USDA
United States Department of Agriculture

Java Image Analysis Program for Lumber Knots, and Related Nonlinear Least Squares Routines for Fitting Lumber Modulus of Rupture to Strength Predictors

Steve P. Verrill
Frank C. Owens
Rubin Shmulsky
Robert J. Ross

Forest Service
Forest Products Laboratory
General Technical Report FPL–GTR–288
December 2021

Abstract

To properly evaluate the reliability of lumber structures, good models for the strength distributions of their components are needed. Modulus of rupture (MOR) distributions of grades of structural lumber have often been modeled as two-parameter Weibulls. However, in a series of papers, Verrill and others have established that strength properties of visual and machine stress rated grades of lumber are not distributed as two-parameter Weibulls and that modeling them as two-parameter Weibulls can yield large over- or underestimates of probabilities of breakage. Instead, grades of lumber have "pseudo-truncated" distributions. Recent research also established that the appropriate MOR model can change significantly with location and time and that model differences have practical significance. Verrill and others concluded that "there may be significant efficiencies that can be obtained through the development of computer models that yield real-time in-line estimates of lumber properties based on measurements of stiffness, specific gravity, knot size and location, slope of grain, and other strength predictors." Verrill and others have now published a paper that proposes such models and fits them to mill run data. In this report, we discuss a computer program that was developed and used to obtain the knot portion of the data and other computer programs that were developed and used to fit the models.

Keywords: modulus of rupture, modulus of elasticity, specific gravity, slope of grain, knots, Weibull distribution, pseudo-truncated distributions, strength prediction models, lumber strength models, knot measurement

December 2021

Verrill, Steve P.; Owens, Frank C.; Shmulsky, Rubin; Ross, Robert J. 2021. Java image analysis program for lumber knots, and related nonlinear least squares routines for fitting lumber modulus of rupture to strength predictors. General Technical Report FPL-GTR-288. Madison, WI: U.S. Department of Agriculture, Forest Service, Forest Products Laboratory. 16 p.

A limited number of free copies of this publication are available to the public from the Forest Products Laboratory, One Gifford Pinchot Drive, Madison, WI 53726-2398. This publication is also available online at www.fpl.fs.fed.us. Laboratory publications are sent to hundreds of libraries in the United States and elsewhere.

The Forest Products Laboratory is maintained in cooperation with the University of Wisconsin.

The use of trade or firm names in this publication is for reader information and does not imply endorsement by the United States Department of Agriculture (USDA) of any product or service.

Contents

Java Image Analysis Program for Lumber Knots, and Related Nonlinear Least Squares Routines for Fitting Lumber Modulus of Rupture to Strength Predictors

Steve P. Verrill, Mathematical Statistician
USDA Forest Service, Forest Products Laboratory, Madison, WI

Frank C. Owens, Assistant Professor
Department of Sustainable Bioproducts, Mississippi State University, MS

Rubin Shmulsky, Professor and Department Head
Department of Sustainable Bioproducts, Mississippi State University, MS

Robert J. Ross, Supervisory Research General Engineer
USDA Forest Service, Forest Products Laboratory, Madison, WI

1 Introduction

Properly evaluating the reliability of lumber structures requires good models for the strength distributions of the components of those structures. In the past, the modulus of rupture distribution of the lumber in a structure has often been modeled as a two-parameter Weibull. Verrill et al. (2012, 2013, 2014, 2015, 2019, 2020a) established theoretically and empirically that the strength properties of visual and machine stress rated (MSR) grades of lumber are not distributed as two-parameter Weibulls, and that modeling them as two-parameter Weibulls can yield large over- or under-estimates of probabilities of breakage. This led Verrill et al. (2017, 2018, 2020b), Owens et al. (2018, 2019, 2020), and Anderson et al. (2021) to investigate alternative models for strength distributions of lumber. However, Verrill et al. (2020b) performed work suggesting that even if we have adequate models for the *general parametric forms* of distributions (e.g., pseudo-truncated Weibull or pseudo-truncated mixed normal or ...), the *specific* strength distributions (due to specific parameter values) can vary significantly (from a practical as well as a theoretical viewpoint) with mill and time. They argued that this variability suggests that "there may be significant efficiencies that can be obtained through the development of computer models that yield real-time in-line estimates of lumber properties based on measurements of stiffness, specific gravity, knot size and location, slope of grain, and other strength predictors." Verrill et al. (2021) developed and fit models that predict modulus of rupture (MOR) from modulus of elasticity (MOE), specific gravity (SG), slope of grain (SOG), and knot data. These models were fit to the summer data discussed in Owens et al. (2018, 2019, 2020). To perform these fits, we needed to identify and "measure" knots in images of the front and back faces of 600 pieces of lumber. To do this, we developed an interactive graphics-based computer program, Knots_2x4 (written in the Java programming language), that eased the task. In this report, we describe the main image analysis program and the auxilliary programs that were used in the model fitting process. We also discuss the mathematics behind the knot measurement process.

2 Java Programs Used in Our Analyses

All the routines listed below were written in the Java programming language (see, for example, `https://www.oracle.com/java/technologies/javase-downloads.html` for the developer's kit and `https://www.java.com` for the runtime environment), and listings of all these routines are available at `https://www1.fpl.fs.fed.us/knots_2x4.html`.

