

A COMPUTER-ASSISTED
TREE-RING CHRONOLOGY COMPOSITION SYSTEM

by

James Blaine Engle

A Thesis Submitted to the Faculty of the
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE

In the Graduate College

THE UNIVERSITY OF ARIZONA

2000

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgement the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: _____

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

Robert A. Schowengerdt
Professor of ELECTRICAL AND
COMPUTER ENGINEERING

Date

ACKNOWLEDGEMENTS

I would like to thank Dr. Robert Schowengerdt and Dr. Malcolm Hughes for their most kind assistance, advice, and support in furthering my education. I would also like to thank Jim Burns, Gary Funkhouser, Giribalan Gopalan, Dr. Richard Holmes, Dr. Paul Sheppard, and Dr. Ramzi Touchan for their help in testing the software and providing valuable advice as to how to make it more useful and intuitive to dendrochronologists. I am especially grateful to Dr. Martin Munro for his extraordinary patience in teaching this engineering student about dendrochronology and to Steve Conner for getting me started with the TREES program. I am additionally grateful to Dr. Larry C. Schooley, the University of Arizona ECE Department, and the International Foundation for Telemetering for their support of my education.

This work was supported by the National Science Foundation, Grant SBR9601867, and the University of Arizona, Office of the Vice President for Research.

DEDICATION

I dedicate this thesis to my lovely wife Diann, without whose love, support, and friendship, this thesis would not have been possible.

TABLE OF CONTENTS

LIST OF TABLES	7
LIST OF FIGURES	8
ABSTRACT	10
CHAPTER 1 INTRODUCTION	11
1.1 Objective and Problem Statement	11
1.2 Dendrochronology.....	11
1.3 The Douglas Method of Tree Ring Cross-Dating.....	15
1.4 The TREES System	19
1.5 Overview of Thesis.....	21
CHAPTER 2 SYSTEM REQUIREMENTS AND SOFTWARE DESIGN	25
2.1 System Requirements.....	25
2.2 Software Engineering.....	27
2.2.1 Software Design Approach.....	27
2.2.2 Graphical User Interface Environment	29
2.2.3 Algorithm Implementation.....	31
2.3 Data Import and Export Capability.....	32
2.4 Basic Graphic User Interface Design	33
CHAPTER 3 TREE-RING GROWTH PATTERN MATCHING.....	37
3.1 Tree-Ring Width Series Standardization	37
3.2 Skeleton Plotting	41
3.3 Correlation Pattern Matching.....	45
3.4 User Interaction — Correlation Tool.....	53
CHAPTER 4 COMPOSING A MASTER CHRONOLOGY	58
4.1 Master Chronology Composition.....	58
4.2 Master Chronology Data Structure.....	61
4.3 Hierarchical Composition of Master Chronology Components.....	63
4.4 Importing and Exporting Data.....	68
4.5 User Interaction — Composition Tool.....	70

TABLE OF CONTENTS — CONTINUED

CHAPTER 5	MASTER CHRONOLOGY QUALITY CONTROL	80
5.1	COFECHA Method Overview	80
5.2	COFECHA Method Details	83
5.3	Master Chronology Quality Control Check Implementation	87
5.3.1	Sample Component Date Checking.....	88
5.3.2	Sample Component Statistical Outliers	90
5.3.3	Sample Component Overall Correlation Fit.....	90
5.3.4	Sample Component Segment Correlation Fit.....	92
5.4	User Interaction — Check Tool.....	93
CHAPTER 6	SYSTEM INTEGRATION AND TESTING.....	102
6.1	Module Testing and Integration.....	102
6.2	System Validation and Verification.....	105
6.3	User Help and Assistance.....	107
6.4	Software Configuration Control.....	109
CHAPTER 7	SUMMARY AND CONCLUSIONS	111
7.1	Summary	111
7.2	Conclusions.....	114
7.3	Recommendations for Future Improvements.....	115
APPENDIX A	ITRDB Data File Formats	117
APPENDIX B	TREES Data File Structure and Format.....	120
APPENDIX C	CROSSDATE Session File Structure and Format	123
APPENDIX D	CROSSDATE Master Chronology File Structure and Format.....	124
APPENDIX E	Example COFECHA Output.....	126
APPENDIX F	Mathematical Basis for Smoothing Spline Algorithm	133
REFERENCES	136

LIST OF TABLES

Table 4-1, Example master chronology sample location area hierarchy	60
Table 7-1, Internet locations for TREES project references.....	114

LIST OF ILLUSTRATIONS

Figure 1-1, Creating a chronology using living and dead wood.....	13
Figure 1-2, Earlywood to latewood transition for one tree-ring.....	13
Figure 1-3, Douglass method skeleton plot example.....	16
Figure 1-4, TREES system components	19
Figure 1-5, Example TREES display graphic user interface.....	22
Figure 2-1, Iterative software development model.....	27
Figure 2-2, CROSSDATE application main graphic user interface.....	35
Figure 2-3, CROSSDATE application main menu.....	36
Figure 2-4, CROSSDATE application File menu.....	36
Figure 2-5, CROSSDATE application time series controls	36
Figure 2-6, CROSSDATE application sample controls.....	36
Figure 3-1, Example of cubic smoothing spline ringwidth standardizations	40
Figure 3-2, Intervals for cut-off quantization.....	43
Figure 3-3, Skeleton plot line drawing process.....	44
Figure 3-4, Skeleton plot crossdating example	45
Figure 3-5, Overlapping match positions between two series	48
Figure 3-6, Calculation of the cross-covariance term, S_{xy}	53
Figure 3-7, Correlation tool menu.....	54
Figure 3-8, Correlation tool sample selection dialog.....	54
Figure 3-9, Correlation tool graph panel.....	54
Figure 3-10, Correlation tool menus and options.....	55
Figure 4-1, Object data structures and inheritance	63
Figure 4-2, Depth-First Search algorithm.....	66
Figure 4-3, Example Depth-First Traversal of a master chronology hierarchy	67
Figure 4-4, CROSSDATE sample open menu	69

LIST OF ILLUSTRATIONS - CONTINUED

Figure 4-5, CROSSDATE Compose menu	75
Figure 4-6, CROSSDATE Compose sample selection dialog box.....	75
Figure 4-7, CROSSDATE set start or end year dialog and entry box.....	75
Figure 4-8, CROSSDATE master chronology header information dialog box.....	76
Figure 4-9, CROSSDATE master chronology file save dialog box.....	76
Figure 4-10, CROSSDATE master chronology component display request dialog box...	76
Figure 4-11, CROSSDATE master chronology component display screen.....	77
Figure 4-12, CROSSDATE master chronology component display screen menus	78
Figure 4-13, CROSSDATE master chronology component list report.....	79
Figure 5-1, Example series segmentation for program COFECHA.....	85
Figure 5-2, CROSSDATE master chronology QC check process	97
Figure 5-3, CROSSDATE master chronology QC check display screen.....	98
Figure 5-4, CROSSDATE sample component QC check panel	99
Figure 5-5, CROSSDATE master chronology QC check menu options.....	100
Figure 5-6, CROSSDATE master chronology QC check report example.....	101
Figure 6-1, CROSSDATE software module block diagram	103
Figure 6-2, CROSSDATE user help interaction sequence.....	108

ABSTRACT

The design and implementation of a computer-assisted tree-ring chronology composition system for dendrochronology is described in this thesis. Methods for tree-ring growth sequence pattern matching, hierarchical composition of master chronologies, and analytical quality control are discussed in detail and systematically implemented in the CROSSDATE program in a highly visual, graphic environment. The objective of this work is to provide dendrochronologists with a robust set of tools for comparing the relative growth patterns of tree-ring samples for purposes of dating the samples and composing new master chronologies from individual tree-ring samples and other master chronologies. This system is complementary to the TREES program, a computer-vision based tree-ring identification and measurement system developed at the University of Arizona.

