING                                  RERIP
C                                                                    RERIP
       CORFLG = 0                                                    RERIP
       RPLOSS = RPLOSS+TLOSS                                         RERIP
       IF (RGRADE.NE.6) RPLOSS = RPLOSS+RPKERF*BDLEN*T/144.          RERIP
       IF (LGRADE.NE.6) RPLOSS = RPLOSS+RPKERF*BDLEN*T/144.          RERIP
       RETURN                                                        RERIP
C                                                                    RERIP
C    BOARD IS NOT TO BE RIPPED                                       RERIP
C                                                                    RERIP
    11 PERCRP(NPG) = PERCRP(NPG)+BF                                  RERIP
       TVALRP = TVALRP+OLDVAL                                        RERIP
    12 CORFLG = 0                                                    RERIP
       RETURN                                                        RERIP
       END                                                           RERIP
```

*** STATEMENT NUMBERS ***

1	36	*38		
2	56	*57		
3	56	*59		
4	60	*61		
5	60	*63		
6	64	*69		
7	64	*74		
8	22	*85		
9	87	*89		
10	131	*133		
11	93	97	126	*145
12	39	*147		

*** VARIABLES ***

AMAXI	69								
BDLEN	13	69	111	112	113	139	140		
BF	7	145							
BOARDW	13								
CORFLG	7	10	30	97	*137	*147			
D	13								
DISTBD	13								
DUMYRP	12	*88	120	121	122	132			
FLX	12	13							
FLY	12	13	21	43					
FUX	12	13							
FUY	12	13	44						
GRDLFT	10	13	*79	102	118				
GRDMID	10	13	*40	*70	119				
GRDRGT	10	13	*78	101	117				
I	7	22							
ID	10	13	37	39					
IMGRD	*119	122							
INDEX	*35	*37	39	43	44				
IOLDLT	*118	121							
IOLDRT	*117	120							
J	*36	37	*87	88	*131	132			
K	13								
KNOTS	13								
L	13								
LFTLOC	13	*43	59	*61	63	110	113		
LGRADE	10	*50	*62	74	79	*102	106	121	140
LOGS	13								
M	*34	36							
NOLEFL	10	13	34						
NPG	7	93	145						
OLDVAL	*86	126	146						
PERCRP	7	12	*132	*145					
PRICEL	120	121	122						
R	13								
RB	13								
RERIP	7								
RGRADE	10	*49	*58	74	78	*101	106	120	139
RGTLOC	13	*44	55	*57	63	110	112		
RIP	74	106							
RIPVAL	*85	120	121	122	126	130			
RPKERF	13	57	61	63	110	139	140		
RPLOSS	13	*138	*139	*140					
SMLFT	*113	121							
SMMID	*111	122							
SMRGT	*112	120							
T	13	69	111	112	113	139	140		
TLOSS	13	*29	*69	138					
TVALRP	7	*130	*146						
V	7	86	120	121	122				
WIDLFT	*59	60							
WIDMID	*63	64	69	*110	111				
WIDRGT	*55	56							
WIDTH	*21	55	57	112					

Subroutine RIP