Knots_2x4.java is the main image analysis program. To fit ellipses to knots, it makes use of Java versions of the Uncmin (Schnabel et al. 1982) and fmin optimization routines. These Java versions can be obtained at `https://www1.fpl.fs.fed.us/optimization.html`. Uncmin, in turn, makes use of a utility program, Ellipses_knots.java, that can be obtained at `https://www1.fpl.fs.fed.us/knots_2x4.html`.

The nonlinear least squares programs ("Nreg" programs) listed below also make use of the Java version of the Uncmin nonlinear optimization program referenced above.

Nreg2.java, Nreg3.java, Nreg4.java, and Nreg5.java are nonlinear least squares programs that take the knot image analysis results from Knots_2x4.java and combine them with MOR, MOE, SG, and SOG data to produce fits to the MOR prediction Models 2–5 discussed in sections 5 and 6 of Verrill et al. (2021).

These programs and models are also used to obtain predictions of board strengths when bending and tension edges are exchanged via a board flip. The results of such flips are discussed in section 10.2 of Verrill et al. (2021).

Programs Nreg2_300.java, Nreg3_300.java, Nreg4_300.java, and Nreg5_300.java are similar to programs Nreg2.java, Nreg3.java, Nreg4.java, and Nreg5.java. The former models differ from the latter in that they fit models to random subsets of 300 of the 590 data points and then use these fits to predict the strengths of the remaining 290 points. The more realistic (increased) root mean squared errors from these fits are discussed in section 7 of Verrill et al. (2021).

Programs Nreg2_above4.java, Nreg3_above4.java, Nreg4_above4.java, and Nreg5_above4.java are also similar to programs Nreg2.java, Nreg3.java, Nreg4.java, and Nreg5.java. The former models differ from the latter in that they fit models to Number 3 and better data rather than to the full mill run data.

Programs Nreg2_above3.java, Nreg3_above3.java, Nreg4_above3.java, and Nreg5_above3.java are similar to programs Nreg2.java, Nreg3.java, Nreg4.java, and Nreg5.java. The former models differ from the latter in that they fit models to Number 2 and better data rather than to the full mill run data.

The "above 4" and "above 3" results are discussed in section 4 of Verrill et al. (2021).

Nreg6_aux.java contains auxilliary routines needed by Nreg4.java, Nreg5.java, Nreg4_300.java, Nreg5_300.java. Nreg4_above4.java, Nreg5_above4.java, Nreg4_above3.java, and Nreg5_above3.java.

SS_log_mor2.java, SS_log_mor3.java, SS_log_mor4.java, and SS_log_mor5.java are sums of squares routines needed by Nreg2.java, Nreg3.java, Nreg4.java, Nreg5.java, Nreg2_300.java, Nreg3_300.java, Nreg4_300.java, and Nreg5_300.java.

SS_log_mor2_above4.java, SS_log_mor3_above4.java, SS_log_mor4_above4.java, and SS_log_mor5_above4.java are sums of squares routines needed by Nreg2_above4.java, Nreg3_above4.java, Nreg4_above4.java, and Nreg5_above4.java.

SS_log_mor2_above3.java, SS_log_mor3_above3.java, SS_log_mor4_above3.java, and SS_log_mor5_above3.java are sums of squares routines needed by Nreg2_above3.java, Nreg3_above3.java, Nreg4_above3.java, and Nreg5_above3.java.

The Knots_2x4.java program is described in greater detail below.

3 Knots_2x4.java

Knots_2x4 permits a user to obtain (relatively) quick and accurate estimates of the locations, sizes, and orientations of knots on the two faces of a piece of lumber. In the following sections, we discuss the necessary input and output files and the sequence of steps that one must perform to analyze a series of boards via Knots_2x4. Because this software requires detailed human interaction, it is useful for research purposes, but not for production purposes.

3.1 Start-up Panel

The start-up panel contains four lines for information that needs to be supplied by the user. An example of a Knots_2x4 start-up panel is provided in Figure 1.

The first two lines of the start-up panel must contain the names of two input files. (As noted in the Introduction, these models were fit to the summer data discussed in Owens et al. (2018, 2019, 2020).) The third and fourth lines of the panel must contain the names of two output files:

Line 1: Image names file. For example, the first three lines in our summer front image names file, summer_knot_image_names_F, were

201-202-F.JPG
203-209-F.JPG
210-216-F.JPG

Images were produced of both the front and back faces of 600 pieces of lumber produced in the summer. See Figure 2 for the 201-202-F.JPG image of boards 201 and 202.

Line 2: Number of boards per image file. For example, the first three lines in our summer number of boards per image file, boards_per_image_summer, were

2
7
7

These lines specify that there are two boards in the 201-202-F.JPG image and seven boards in each of the next two images.