CHAPTER 1 — INTRODUCTION

1.1 Objective and Problem Statement

This thesis describes the software engineering and human factors issues involved in the implementation of a computer-assisted tree-ring chronology composition system. The objective of this work is the creation of a software application tool which will aid dendrochronologists in (1) determining the dating of an individual tree-ring sample and (2) composing a master time series growth rate chronology from individual samples.

This thesis is the result of cooperative efforts between the Laboratory of Tree-Ring Research (LTRR) and the Digital Image Analysis Laboratory (DIAL) in the Electrical and Computer Engineering Department at the University of Arizona. The research was funded by the National Science Foundation (Grant SBR9601867) and the University of Arizona, Office of the Vice President for Research.

1.2 Dendrochronology

Dendrochronology is the science of tree-ring dating. According to [20], the term *dendro* is from *dendron*, the Greek word for tree, and *chronology* means the assignment of dates to particular events in a time series. When a tree is cut down, its cross section shows numerous concentric rings about its center. It is widely known that many trees grow a new ring for each year of life. It may also be shown that the width of a specific

tree-ring is closely related to the climate the tree experienced during the period of growth. In general, moisture and warmth promote growth, and lack of moisture and warmth inhibit growth. Of course, these are not the only factors that affect tree growth, but over long periods of time they are the most significant in many parts of the world. The fundamental scientific basis of dendrochronology, established by Andrew E. Douglass in the early 20th century, is that climate affects all trees in a given area consistently. This means that the pattern of *relative* ring width variation between different trees from close locations should be very similar for a given time period, even though their *absolute* ring widths may be quite different. This could be, for example, due to local soil conditions at a particular tree. This makes it possible to precisely date tree-ring samples through ring width variation pattern matching. Figure 1-1 shows how a chronology may be established using living and dead wood.

A distinct pattern of light to dark color variation is noticed in tree-ring samples, especially from coniferes. Dendrochronologists call the lighter colored area of a ring *earlywood* and the darker colored area *latewood*. At the cellular level, earlywood consists of large, thin-walled cells and latewood consists of small, thick-walled cells. As their names suggest, earlywood grows at the beginning of the growing season and latewood grows toward the end of a growing season. The beginning of the growing season roughly corresponds to the spring and summer seasons and the end of the growing season roughly corresponds to late summer and early fall. In general spring and summer tend to be

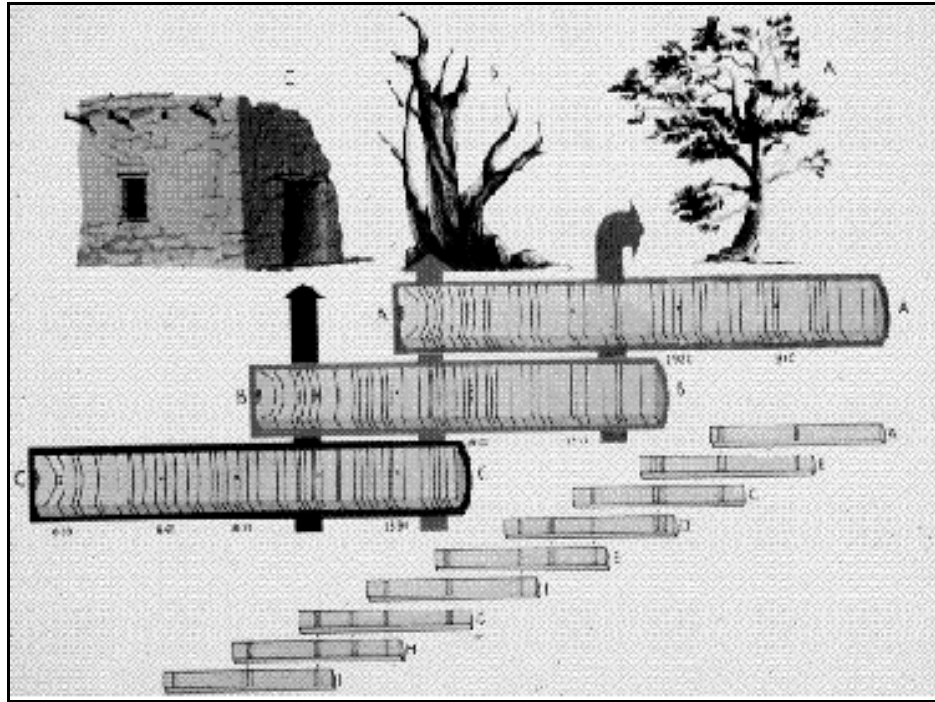


Figure 1-1 — Creating a chronology using living and dead wood
(Courtesy of the Laboratory of Tree-Ring Research, University of Arizona, [46, pg. 4])

warmer and have more useful moisture than fall and winter, hence, the difference in tree growth rates. Figure 1-2 shows a typical earlywood to latewood tree-ring transition.

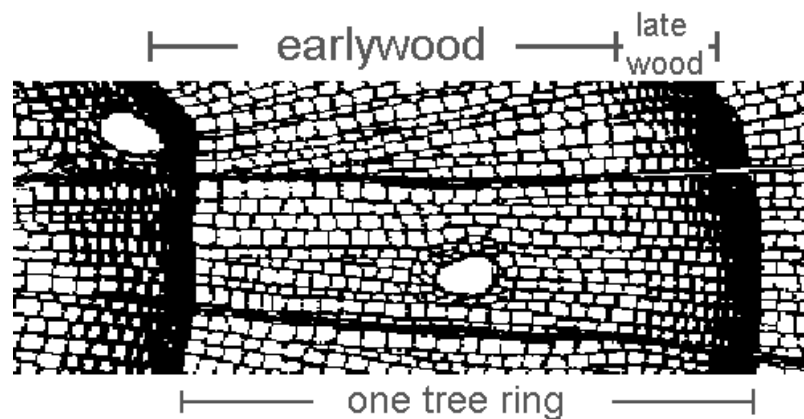


Figure 1-2 — Earlywood to latewood transition for one tree-ring
(Courtesy of the Laboratory of Tree-Ring Research, University of Arizona,
<http://www.ltrr.arizona.edu/dendrochronology.html>)

Andrew E. Douglass was an astronomer interested in sunspot events and a possible causal relationship between periodic sunspot activity and temporal variations in climate on earth. In order to prove his theory, Professor Douglass needed long term climatic and weather data. In Northern Arizona in 1901, Douglass noticed the similarities in tree-ring width variation patterns among several recently cut logs. He specifically noticed a series of very small rings located twenty-one rings inside the bark. He also examined a stump, which had been cut at an unknown time in the past. He noticed the same pattern of small rings on the stump's cross-section, but they were located eleven rings inside the bark. He concluded that the tree must have been cut ten years prior. Douglass knew that tree-rings grew as a function of climate; now he knew how to determine what year they represented. This was effectively the first recorded instance of tree-ring crossdating.

The practical significance of dendrochronology must not be underestimated. Many fields of study require long-term climate variation data, which can be readily provided by studying tree-ring width variations. There are many areas of the world where water is a particularly valuable resource. Knowing what rainfall to expect over the long term has a tremendous impact on agriculture and agricultural planning.

Tree-ring width variations additionally serve as records of past conditions. Knowing the frequency and severity of past climatic conditions is useful in anticipating the occurrence of future events. There are other fields of study, for example archaeology

and anthropology, where simply knowing how old something is provides significant insight as to when certain events occurred.