```
  1 C   SUBROUTINE 'RIP' RIPS THE FACE INTO TWO OR THREE PIECES AND GETS    RIP
  2 C   THE GRADE OF EACH PIECE                                             RIP
  3 C                                                                       RIP
  4 C                                                                       RIP
  5       SUBROUTINE RIP (RGRADE,LGRADE)                                    RIP
  6 C                                                                       RIP
  7       IMPLICIT REAL (K-L)                                               RIP
  8       INTEGER ID(22,2),NODEFC(2),ITYPE(22),RGRADE,LGRADE,GRDRGT,GRDLFT, RIP
  9      .        GRDMID                                                    RIP
 10       REAL FLX(26,2),FLY(26,2),FUX(26,2),FUY(26,2),LX(22),LY(22),UX(22), RIP
 11      .     UY(22)                                                       RIP
 12       COMMON /FORRIP/ RPLOSS,RPKERF,LFTLOC,RGTLOC,GRDRGT,GRDLFT,GRDMID, RIP
 13      .                TLOSS                                             RIP
 14      .       /DEFEC/ NODEFC,FLX,FLY,FUX,FUY,ID                          RIP
 15 C                                                                       RIP
 16 C                                                                       RIP
 17 C   SAVE BOARD WIDTH                                                    RIP
 18 C                                                                       RIP
 19       WIDTH = FLY(26,1)                                                 RIP
 20 C                                                                       RIP
 21 C   SAVE TYPE, LOCATION, AND NUMBER OF DEFECTS                          RIP
 22 C                                                                       RIP
 23       DO 1 J=1,22                                                       RIP
 24         ITYPE(J) = ID(J,1)                                              RIP
 25         UX(J) = FUX(J,1)                                                RIP
 26         UY(J) = FUY(J,1)                                                RIP
 27         LX(J) = FLX(J,1)                                                RIP
 28         LY(J) = FLY(J,1)                                                RIP
 29     1 CONTINUE                                                          RIP
 30       M = NODEFC(1)                                                     RIP
 31 C                                                                       RIP
 32 C   IF EITHER PIECE IS BELOW GRADE, NOTHING IS DONE TO THAT PIECE       RIP
 33 C                                                                       RIP
 34 C   RIGHT (UPPER) PIECE                                                 RIP
 35 C                                                                       RIP
 36       IF (RGRADE.EQ.6) GO TO 3                                          RIP
 37 C                                                                       RIP
 38 C   LOCATE DEFECTS IN THIS PIECE, IGNORING THOSE WHICH LIE BELOW THE    RIP
 39 C   RIP LINE                                                            RIP
 40 C                                                                       RIP
 41       N = 0                                                             RIP
 42       DO 2 J=1,M                                                        RIP
 43         IF (UY(J).LE.RGTLOC) GO TO 2                                    RIP
 44         N = N+1                                                         RIP
 45         FUY(N,1) = UY(J)-RGTLOC                                         RIP
 46         FLY(N,1) = AMAX1(0.,(LY(J)-RGTLOC))                             RIP
 47         FUX(N,1) = UX(J)                                                RIP
 48         FLX(N,1) = LX(J)                                                RIP
 49         ID(N,1) = ITYPE(J)                                              RIP
 50     2 CONTINUE                                                          RIP
 51 C                                                                       RIP
 52 C   DETERMINE PIECE WIDTH: NUMBER OF DEFECTS                            RIP
 53 C                                                                       RIP
 54       FLY(26,1) = WIDTH-RGTLOC                                          RIP
 55       NODEFC(1) = N                                                     RIP
 56 C                                                                       RIP
 57 C   GRADE RIGHT (UPPER) PIECE                                           RIP
 58 C                                                                       RIP
 59       CALL GRADE (RGRADE)                                               RIP
 60 C                                                                       RIP
 61 C   LEFT (LOWER) PIECE                                                  RIP
 62 C                                                                       RIP
 63     3 IF (LGRADE.EQ.6) GO TO 5                                          RIP
 64 C                                                                       RIP
 65 C   LOCATE DEFECTS IN THIS PIECE, IGNORING THOSE WHICH LIE ABOVE THE    RIP
 66 C   RIP LINE                                                            RIP
 67 C                                                                       RIP
 68       N = 0                                                             RIP
 69       DO 4 J=1,M                                                        RIP
 70         IF (LY(J).GE.LFTLOC) GO TO 4                                    RIP
 71         N = N+1                                                         RIP
 72         FUY(N,1) = AMIN1(LFTLOC,UY(J))                                  RIP
 73         FLY(N,1) = LY(J)                                                RIP
 74         FUX(N,1) = UX(J)                                                RIP
 75         FLX(N,1) = LX(J)                                                RIP
 76         ID(N,1) = ITYPE(J)                                              RIP
 77     4 CONTINUE                                                          RIP
 78 C                                                                       RIP
 79 C   DETERMINE PIECE WIDTH: NUMBER OF DEFECTS                            RIP
 80 C                                                                       RIP
 81       FLY(26,1) = LFTLOC                                                RIP
 82       NODEFC(1) = N                                                     RIP
 83 C                                                                       RIP
 84 C   GRADE LEFT (LOWER) PIECE                                            RIP
 85 C                                                                       RIP
 86       CALL GRADE (LGRADE)                                               RIP
 87 C                                                                       RIP
 88 C   RESTORE TYPE, LOCATION, AND NUMBER OF DEFECTS; BOARD WIDTH          RIP
 89 C                                                                       RIP
 90     5 DO 6 J=1,22                                                       RIP
 91         ID(J,1) = ITYPE(J)                                              RIP
 92         FLY(J,1) = LY(J)                                                RIP
 93         FUY(J,1) = UY(J)                                                RIP
 94         FLX(J,1) = LX(J)                                                RIP
 95         FUX(J,1) = UX(J)                                                RIP
 96     6 CONTINUE                                                          RIP
 97       NODEFC(1) = M                                                     RIP
 98       FLY(26,1) = WIDTH                                                 RIP
 99       RETURN                                                            RIP
100       END                                                               RIP
```

```
                    *** STATEMENT NUMBERS ***

 1      23   *29
 2      42    43   *50
 3      36   *63
 4      69    70   *77
 5      63   *90
 6      90   *96

                    *** VARIABLES ***

AMAX1    46
AMIN1    72
FLX      10   12   27  *48  *75  *94
FLY      10   12   19   28  *46  *54  *73  *81  *92  *98
FUX      10   12   25  *47  *74  *95
FUY      10   12   26  *45  *72  *93
GRADE    59   86
GRDLFT    8   12
GRDMID    8   12
GRDRGT    8   12
ID        8   12   24  *49  *76  *91
ITYPE     8  *24   49   76   91
J       *23   24   25   26   27   28   42   43   45   46   47   48
         49  *69   70   72   73   74   75   76  *90   91   92   93
         94   95
LFTLOC   12   70   72   81
LGRADE    5    8   63   86
LX       10  *27   48   75   94
LY       10  *28   46   70   73   92
M       *30   42   69   97
N       *41  *44   45   46   47   48   49   55  *68  *71   72   73
         74   75   76   82
NODEFC    8   12   30  *55  *82  *97
RGRADE    5    8   36   59
RGTLOC   12   13   45   46   54
RIP       5
RPKERF   12
RPLOSS   12
```


☆ U.S. GOVERNMENT PRINTING OFFICE: 1980-651-111/15

Function DEG

```
C     FUNCTION DEG CONVERTS ITS ARGUMENT (IN RADIANS) TO DEGREES
[illegible]
      RETURN
      END
```

*** VARIABLES ***

Function RAD

```
C     FUNCTION RAD CONVERTS ITS ARGUMENT (IN DEGREES) TO RADIANS
[illegible]
```

*** VARIABLES ***