Line 3: Results file. For example, for our run on summer front face images, our results file was called summer_knot_F.res. The output printed in summer_knot_F.res for the front image of board 201 is given below. Some of this material is discussed Sections 4 – 6.

boardid:
201

board_width:
3.42

tLy and tRy:
2124 2125

dtLxd4, dtLyd4 and dtRxd4, dtRyd4:
97.5 531.0 1197.75 531.25

dbLxd4, dbLyd4 and dbRxd4, dbRyd4:
98.25 592.5 1198.25 594.5

dtMxd4, dtMyd4 and dbMxd4, dbMyd4:
647.625 531.125 647.6108273488144 593.498837867907

ptocm[1] and ptocm[2]:
0.137359687334112 0.13926992596252868

number of knots:
3

a,b,theta,x0,y0:

0.801571117 0.502110515 -0.465998384 49.215151889 74.566814841
2.941733466 2.007718889 -0.014318440 80.633043158 77.683696514
0.326319130 0.448557725 -1.390748316 134.52800699 77.770350545

1 - p11, 1 - p12, 1 - p13, 1 - p14, 1 - p15, 1 - p16
1 - p21, 1 - p22, 1 - p23, 1 - p24, 1 - p25, 1 - p26
1 - p31, 1 - p32, 1 - p33, 1 - p34, 1 - p35, 1 - p36
1 - p41, 1 - p42, 1 - p43, 1 - p44, 1 - p45, 1 - p46
1.0 0.9772 0.9788 1.0 1.0 1.0
1.0 1.0 0.787 1.0 0.992 1.0
1.0 1.0 0.8968 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0

1 - p1, 1 - p2, 1 - p3, 1 - p4, 1 - p5
1.0 0.87 0.54 0.9299999999999999 1.0

Line 4: Comments file. This file contains notes that were made about various boards as Knots_2x4 was used to obtain knot fits. For example, for the summer front face image of board 212, the comments file contains:

> 212: Machine (?) damage along lower right edge
> (6 equi-spaced "punches"?)

This file can serve as an informal (and searchable) "Lab notebook" as a researcher works through the knot fits for the various boards (and board faces). Entries are automatically written in the comments file (and erased in the comments text area that appears on the computer screen) with each click of the "next board" button.

In addition to the four files (two input, two output) that must be identified on the start-up panel, three other files (or file-types) are required by some of our programs.

The Knots_2x4 program requires that image files corresponding to the names listed in the image names file (such as 201-202-F.JPG) exist in the directory from which Knots_2x4 is being run. In addition, the Knots_2x4 program requires that a file (called summer_widths in our use of the program) exists in the directory from which Knots_2x4 is being run. This file must contain the approximate widths in inches (at the center of the 59.5 inch length region between the supports) of the boards. For us, the first three lines (corresponding to our boards 201, 202, and 203) of this file were

> 3.42
> 3.51
> 3.47

These values are used in the pixel/cm calibration process.

The third additional file needed by our programs (called franks_data_summer_minus_10 in our implementation) is used by the Nreg/SS_log nonlinear least squares routines. For each board in the data set, it contains 12 entries. For our board 201, those entries are

> 201 1 1.75 1.63 2 1809684 9197 0.47 1.53 3.42 98 22.2

and they correspond to board number, mill ID, Dir-E in millions of pounds per square inch (Mpsi), E-comp in Mpsi, visual grade, MOE in pounds per square inch (psi), MOR in psi, specific gravity, board thickness in inches, board width in inches, board length in inches, and slope of grain in degrees. (See Owens et al. (2018) for definitions of "Dir-E" and "Ecomp-E".) To modify this program for use on different data, one would need a bit of assistance from a Java programmer.

After the four input lines in the start-up panel are filled and the "Go" button on the start-up panel is clicked, the main image panel (Knots_2x41) is displayed (see Fig. 2). In the following sections we describe the manner in which a user must interact with the panel for knot information to be calculated and stored. As we work through the steps that a user must take, we will also point to sections of this report that describe the calculations that are being performed by the program as well as the manner in which knot information is then stored in the results files.

3.2 Calibration

For the purposes of taking photographic face images, boards were stacked either "2 tall" (see Fig. 2) or "7 tall" (see Fig. 3).

The boards were not, of course, *tested* in a stacked configuration, but instead as individual boards under "third point loading." That is, they were tested with equal loads on the compression edge at points one-third and two-thirds the distance from the left support to the right support. Under our third point loading scheme, the board supports were placed 59.5 inches apart and the equal loads were applied to the compression edge from load heads that were placed 59.5/3 and 119/3 inches from a vertical line through the left board support.

Our images were 5184 horizontal pixels by 3456 vertical pixels. They were taken by a camera that was approximately 8.5 feet from the vertical "wall" of boards. If someone were working with different resolution photographs and wanted to use our software, portions of our programs would need to be altered.

For each board in an image, starting with the top board (lowest board ID given the way that the boards were marked and stacked) and working down, we calibrate pixels to distances by knowing the nominal width of each board (at least near its center) and the nominal distance between the vertical green lines at the left and right support points. The mathematics of the calibration are described in Section 4.

To initiate an analysis on a board in a particular image, the user must

1. Make sure that the image's name is typed into the "Load image:" text field (for example, 201-202-F.JPG) at the bottom of the page.

2. Make sure that the id of the board (within the image) that one wants to begin with (for example, 202) is typed into the "Board:" text field at the bottom of the page.

3. Click the "Load image:" button. The remaining text fields at the bottom of the panel will then be automatically filled appropriately.

At this point the user is ready to calibrate the board. We note that this "Load image:" action needs to be done only at the beginning of a data collection session. Subsequent new boards (and new images) automatically become the focus of the analysis with each click of the "next board" button at the right of the page. See below.