1.3 The Douglass Method of Tree Ring Crossdating

Four requirements for performing crossdating on tree-ring samples have been specified by Stokes and Smiley in [45]:

1. The trees used for crossdating must grow one new ring for each year of life. This is referred to as an *annual ring*. There are species that grow more than one ring in a season. These cannot be considered reliable for dating purposes.
2. There must be one environmental factor that is dominant in limiting the annual growth of the tree. In the desert southwest, this might be rainfall. In Alaska, it might be temperature.
3. The magnitude of the growth-limiting factor must vary from year to year, and the resulting annual tree-rings should reflect this variation in their relative growth patterns.
4. The growth-limiting factor should be uniformly effective over a large geographical area.

Assuming that the above four conditions are true for the tree-ring samples, it is then possible to perform crossdating on the samples. A technique known as Skeleton Plotting was developed by Dr. A. E. Douglass and remains a popular method for

comparing and dating tree-ring samples today. Figure 1-3 shows a skeleton plot representation of a tree ring sample.

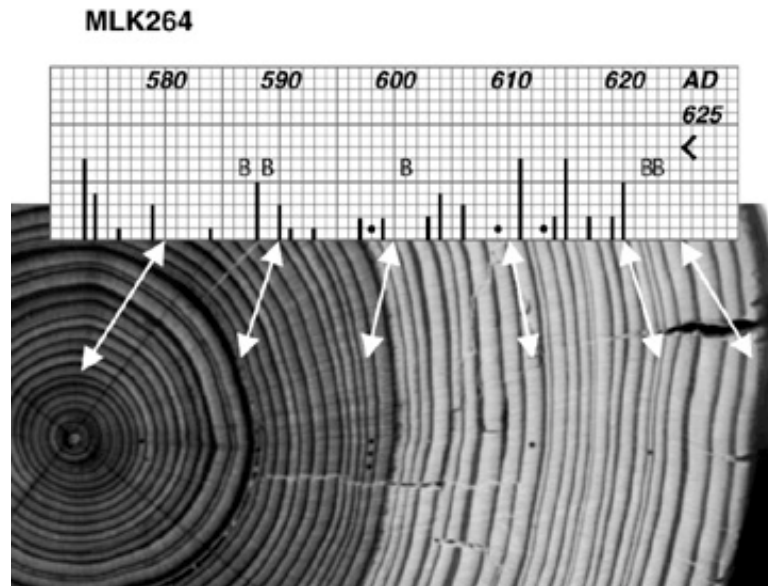


Figure 1-3 — Douglass method skeleton plot example
(Courtesy of the Laboratory of Tree-Ring Research, University of Arizona, [45])

Skeleton plotting provides the means for characterizing and comparing tree-ring samples primarily using their width measurements. A skeleton plot is drawn on graph paper where each vertical grid line corresponds to one annual tree-ring. This allows for alignment and comparison with other skeleton plots, since the distance between each vertical grid line is constant.

Tree-rings are plotted beginning with the innermost ring on the left, progressing to the outermost ring on the right. Before absolute dates are known, zero is used for the beginning ring, and each decade is marked at the top of the plot. Skeleton plotting is primarily concerned with narrow rings; particularly those rings that are narrow compared

to their immediate neighbors. Narrow rings are marked on the skeleton plot with a bold dark line. The narrower the ring, the longer the line. The narrowest rings are arbitrarily represented with bold lines that are 2 major grids in height. In Figure 1-3, the rings at years 573, 611, and 615 are the narrowest rings on the sample. Wide rings are marked with a B on the plot. In Figure 1-3, the rings at years 587, 589, 601, 622, and 623 are the widest (compared to their neighbors) on the sample. If a ring is narrow, but not so narrow as to be represented by a line on the skeleton plot, a dot is sometimes used to represent it. In Figure 1-3, the rings at years 598, 608, and 613 are marked with dots. Rings with average widths receive no special marking on the skeleton plot. It is important to note that in practice most skeleton plots are done by visual observation and the scientist's judgement, and do not use quantitative measurements. However, the technique has proven robust and reliable over many years of use in many laboratories around the world.

Skeleton plotting captures the extremes in variations of the sample's ring widths. These may be thought of as the sample's *features*, which are likely due to significant variations in the dominant environmental factor described above. Since samples from a geographic region exhibit similar variations at a given time period, it is possible to match samples in time based on their common features. The description of skeleton plotting given above is greatly oversimplified. This thesis will explore algorithms for skeleton plotting in more detail in Chapter 3.

The skeleton plot is not necessarily a 100% accurate proxy for the tree-ring sample. Many physical anomalies may occur that will invalidate the procedure. For these reasons, the Douglass method of crossdating requires the original wood sample to remain available throughout the dating process for the dendrochronologist to reference in resolving problems. Examples of physical anomalies include micro rings and false rings. A micro ring is so small that it is likely to have been missed when the ring widths were originally measured. If a micro ring is missed, the skeleton plot will obviously not include it, and the assigned years of the following rings will be one year low. It is possible that some unusual climate conditions may cause a ring to darken in the middle of a particular year. This may be picked up as a false ring during ring width measurement. A false ring has the opposite effect on the skeleton plot as a micro ring with the assigned years of the following rings being one year high. Dendrochronologists must therefore continuously review and edit skeleton plots during the crossdating process.

After individual samples have been plotted, several plots may be compared at one time. The position of the plots may be adjusted so that areas of similar ring patterns are aligned. When this is done correctly, the rings represented by each vertical line may be assumed to have occurred during the same year. If one of the samples contains actual date information, it may then be used to determine dates for the other samples. A composite *master chronology* may then be prepared by plotting the average of the sample line lengths for each year and transferring the dates. The Master Chronology may be used for

dating more samples, extending other Master Chronologies, or comparing climatic variations between geographic regions.

1.4 The TREES System

The TREES program is a computer vision based tree-ring analysis and dating system developed at the University of Arizona [6,7,34]. Figure 1-4 shows the basic components of the TREES system.

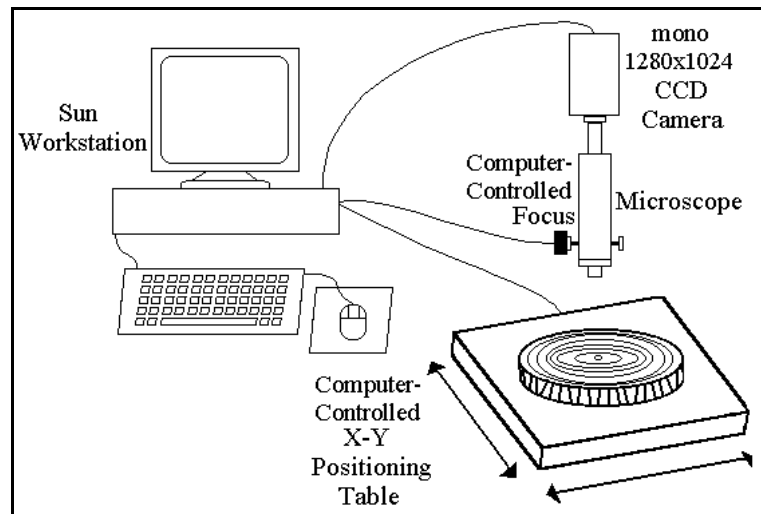


Figure 1-4 — TREES system components [7]

The system performs the following tree-ring sample image acquisition and analysis tasks:

Tree-Ring Sample Image Acquisition, including:

- Frame Capture (Capture)
- Mosaic Creation (Mosaic)

Tree-Ring Sample Image Analysis, including:

- Tree-Ring Boundary Edge Detection (Find Rings)
- Tree-Ring Width Measurement (Measure Rings)
- Skeleton Plot Generation (Skeleton Plot)

The mission of the TREES system is to provide dendrochronologists with a tool for rapidly characterizing and dating tree-ring samples, while maintaining the ability to trace results back to the original wood sample.