To begin the calibration of a new board after an initial "Load image:" button click or a subsequent "next board" button click, the user clicks on the "calibr board" button at the top right of the page, and then follows these steps:

1. The user must use the mouse to perform a "mouse-button-down, mouse-drag, mouse-button-up" to form a box around the vertical line (green in our images) at the *left* board support. A magnified version of this box will then appear at the top center of the image. The user must then click in the magnified box where the vertical line meets the *top* edge of the board (a + will appear in the magnified image at that point), and then click in the magnified box at the spot on the *bottom* edge of the board that lies (approximately) directly below the + on the top edge. This will create a + at the bottom click point and a line will be drawn between the top and bottom +'s in the magnified image. If, at any point in this process, the user feels that the +'s are misplaced, the user can click the "back" button next to the "calibr" button (top right of the page) and the latest click will be erased.

2. After the user is satisfied with the placement of the two +'s in the magnified image of the left "vertical" line in the image (at the left board support), the user needs to repeat this sequence at the *right* "vertical" line (at the right board support location). See Figure 3. Again the user can correct perceived errors in the placement of +'s by clicking the back button. *However* (and this is one of the inelegant, user-unfriendly features of the current version), one cannot click the back button to get back to the left end of the board. To do that, the user needs to click the "cancel" button in the upper right corner of the page and then needs to redo both the left and right pairs of +'s.

3. After the user is satisfied that the right pair of +'s are good enough (the top + at the location at which the "vertical" line intersects the top edge of the board, and the bottom + on the bottom edge and "directly below" the top +), the user accepts the calibration by clicking the "submit" button immediately below the "calibr board" button in the upper right corner of the page.

We note that this process might appear to be complex, but once a user gains some experience, the calibration process will take about thirty seconds (longer if one uses the back button).

3.3 Fitting a Knot

After the board has been calibrated (the relationships between horizontal and vertical pixel measurements and horizontal and vertical length measurements have been established for a board), knot measurements (knot location, size, and orientation) can be made. We approximate knots with ellipses. (See Section 5 for mathematical details.) To begin the fitting process (after board calibration), a user first clicks the "ellipses" button in the center right of the page. The user then must do "mouse-button-down, mouse-drag, mouse-button-up" to form a box around a knot. A magnified version of this box will then appear at the top of the page. The user then must perform eight clicks to characterize the knot (Fig. 4). The first two clicks should be made at the (approximate) endpoints of the ellipse's major (longer) axis. The next two clicks should be made at the (approximate) endpoints of the ellipse's minor (shorter) axis. The remaining four clicks can be made anywhere on the ellipse's perimeter. We tended to click on the perimeter of the ellipse in each of the ellipse's four quadrants.

Some knots are edge knots. The full ellipse is not available. In that case a user might need to include off-board, human-imagined points to obtain an ellipse that appears to yield a good fit to the portion of the knot that is on the board.

As points "on" the ellipse perimeter are clicked, +'s appear. If the number of clicks (and thus +'s) lies between 1 and 8, inclusive, the user can choose to click the ellipse "back" button (to the right of the "ellipses" button). This

erases the latest +, and the user can keep clicking the "back" button until no +'s are displayed in the magnified knot image. However a user should *not* click the ellipse "back" button if there are no +'s in the magnified image of the knot. If the user does click the back button while there are no +'s showing in the magnified image, the user will then need to click the "cancel" button to the right and below the "ellipses" button to be able to resume work (and will need to click the "ellipses" button and do another "mouse-button-down, mouse-drag, mouse-button-up" to form another magnified box around a knot on the board).

When satisfied with a fitted ellipse, the user should click the "submit" button below the "ellipses" button. To fit another knot on the board, the user should repeat the click of the "ellipses" button, the "mouse-button-down, mouse-drag, mouse-button-up" to form a box around the new knot, the eight clicks on the perimeter of the new knot, and the click of the "submit" button below the "ellipses" button. After each "submit" click, the ellipse count in the "ellipse #:" text field is incremented by 1.

After all knots on the board face have been satisfactorily fit, the user should click the "next board" button below the "ellipse #" label. This will write the estimated knot results for the current board to the results file and increment the material in the boxes at the bottom of the page as appropriate.

3.4 Magnifying

Magnified windows are used to help in the calibration and fitting procedures. However, it is also useful to be able to see magnified portions of boards outside of those procedures. For example, one might be interested in looking at potential machine damage, shake, wane, ... to determine whether such features might account for a low strength value. For this reason, our program provides magnified views of arbitary sections of the images.

To obtain a magnified view of a portion of an image, click the "magnify" button at the lower right of the panel. Then do a "mouse-button-down, mouse-drag, mouse-button-up" to form a box around the area to be magnified. The magnified portion of the image appears at the top of the panel (Fig. 5). To clear this image, click on the "end magnify" button to the right of the "magnify" button.

3.5 Comments

We found it convenient to have a means to write and save comments associated with particular boards as we proceeded through an analysis. To write comments about the current board (noting, for example, machine damage or holes or shake), type the comments in the comments text area at the bottom right of the panel. These comments must be added before the "next board" button is clicked.

3.6 Stop

The "Stop" button at the bottom of the panel is used to completely stop the current data collection session. (Of course, a user could "stop" a session by simply locking the computer screen and taking a break. To resume after such a break, a user would not have to go through the steps described below. These steps are needed only after a click of the "Stop" button ends a session.)