The process begins with a prepared sample mounted on the X-Y positioning table. Under software control, the user specifies the area of the sample to be imaged. The system automatically acquires images of the required area. The individual images are balanced for variations in illumination levels, registered separately, and assembled as a mosaic to create a single image. An edge detection algorithm is used to determine the location of the boundaries of the rings in the mosaic image. The pixel coordinates of the boundaries are highlighted for the user on the display.

Each ring is measured along its full length (in the image) using the average of orthogonal vector lengths. This measurement technique is significantly more robust than the typical manual single point measurement. As a final step, the ring width measurements are used to generate a skeleton plot, which is presented to the user on the display. See Figure 1-5 for an example of the TREES graphic user interface.

The TREES system additionally provides the dendrochronologist with many tools

for analyzing and editing the sample. Some of the major features include:

1. The ability to add rings to- and remove rings from- the sample with corresponding changes displayed immediately on the skeleton plot.
2. The ability to select a region of interest at multiple resolutions on the main sample image.
3. The ability to zoom in on a particular area of the sample.
4. The ability to select a point on the image and have the system display live video of the actual sample when the sample is still mounted on the X-Y positioning table.

According to Conner [6], TREES is not the first system to apply image analysis to the field of dendrochronology. However, to the knowledge of the TREES project design team, it is the first image analysis system to apply the Douglass method of tree ring dating.

1.5 Overview of Thesis

This thesis is a continuation of the work started by Conner et al. [6,7,34] on TREES, a computer vision based tree-ring analysis and dating system. Conner recommended that a future enhancement should be the integration of pattern matching techniques to aid the dendrochronologist in crossdating tree-ring samples from different

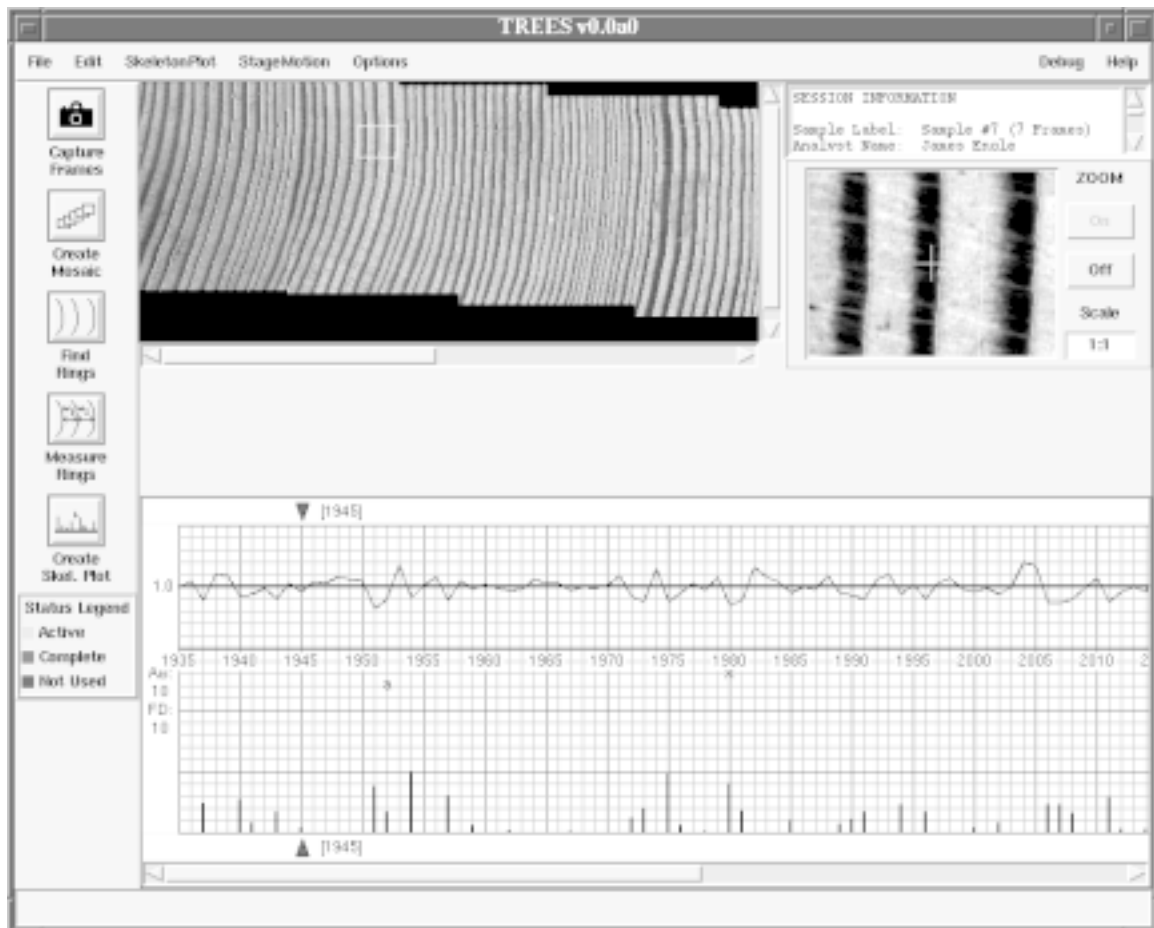


Figure 1-5 — Example TREES display graphic user interface

trees or wood samples. Philosophically, this work is not an attempt to produce a fully automated solution, but rather a computer tool that assists tree-ring scientists in their work. The fundamental difference with previous work is that now the focus is on comparing multiple tree-ring samples for growth pattern matching, rather than measuring an individual sample.

The thesis begins with an introduction to the science of dendrochronology including a discussion of techniques for comparing and dating tree-ring samples. The Laboratory of Tree-Ring Research (LTRR) at the University of Arizona is a leading

source for advances in this field. A description of the TREES computer program for tree-ring sample image acquisition and analysis is also presented.

In the development of the CROSSDATE computer program described in this thesis, strict adherence to software engineering principles has been maintained. In Chapter 2, system requirements, the plan for system design, and the basic user interface are described in detail. Integration with previous work and the ability to return to the original tree-ring sample are particularly crucial requirements.

The core capabilities of the CROSSDATE program are developed and discussed in Chapters 3, 4, and 5. Tree-ring sample measurements are prepared for crossdating by first detrending and reducing absolute tree-ring measurements to a common scale of relative index values. This process is known as *standardization*. In order to determine the dating for an individual sample, it must be compared to another sample with known dating. Comparing an unknown sample with a known sample involves determining the point where their relative growth rates (i.e., relative ring width variations) match. Historically, this has been achieved primarily by visual examination, but the computer can be an especially valuable tool to calculate sample time series correlation for sample dating. The CROSSDATE program provides a correlation tool for determining the strength of correlation between two samples at each possible overlapping position in time.

After determining the dating of many samples from a particular site, dendrochronologists find it useful to compose master chronologies by averaging a number

of individual samples. Master chronologies are representative of the average relative variation of tree-ring growth for a particular site over a particular time period. They are useful in dating further samples from the same or nearby sites. They are also useful for comparing climatic variations between sites. The CROSSDATE program provides a composition tool for aggregating multiple samples into a master chronology. Master chronologies are created hierarchically, and the program provides the ability to trace back and review all samples used in composing a master chronology.

One technique for validating manual crossdating results is the COFECHA⁽¹⁾ computer program [26]. Program COFECHA is statistically based, but goes beyond simple correlation to check pattern matches between a tree-ring sample and a master chronology. COFECHA was created at a time when graphical user interfaces (GUIs) were not widely available. The CROSSDATE program provides a GUI implementation of the COFECHA program to check the validity of the dating of a sample.