Before clicking the "Stop" button the user should note the image name (in the "Load image:" text field at the bottom of the panel) and the last board completed in the image (in the "Board:" text field at the bottom of the panel). The user will need this stopping information to know where to begin the next data collection session. For example, as noted above, the first two images in our summer front face image file were 201-202-F.JPG and 203-209-F.JPG. That is, the first image was of the front faces of boards 201 and 202. The second image was of the front faces of boards 203 – 209. If we stopped a session after completing an analysis of board 202, then at the beginning of our next session, after hitting the "Go" button at the bottom of the start-up panel, we would see the first image (201-202-F.JPG) in the panel. To start instead at the first board in the 203–209 image, we would fill the "Load image:" text field with 203-209-F.JPG and the "Board:" text field with 203. We would then click the "Load image:" button and proceed with the steps needed to analyze board 203. If, instead, we stopped a session after completing an analysis of board 205 in the 203-209-F.JPG image, we would begin the next session (after hitting the "Go" button at the bottom of the start-up panel) by filling the "Load image:" text field with 203-209-F.JPG and the "Board:" text field with 206, and then clicking the "Load image:" button.

4 Calibration "Math"

To work with actual lumber dimensions rather than pixels (of an image), it was necessary to identify and calibrate each board through a series of mouse actions. As noted in Section 3.2, the images were of either two or seven stacked board faces. To calibrate a board, we first clicked on the calibrate button. Then we did a mouse-button-down, mouse-drag, mouse-buttom-up to obtain a magnified image of the region of board directly above the left support of the board under third point loading (as indicated by a green "vertical" line at the left "end" of the board). We clicked on the magnified image at the top edge of the board above the left support point (at the top of the green line) and then clicked on the magnified image at the bottom edge of the board directly below the top edge click. This yielded two points in pixel coordinates. Here we will refer to these points as $(dtLxd4, dtLyd4)$ and $(dbLxd4, dbLyd4)$. (This notation is somewhat ugly but it matches the notation in our Java program. "d" indicates a double precision value. "t" indicates "top". "b" indicates "bottom". "L" indicates "left". "d4" indicates that, rather than working with the original 5184×3456 pixel camera images, we are working with images that have had their resolutions reduced to 1296×864 so that they can be fully displayed on our Java canvas.)

A similar mouse-button-down, mouse-drag, mouse-button-up yielded a magnified image of the region of board directly above the right support of the board under third point loading (as indicated by a green "vertical" line at the right "end" of the board). Again, we first clicked on the magnified image at the top edge of the board above the right support point (at the top of the green line) and then clicked on the magnified image at the bottom edge of the board directly below the top edge click. This yielded two more points in pixel coordinates. Here we will refer to these points as $(dtRxd4, dtRyd4)$ and $(dbRxd4, dbRyd4)$. ("R" indicates "right".)

The "top edge line" is the line through the points $(dtLxd4, dtLyd4)$ and $(dtRxd4, dtRyd4)$. Its equation is

$$y = a_{\text{top}} + b_{\text{top}} \times x$$

where

$$b_{\text{top}} = (dtRyd4 - dtLyd4)/(dtRxd4 - dtLxd4)$$

and

$$a_{\text{top}} = dtLyd4 - b_{\text{top}} \times dtLxd4$$

The "bottom edge line" is the line through the points $(dbLxd4, dbLyd4)$ and $(dbRxd4, dbRyd4)$. Its equation is

$$y = a_{\text{bott}} + b_{\text{bott}} \times x$$

where

$$b_{\text{bott}} = (dbRyd4 - dbLyd4)/(dbRxd4 - dbLxd4)$$

and

$$a_{\text{bott}} = dbLyd4 - b_{\text{bott}} \times dbLxd4$$

We obtained an approximate calibration (to obtain x and y pixels to centimeters conversion factors) by solving two equations in two unknowns. The first equation is

$$a_x^2(dtRxd4 - dtLxd4)^2 + a_y^2(dtRyd4 - dtLyd4)^2 = 151.13^2 \tag{1}$$

where a_x is the horizontal pixels to cm conversion factor, a_y is the vertical pixels to cm conversion factor, and 151.13 cm (59.5 inches) is the nominal length of board between the left and right support points.

To obtain the second equation we worked with two additional points, $(dtMxd4, dtMyd4)$ and $(dbMxd4, dbMyd4)$. ("M" indicates "middle".) $dtMxd4$ and $dtMyd4$ are given by

$$dtMxd4 = (dtLxd4 + dtRxd4)/2.0$$

and

$$dtMyd4 = (dtLyd4 + dtRyd4)/2.0$$

$(dbMxd4, dbMyd4)$ is calculated as the intersection point between the bottom edge line, and the line that contains the top "midpoint", $(dtMxd4, dtMyd4)$, and is perpendicular to the top edge line.

If the top edge line is horizontal (if $dtLyd4 = dtRyd4$), the perpendicular bisector of the top line is just the line given by $x = dtMxd4$ so $dbMxd4 = dtMxd4$, and we solve for $dbMyd4$ by

$$dbMyd4 = a_{\text{bott}} + b_{\text{bott}} \times dbMxd4$$

If the top edge line is not horizontal, the slope of the perpendicular bisector of the top line segment is

$$b_{\text{perp}} = -1.0/b_{\text{top}}$$

and the intercept of the perpendicular bisector of the top line segment is given by

$$a_{\text{perp}} = dtMyd4 - b_{\text{perp}} \times dtMxd4$$

$(dbMxd4, dbMyd4)$ is then calculated as the intersection of the perpendicular bisector line and the bottom line. From

$$a_{\text{bott}} + b_{\text{bott}} \times x_{\text{inter}} = a_{\text{perp}} + b_{\text{perp}} \times x_{\text{inter}}$$

we obtain

$$dbMxd4 = x_{\text{inter}} = (a_{\text{bott}} - a_{\text{perp}})/(b_{\text{perp}} - b_{\text{bott}})$$

and

$$dbMyd4 = y_{\text{inter}} = a_{\text{bott}} + b_{\text{bott}} \times x_{\text{inter}}$$

The second calibration equation is

$$a_x^2(dtMxd4 - dbMxd4)^2 + a_y^2(dtMyd4 - dbMyd4)^2 = W^2 \tag{2}$$

where W is the average of the measured width (in centimeters) of the board at the left and right load points.

Equations 1 and 2 yield

$$\mathbf{As} = \mathbf{t} \tag{3}$$

where

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

$$\begin{aligned} a_{11} &= (dtRxd4 - dtLxd4)^2 \\ a_{12} &= (dtRyd4 - dtLyd4)^2 \\ a_{21} &= (dtMxd4 - dbMxd4)^2 \\ a_{22} &= (dtMyd4 - dbMyd4)^2 \end{aligned}$$

$$\mathbf{s} = \begin{pmatrix} a_x^2 \\ a_y^2 \end{pmatrix}$$

and

$$\mathbf{t} = \begin{pmatrix} 151.13^2 \\ W^2 \end{pmatrix}$$

Equation (3) is solved by inverting $\mathbf{A}$ to obtain

$$\mathbf{s} = \mathbf{A}^{-1}\mathbf{t}$$

Taking square roots then yields the x pixels to centimeters factor, a_x, and the y pixels to centimeters factor, a_y. (H horizontal pixels correponds to $a_x \times H$ centimeters, and V vertical pixels corresponds to $a_y \times V$ centimeters.)

5 Ellipse Fitting "Math"

As noted in Section 3.3, to obtain an elliptical fit to a knot, a user follows these steps:

1. Clicks on the left and right endpoints of the major axis of the ellipse.
2. Clicks on the lower and upper endpoints of the minor axis of the ellipse.
3. Clicks on four other other points "on" the ellipse perimeter.
4. Clicks on the ellipse submit button.

This yields points $(x(1), y(1)), \ldots, (x(8), y(8))$.

The program then obtains estimated values of the the elliptical parameters a, b, θ, x_c, y_c (half-length of the major axis, half-length of the minor axis, rotation of the ellipse, and center point coordinates of the ellipse) by finding the ellipse that minimizes the sum of the squared distances of the eight points from the ellipse.

This is done as follows:

For $\omega \in (-\pi, \pi)$, the points on a fixed a, b, θ, x_c, y_c ellipse are obtained by performing the transformation that takes points on the unit circle, $C(\mathbf{0}; 1)$, to points on the ellipse. The transformation can be broken down into a series of four subtransformations:

1. For $\omega \in (-\pi, \pi)$, $x_1 = \cos(\omega)$ and $y_1 = \sin(\omega)$.
2. Scale transformation: $x_2 = a \times x_1$ and $y_2 = b \times y_1$
3. Rotation transformation: $x_3 = x_2 \cos(\theta) - y_2 \sin(\theta)$ and $y_3 = x_2 \sin(\theta) + y_2 \cos(\theta)$
4. Translation transformation: $x_4 = x_3 + x_c$ and $y_4 = y_3 + y_c$

These four subtransformations take $(-\pi, \pi)$ 1 to 1 onto the a, b, θ, x_c, y_c ellipse.

Let $p(a, b, \theta, x_c, y_c; \omega)$ denote the point on the ellipse that is obtained by applying the four subtransformations to ω.

To obtain an ellipse that is a good fit to the eight clicks that we use to characterize a knot, for each of the $(x(i), y(i))$ points $(i = 1, \ldots, 8)$ identified by a user's clicks, we use our Java version of the fmin routine to find the ω's in $(-\pi, -\pi/2)$, $(-\pi/2, 0)$, $(0, \pi/2)$, and $(\pi/2, \pi)$ that yield minimum distances between $p(a, b, \theta, x_c, y_c; \omega)$ and $(x(i), y(i))$. We then take the minimum distance between the a, b, θ, x_c, y_c ellipse and point $(x(i), y(i))$ to be the smallest of these four values, and denote the sum of the squared distances between the eight points and the a, b, θ, x_c, y_c ellipse by S^2. We then use our Java version of the Uncmin program to find the a, b, θ, x_c, y_c values that minimize S^2. (Thus, our optimization requires minimizations within a minimization. It can be compute intensive.)

6 Filling and Using the iflag_fill Matrix

To fit the MOR models 2-5 described in sections 5 and 6 of Verrill et al. (2021), we need to be able to calculate the areas of the intersections of knots and certain rectangular regions on a board. We also need to be able to calculate the fraction of a "vertical" line segment (one that is "perpendicular to a lengthwise board edge) that lies within a knot.