The final topics of the thesis involve testing and validating the operation of the CROSSDATE program and ensuring proper system integration of the major modules. The thesis is concluded with a discussion of areas for further research. The contribution of the CROSSDATE computer program to dendrochronology is a highly visual graphic framework for performing tree-ring sample comparison and chronology composition activities as a companion to the TREES computer program.

⁽¹⁾ - From Spanish, Cofecha roughly translates to "cross-date".

CHAPTER 2 — SYSTEM REQUIREMENTS AND SOFTWARE DESIGN

2.1 System Requirements

The fundamental requirements of the CROSSDATE program include the following:

1. Ability to determine, manually and with computer assistance, the point where two tree ring samples match in time.
2. Ability to compose master chronologies from individual samples and other master chronologies.
3. Ability to determine and review the specific samples from which a master chronology is composed for purposes of returning to the original wood sample.
4. Ability to rigorously check a tentative chronology for correctness.
5. Easy to use, intuitive graphic user interface (GUI) which mimics the steps a tree-ring scientist follows when performing manual crossdating.
6. Ability to transfer data (Import/Export) in common formats to other tree-ring application programs.
7. Ability to operate on PC and UNIX workstation hardware platforms.
8. Ability to operate seamlessly with the TREES program.

From this list of requirements, several further requirements follow. First, in order to

operate seamlessly with the TREES program, the CROSSDATE program should use the same programming languages, namely Tcl/Tk, C, and C++. These environments will be discussed in detail shortly. Next, TREES is operational on Sun workstations running the Solaris UNIX operating system and on PCs running the Red-Hat™ LINUX version of the UNIX operating system. CROSSDATE must therefore also operate on these platforms under these operating systems. At some point in the future, TREES may be ported to the Microsoft Windows™ or Apple MacOS™ operating systems; CROSSDATE will also require similar ports at the same time.

The CROSSDATE program's fundamental requirements have mandated its creation as a separate application from TREES. The focus of the TREES program is the acquisition and analysis of a *single* tree-ring sample. The focus of the CROSSDATE program is the comparison of *multiple* tree-ring samples to create master chronologies. This implies that the CROSSDATE program should be separate from the TREES program, yet integrated with it. This has the further implication that original image data acquisition and measurement are performed in the TREES program and simply made available to the CROSSDATE program. In other words, CROSSDATE should not have the ability to modify image data or measurements.

The TREES and CROSSDATE programs require intensive user interaction. In order to perform their functions, each application requires a large area of the display screen. A single screen that performs both functions would likely be very confusing and

not user friendly. Even if both programs resided in a common application, the resulting user interface would be more effective using separate screens for each program. Since both applications are written for a windowing environment, the fact that they are separate applications is not much different than separate screens in the same application.

2.2 Software Engineering

2.2.1 Software Design Approach

An iterative development model using rapid prototyping was followed for the CROSSDATE application development. An iterative life cycle model does not attempt to start with a fully detailed specification of requirements. Instead, development begins by specifying and implementing parts of the software, which can then be reviewed in order to identify further requirements. This process is then repeated, producing a new version of the software for each cycle of the model.

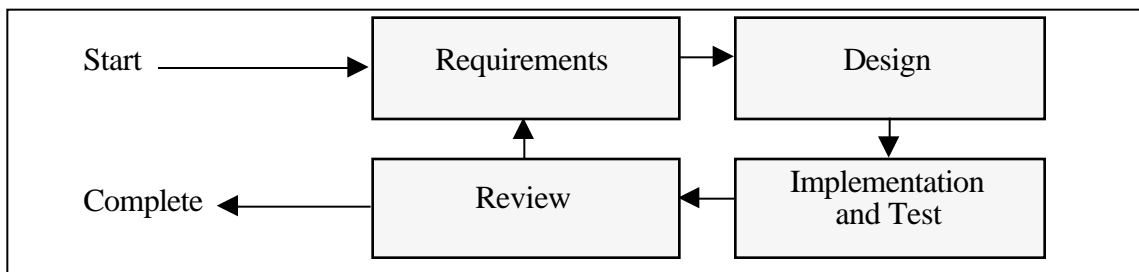


Figure 2-1 — Iterative software development model

An iterative life cycle model consists of repeating four phases in sequence, as

shown in Figure 2-1, for each portion of the software application. In this model, the requirements phase consists of defining (or refining) the requirements for a software component. For the CROSSDATE program, the software components would be the basic graphic user interface, the correlation tool, the composition tool, and the quality control check tool. Iteration eventually results in a complete and final specification of requirements. The design phase consists of developing a software solution that meets the requirements. This resulting design may be new, or an extension of an earlier design. The implementation and test phase consists of coding, integrating, and testing the software. The review phase consists of evaluating the software against its current requirements and proposing changes and additions as required.

Each software component is developed as a *cycle* of the model. For each cycle, a decision has to be made as to whether the software produced by the cycle will be discarded, or kept as a starting point for the next cycle (sometimes referred to as *incremental prototyping*). Eventually a point will be reached where the requirements are complete and the software can be delivered, or it becomes impossible to enhance the software as required, and a fresh start has to be made.

Using a mathematical analogy, the iterative lifecycle model is a successive approximation process. The success of the method depends on how rapidly it converges to a solution. Conversely, successive approximation may never find a solution. The iterations may oscillate around a feasible solution or even diverge. It is possible for the

number of iterations to become so large as to be unrealistic.

The key to successful use of an iterative software development lifecycle is rigorous validation of requirements, and verification (including testing) of each version of the software against those requirements within each cycle of the model. For the CROSSDATE application, the requirements listed in Section 2.1 are statements of major objectives only. They are not specific and detailed software design requirements. The major software engineering goal of this thesis is *to design an intuitive and effective user interaction process for each crossdating task*. This is accomplished through trial and error prototyping using input from dendrochronologists to converge on a solution.

2.2.2 Graphic User Interface Environment

The TREES application was developed as a Tcl/Tk graphic user interface (GUI) application. For compatibility, the CROSSDATE application is also developed as a Tcl/Tk GUI application. Tcl (Tool Command Language) is a scripting language with an interpreter that may be embedded in other applications. Tk is the associated graphic user interface toolkit. John Ousterhout created Tcl/Tk at the University of California at Berkeley [47], and Tcl/Tk is currently supported by Ajuba Solutions of Mountain View, California (<http://www.ajubasolutions.com>).

Tcl is a scripting language similar to other UNIX shell languages such as the c-shell (csh) and perl. Tcl contains sufficient programmability to allow the creation of complex

script applications that are virtually indistinguishable from custom applications in usage, with the possible exception of execution speed. Further, Tcl's interpreter is easily added to compiled applications. This means that an application that uses the Tcl interpreter has ready access to all of Tcl/Tk's functionality, and Tcl/Tk's functionality can be easily extended to suit specialized needs. Both of these capabilities are utilized in the creation of the TREES and CROSSDATE applications.

In support of rapid prototyping, using Tcl/Tk as the user interface programming language has four benefits:

1. Rapid turnaround, since there are no lengthy compilations.
2. Arbitrarily complex user interfaces are created with only a handful of commands.
3. The user interface is clearly separated from the rest of the application. Core application functionality may be developed independently of the user interface and then seamlessly integrated into the application.
4. Tcl/Tk is portable across three platforms: Windows, Macintosh, and UNIX. This is highly desirable for this project.

Tcl/Tk also has some disadvantages compared to custom compiled applications:

1. Tcl/Tk is optimized for objects that are primarily character strings. Mathematical operations are not fast or efficient in Tcl/Tk.
2. Since Tcl/Tk programs are essentially scripts, the source code may be readily copied or modified. This would not be acceptable for many commercial

applications.