It is possible to calculate analytically the area of the intersection of an ellipse and a rectangular region, and it is possible to calculate analytically the length of the intersection of an ellipse and a line. However, when we are interested in a board that has been subdivided into multiple rectangular regions, these calculations become complicated.

Thus, to estimate the areas of the intersections between ellipses and rectangular regions, and the lengths of intersections between ellipses and "vertical" lines, we take an approximate "grid approach." That is, we divide a board's image between the "vertical" green lines above its support points into a 200 by 1800 grid of cells.

The grid has an upper line between the $(dtLxd4, dtLyd4)$ and $(dtRxd4, dtRyd4)$ points described in Section 4, and a lower line, parallel to the upper line, that goes through the point $(dbMxd4, dbMyd4)$ described in Section 4.

For each of the 200×1800 rectangular cells of a board, we determine whether its center lies within a knot. If so, the cell's iflag_fill value is set to 1. Otherwise, we set its iflag_fill value to 0. This work is done in the iflag_fill method.

We determine whether a point (x, y) lies within a knot parameterized by (a, b, θ, x_c, y_c) by checking whether

$$(\cos(\theta)(x - x_c) + \sin(\theta)(y - y_c))^2/a^2 + (-\sin(\theta)(x - x_c) + \cos(\theta)(y - y_c))^2/b^2 \leq 1$$

For Models 2 and 3, the iflag_fill method is run within Knots_2x4 upon a click of the next board button. Then the iflag_sums method is run within Knots_2x4 to calculate (from the iflag_fill grid of 0's and 1's) the 24 $1 - p_{ij}$ values described in section 5.2 of Verrill et al. (2021). These 24 values are printed for each board in the summer_knot_F.res (for the front face images) and summer_knot_B.res (for the back face images) results files produced by runs of Knots_2x4.java (see Section 3.1) and used in the Nreg2.java and Nreg3.java nonlinear least squares programs that obtain the parameter estimates for Models 2 and 3.

Models 4 and 5 make use of the full 200×1800 grid of 0's and 1's for each of the 590 boards. These arrays are not stored in data files but are generated within Nreg4.java and Nreg5.java (the nonlinear least squares programs that obtain the parameter estimates for Models 4 and 5) via calls to Nreg6_aux.iflag_fill. For each board they are then converted (via a call to Nreg6_aux.frac4_fill) to the 4x1800 f_{ij} values used in the calculation of the predicted MOR (see equations 10 and 12 in Verrill et al. 2021). For example, f_{11} is the fraction of 1's among the upper 50 (out of 200) values in the leftmost column among the 1800. (It is an estimate of the fraction of the length of the top fourth of the leftmost of the 1800 lines that lies in a knot.)

7 Summary

Verrill et al. (2021) predicted the MORs of boards from measurements of MOE, SG, SOG, and knot sizes, locations, and orientations. To do so they developed a Java lumber knot image analysis program and Java nonlinear least squares fitting routines. These programs are discussed in this report. These programs are research tools rather than production tools. To make use of them, one would need the services of a Java programmer. The lead author is also willing to provide some software support.

8 References

Anderson, G.C., Owens, F.C., Verrill, S.P., Ross, R.J., and Shmulsky, R. (2021), "Fitting Statistical Distribution Models to MOE and MOR in Mill-Run Spruce and Red Pine Lumber Populations," *Wood and Fiber Science*, **53**(1), pp. 17-26.

Owens, F.C, Verrill, S.P., Shmulsky, R., and Kretschmann, D.E. (2018), "Distributions of MOE and MOR in a Full Lumber Population," *Wood and Fiber Science*, **50**(3), pp. 265-279.

Owens, F.C, Verrill, S.P., Shmulsky, R., and Ross, R.J. (2019), "Distributions of Modulus of Elasticity and Modulus of Rupture in Four Mill-Run Lumber Populations," *Wood and Fiber Science*, **51**(2), pp. 183-192.

Owens, F.C, Verrill, S.P., Shmulsky, R., and Ross, R.J. (2020), "Distributions of MOE and MOR in Eight Mill-Run Lumber Populations (Four Mills at Two Times)," *Wood and Fiber Science*, **52**(2), pp. 165-177.

Schnabel, R.B., Koontz, J.E., and Weiss, B.E. (1982), *A Modular System of Algorithms for Unconstrained Minimization*, Report CU-CS-240-82, Comp. Sci. Dept., University of Colorado at Boulder.

Verrill, S.P., Evans, J.W., Kretschmann, D.E., and Hatfield, C.A. (2012). "Asymptotically Efficient Estimation of a Bivariate Gaussian–Weibull Distribution and an Introduction to the Pseudo-truncated Weibull." Research Paper FPL-RP-666. Madison, WI: U.S. Department of Agriculture, Forest Service, Forest Products Laboratory. 76 pages.

Verrill, S.P., Evans, J.W., Kretschmann, D.E., and Hatfield, C.A. (2013). "An Evaluation of a Proposed Revision of the ASTM D 1990 Grouping Procedure." Research Paper FPL-RP-671. Madison, WI: U.S. Department of Agriculture, Forest Service, Forest Products Laboratory. 34 pages.