3. Since scripts must be interpreted in order to execute, the execution speed of a Tcl/Tk program can be significantly longer than that of a compiled application.

None of these disadvantages adversely impacts the TREES and CROSSDATE programs, and, overall, Tcl/Tk is an effective, efficient, and inexpensive language for development.

2.2.3 Algorithm Implementation

Tcl/Tk is written using the C programming language. As mentioned earlier, it is easy to add Tcl/Tk to a compiled application, and it is similarly easy to extend Tcl/Tk's capabilities. The TREES application depends on the SADIE (System at Arizona for Digital Image Experimentation) image processing libraries for functionality. Since the SADIE libraries were written in C, it was easy to extend them to suit the needs of the TREES application with additional C source code. It was similarly easy to adapt the SADIE libraries for use in the Tk graphical environment by extending the Tcl/Tk interpreter to include SADIE. Tcl/Tk and C are highly compatible and many of Tcl/Tk's commands are very similar to C constructs.

The CROSSDATE application uses the same approach to programming language selection, even though it is not dependent on the SADIE image-processing library. In general, numerically intense operations such as convolution, statistics, Fourier transforms,

etc. are best done with C functions. Operations having to do with user interaction, list and string manipulation, and input/output are best done with Tcl/Tk procedures. The CROSSDATE program is initialized with a binary executable file compiled from C language source code. The binary executable code immediately invokes an event loop that uses a series of Tcl scripts to create the CROSSDATE display and perform CROSSDATE functions. Customized CROSSDATE functions are made available to the Tcl interpreter from the initial binary executable code.

2.3 Data Import and Export Capability

The CROSSDATE program has the capability of importing and exporting tree-ring chronology data files in the International Tree-Ring Data Bank ⁽¹⁾ (ITRDB) format for tree-ring chronologies. Appendix A provides a complete description of the data file formats for raw tree-ring width data and processed data files.

For the CROSSDATE application, master chronologies may also be imported and exported in the standardized ITRDB chronology format. The TREES program performs conversion of individual tree-ring width measurements to standardized index values. The standardization process involves fitting a curve to a tree-ring sample's width series, and then dividing each width value by the corresponding curve value to get an index value (e.g., detrending). This process allows samples with large differences in absolute growth

⁽¹⁾ - See <http://www.ngdc.noaa.gov/paleo/treering.html> for more information on the ITRDB.

rates to be combined, and can be used to remove any undesired growth trends present. This process is described in more detail in Chapter 3.

2.4 Basic Graphic User Interface Design

Figures 2-2 through 2-6 show basic features of the graphic user interface for the CROSSDATE program. The main screen window layout includes a menu bar, a time series graph display panel, and three sample skeleton plot display panels. Each panel has its own controls and information section on the left-hand side. The design of the main screen is an effort to mimic manual skeleton plot comparison. Up to three files may be opened at one time using the sample control buttons (Figure 2-6). The skeleton plot is shown in the individual sample panels. The corresponding index value time series is shown in the top panel. The individual samples may be repositioned on the screen by holding the left mouse button down and dragging the skeleton plot graph. This also moves the corresponding time series graph in the top panel to achieve an overlay effect. There is a scroll bar at the bottom of the screen to move the display to view any part of any sample's skeleton plot.

The choice to display three samples simultaneously was arbitrary, but considered reasonable by the project testing team. The decision was based on the amount of information to be presented in the skeleton plot panels, and the amount of information

that could be comprehended simultaneously in the time series panel. Basic sample information such as name, number of rings, and current position on the display are shown for each sample in the sample controls panel.

The main menu for the application (Figure 2-3) provides command functions to open and close session files (File Menu, Figure 2-4), utilize the Correlation Tool to compare samples (Correlate), compose a master chronology (Compose), and perform a quality control check on the validity of the composition (Check). A session file is a user convenience that allows for all current settings (samples, positions, colors, etc.) to be saved and re-opened as required. The analyst may also record notes with a session using the Edit menu.

The time series controls for the application (Figure 2-5) provide the ability to display time series information as non-normalized index values (Range shown: 0-2, series mean is 1.0) or as normalized index values (Range shown: -2 to +2, series mean is 0.0). This is a matter of individual analyst preference.

The sample controls for the application (Figure 2-6) provide the ability to open, close, and re-read sample data from disk, display sample and ring information, manipulate the display attributes of the sample, and change its start/end dates. The ability to re-read sample data from disk allows for image data or measurement changes made using the TREES program to be read immediately into the CROSSDATE application. The skeleton plot may be inverted (i.e., turned upside-down) using the *Invert* button. The time series

for a sample may be turned-off on the time series panel by clicking the *Hide Timeseries* button. The color of the skeleton plot lines and time series graph lines for each sample may be chosen using the *Color* button. The analyst may reset the start or end year of a sample or master chronology using the *Dates* button. The skeleton plot cutoffs (Chapter 3) may be changed using the *Skeleton Plot Cut-Offs* button.

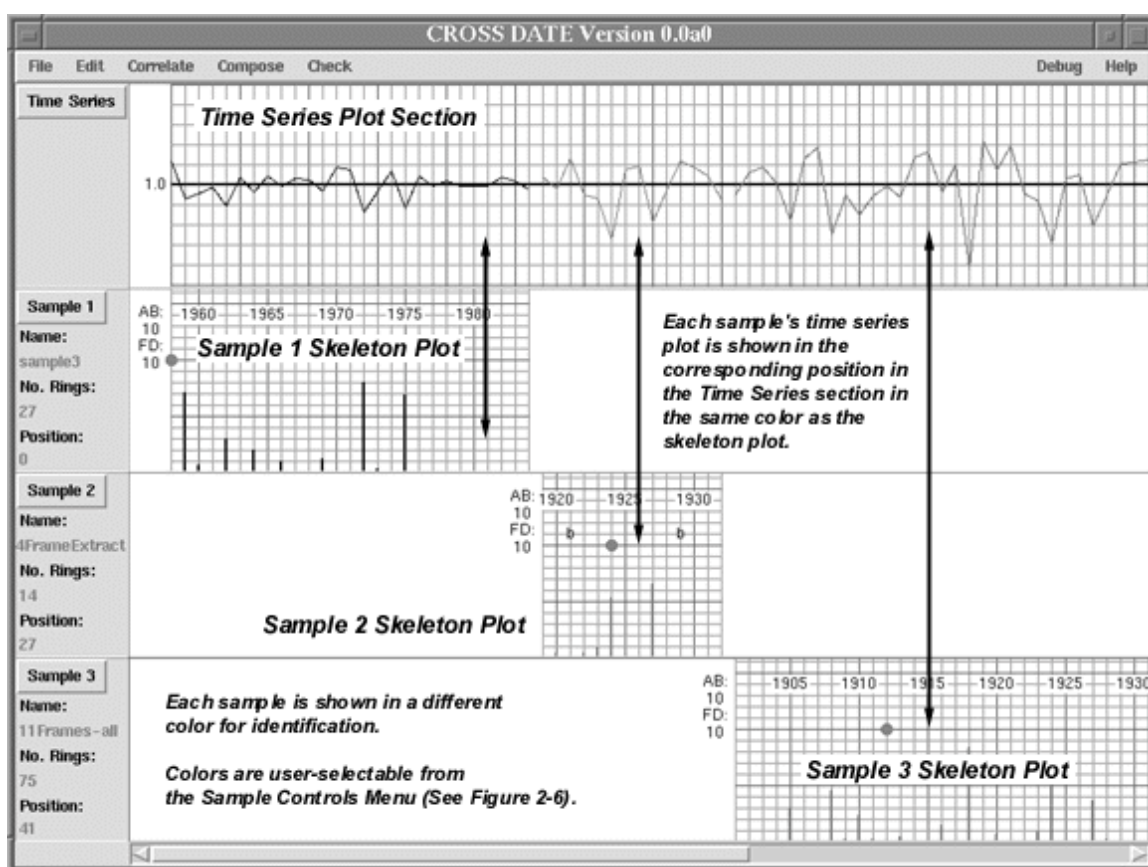


Figure 2-2 — CROSSDATE application main graphic user interface
(Descriptive notes are *italicized* and are not part of the CROSSDATE display screen.)