Verrill, S.P., Evans, J.W., Kretschmann, D.E., and Hatfield, C.A. (2014), "Reliability Implications in Wood Systems of a Bivariate Gaussian–Weibull Distribution and the Associated Univariate Pseudo-truncated Weibull," *ASTM Journal of Testing and Evaluation*, **42**, Number 2, pp. 412-419.

Verrill, S.P., Evans, J.W., Kretschmann, D.E., and Hatfield, C.A. (2015), "Asymptotically Efficient Estimation of a Bivariate Gaussian–Weibull Distribution and an Introduction to the Pseudo-truncated Weibull," *Communications in Statistics – Theory and Methods*, **44**, pp. 2957-2975.

Verrill, S.P., Owens, F.C., Kretschmann, D.E., and Shmulsky, R. (2017). "Statistical Models for the Distribution of Modulus of Elasticity and Modulus of Rupture in Lumber with Implications for Reliability Calculations." Research Paper FPL-RP-692. Madison, WI: U.S. Department of Agriculture, Forest Service, Forest Products Laboratory. 51 pages.

Verrill, S.P., Owens, F.C., Kretschmann, D.E., and Shmulsky, R. (2018). "A Fit of a Mixture of Bivariate Normals to Lumber Stiffness–Strength Data." Research Paper FPL-RP-696. Madison, WI: U.S. Department of Agriculture, Forest Service, Forest Products Laboratory. 44 pages.

Verrill, S.P., Owens, F.C., Kretschmann, D.E., Shmulsky, R., and Brown, L.S. (2019). "Visual and MSR Grades of Lumber Are Not 2-Parameter Weibulls and Why It Matters (With a Discussion of Censored Data Fitting)." Research Paper FPL-RP-703. Madison, WI: U.S. Department of Agriculture, Forest Service, Forest Products Laboratory. 40 pages.

Verrill, S.P., Owens, F.C., Kretschmann, D.E., Shmulsky, R., and Brown, L.S. (2020a). "Visual and MSR Grades of Lumber Are Not 2-Parameter Weibulls and Why This May Matter," *ASTM Journal of Testing and Evaluation*, **48**, Number 5, pp. 3946-3962.

Verrill, S.P., Owens, F.C., Arvanitis, M.A., Kretschmann, D.E., Shmulsky, R., Ross, R.J., and Lebow, P. (2020b). "Estimated Probability of Breakage of Lumber of a Fixed 'Grade' Can Vary Greatly from Mill to Mill and Time to Time." Research Paper FPL-RP-705. Madison, WI: U.S. Department of Agriculture, Forest Service, Forest Products Laboratory. 47 pages.

Verrill, S.P., Owens, F.C., Shmulsky, R., and Ross, R.J. (2021). "Improved Models for Predicting the Modulus of Rupture of Lumber under Third Point Loading." Research Paper FPL-RP-712. Madison, WI: U.S. Department of Agriculture, Forest Service, Forest Products Laboratory. `https://www1.fpl.fs.fed.us/knots_2x4.html`

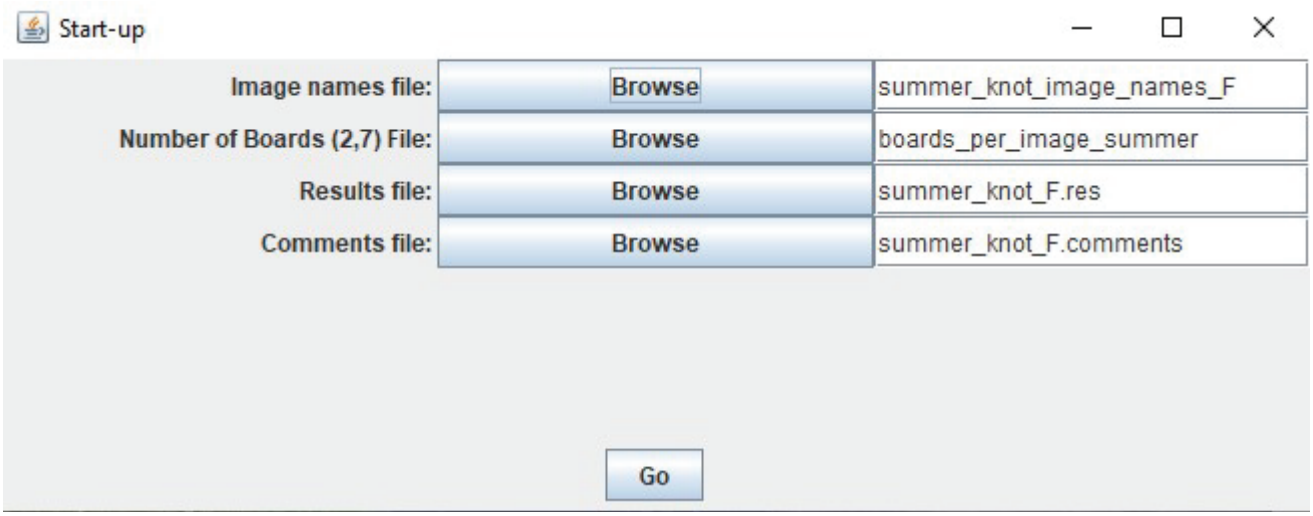

Figure 1: Knots_2x4 start-up panel.

Figure 2: Knots_2x4 main panel.

Figure 3: Calibration of a board.

Figure 4: Ellipse fitting.

Figure 5: Magnification and comments.