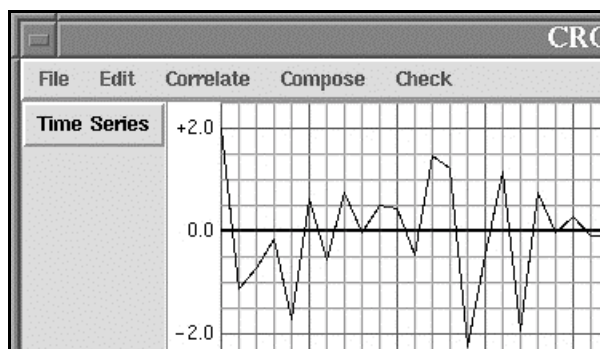


Figure 2-3 — CROSSDATE application main menu

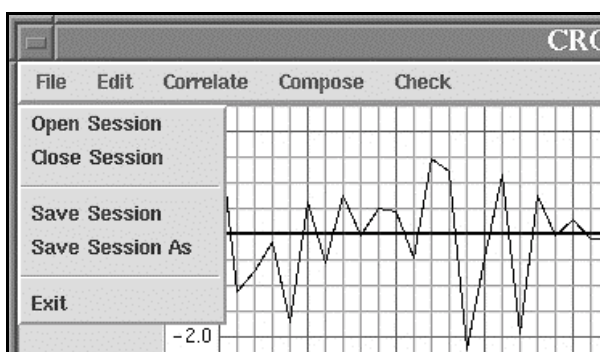


Figure 2-4 — CROSSDATE application File menu

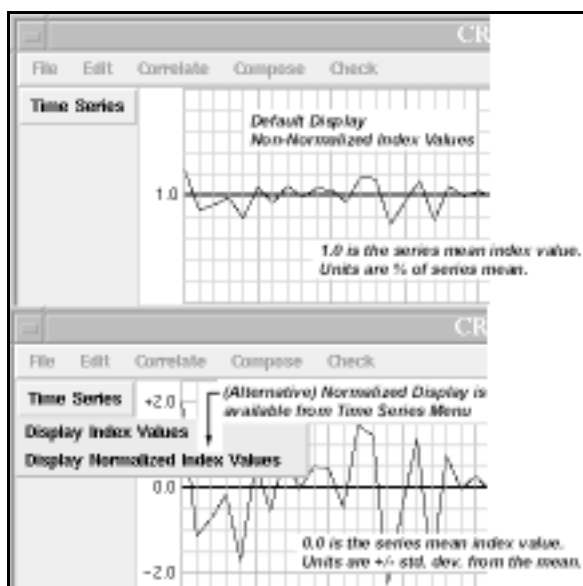


Figure 2-5 — CROSSDATE application time series controls

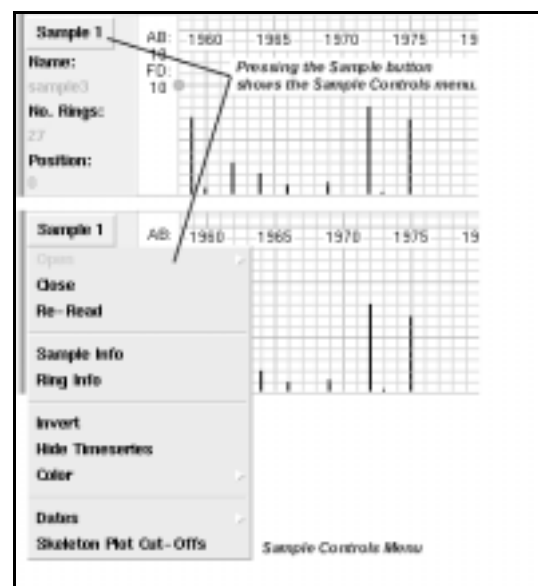


Figure 2-6 — CROSSDATE application sample controls

CHAPTER 3 — TREE-RING GROWTH PATTERN MATCHING

3.1 Tree-Ring Width Series Standardization

In order to compare tree-ring width series to determine site chronologies, a common frame of reference is necessary. Thus, methods to standardize width measurement series are crucial to dendrochronology. It is well known [8] that open-grown trees (e.g., trees widely spaced from each other) exhibit ring widths which decline over time. This is not necessarily true for closed-canopy forest sites where the proximity of trees causes a competition for water and sunlight that results in a complex combination of environmental factors affecting tree growth.

The primary reason for the decline in ring widths over time is due to the geometric and biologic considerations of adding wood each year to a cylindrical stem of increasing radius. Assuming that the volume of wood added each year is a function of a dominant climatic factor, and that there exists a systematic variation in ring widths for non-climatic reasons, the variation in the climate could be estimated by removing the non-climatic, systematic variations. The non-climatic, systematic variations are known as the *growth curve*. In dendrochronology, the detrending operation of modeling and removing non-climatic variance is known as *standardization*. After estimating a growth curve from the tree-ring width series, standardization is usually accomplished by dividing the observed ring width by the growth curve estimate as shown in Equation 3-1.

$$z(n) = r(n)/g(n) \quad (3-1)$$

where:

$z(n)$ is a dimensionless index value.

$r(n)$ is the observed ringwidth.

$g(n)$ is the growth curve estimate.

n is the year number in the ringwidth series.

The resulting indices are approximately stationary in variance, since the local variance of a ring width series is considered proportional to its local mean.

Other growth factors being equal, the growth curve exhibits exponential decay, independent of climate. Indeed, for many tree-ring width series the growth increment curve is modeled by a negative exponential curve. However, nature does not always behave according to such simple mathematical models. It is possible for events to occur which are non-climatic in nature, such as fire or insect infestation, that will cause a ring width series to depart from its expected systematic growth pattern. The term that dendrochronologists use for such events is *disturbances*.

Examples of models developed to handle disturbances include orthogonal polynomials [20] and cubic smoothing splines [10]. The orthogonal polynomial method approximates the non-climatic growth curve by successively higher order polynomials. A polynomial fit is accepted when the next two higher orders do not reduce the residual variance by 5% or more. The cubic smoothing spline is used as a digital filter to determine low frequency growth rate fluctuations. These low frequency fluctuations are then removed from the series using Equation 3-1. The cubic smoothing spline has been

used by dendrochronologists for many years to establish standardized ring width series and is implemented in the TREES program. Its use requires the specification of the degree of smoothing. An optimal degree of smoothing can be chosen when the tree's climate history is known. Since this is what is being sought in the first place, it is clear that the specification of the degree of smoothing is the subject of much experimentation. The TREES program uses a low frequency cut-off of twelve years, a value that is not so conservative as to remove too much of the high-frequency climatic signal. The cut-off frequency parameter is fixed for the TREES program for analytic consistency of sample comparisons. See Holmes et al. [26,27,28] and Munro [33] for further discussion of low-frequency cut-off values.

The cubic smoothing spline is simply an algorithm for accomplishing what was previously done manually by visual inspection. Prior to the arrival of computer-assisted methods, a tree-ring scientist would plot the ringwidth series measurements on a sheet of graph paper. A visual estimate would be made of the data's trend and drawn in as a curved line on the graph. A new series of detrended index values would be separately graphed by dividing the original series by the trend line (Equation 3-1). See [41] for a discussion of detrending from a signal processing perspective. Appendix F provides the mathematical basis for the implementation of the smoothing spline algorithm in the TREES and CROSSDATE programs.

Figure 3-1 shows the result of applying the cubic smoothing spline procedure to a

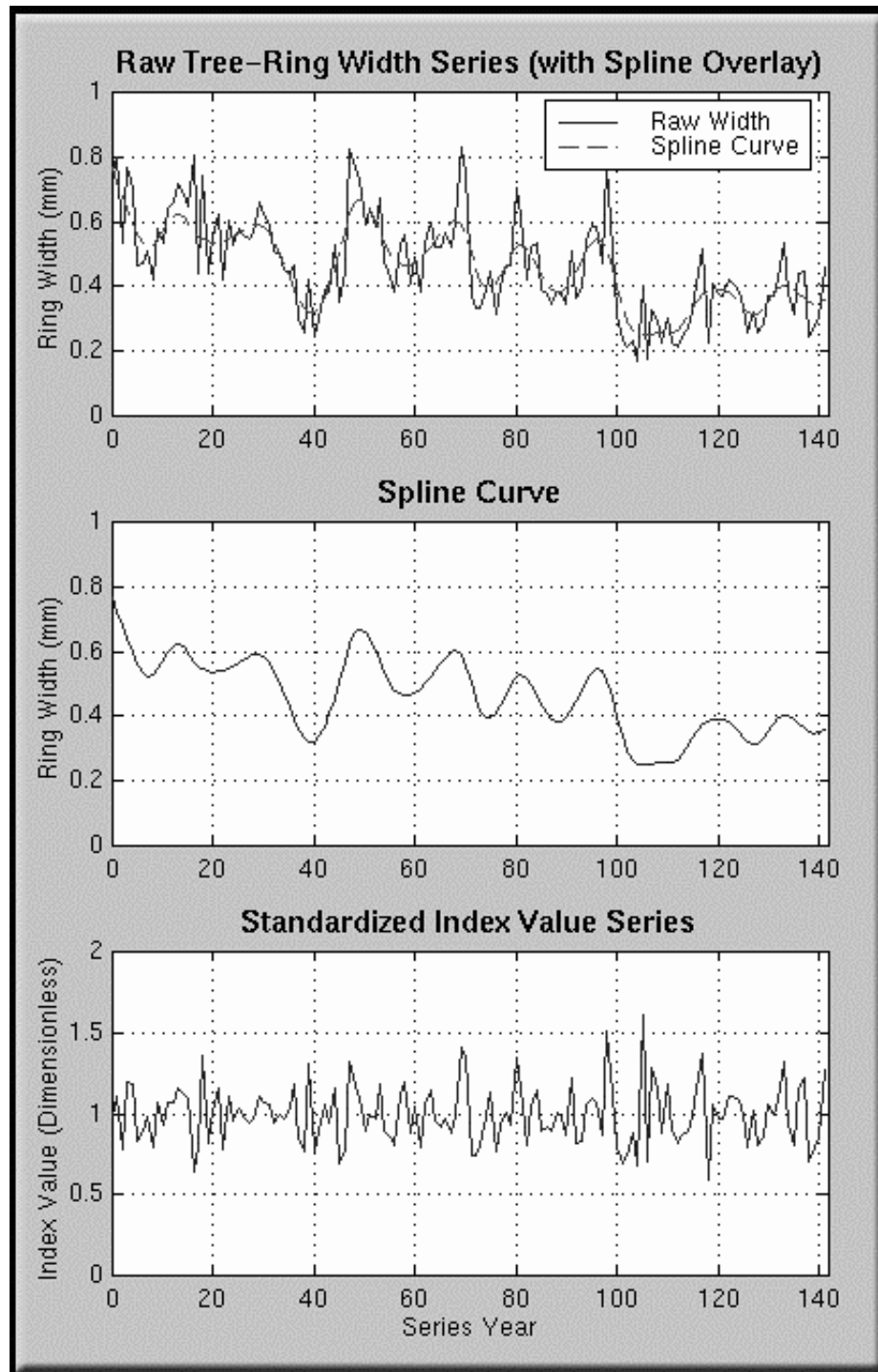


Figure 3-1 — Example of cubic smoothing spline ringwidth standardization

142-year tree-ring width series. The ring width series shows a low frequency trend that

decreases over time, but clearly cannot be modeled as a negative exponential function.

The middle graph shows the result of removing data with a frequency of occurrence greater than twelve years. The bottom graph shows the result of dividing the original series by the spline curve. Ring width standardization is performed by the TREES program and passed to the CROSSDATE program for further processing.

3.2 Skeleton Plotting

Skeleton plotting provides a means for characterizing and comparing tree-ring samples using ring-to-ring changes in width measurements. Large relative changes in local ring width measurements are exploited for tree-ring sample comparison. When done manually, a skeleton plot is created on graph paper where each vertical line corresponds to one annual tree-ring. This allows for alignment and comparison with other skeleton plots, since the distance between each vertical line is constant. Since the TREES and CROSSDATE programs seek to mimic manual methods as much as possible, the same graphing and alignment method is implemented in each program's graphical user interface.

Skeleton plotting highlights rings that are narrow compared to their immediate neighbors. See Figure 1-3 for an example skeleton plot. Narrow rings are marked on the skeleton plot with a bold dark line. The narrower the ring, the longer the line. The narrowest rings are arbitrarily represented with bold lines that are 2 major graph units in height. By marking relatively wide rings with a B, skeleton plotting captures the

extremes in variations of the sample's ring widths —average ring widths receive no attention. As exceptions in a ring width measurement series, the sequential pattern of large/narrow ring widths may be thought of as the sample's *features*, which are likely due to significant climatic variations. Since tree samples taken from a particular location should be consistently affected by the climate, it is then possible to crossdate samples based on *feature* comparison. This was one of the earliest methods of dendrochronology [45].

Skeleton plotting in the TREES and CROSSDATE programs has been adapted from [12] and [43]. The skeleton plotting algorithm begins with standardized index values as described in the previous section. The standardization process results in a series of index values with a mean value of 1.0, which simplifies many of the calculations for skeleton plotting. It is first necessary to determine the minimum and maximum index values in order to determine the dynamic range applicable for skeleton plotting. Next, the user is asked to provide an absolute difference cut-off (threshold) value (C_{ab}) and a first difference cut-off (threshold) value (C_{fd}). The skeleton plotting procedure uses the cut-off values to quantize the (*mean — min*) and (*max — min*) ranges to determine whether a skeleton plot line should be drawn for a particular year. Figure 3-2 shows the interval ranges for cut-off quantization.

The absolute difference cut-off is used for comparing an index value with the series (*mean — min*) range only. The first difference cut-off is used for comparing two

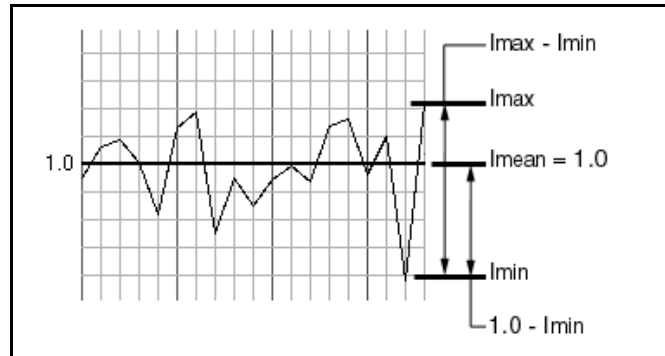


Figure 3-2 — Intervals for cut-off quantization

sequential index values with the series (*max* — *min*) dynamic range. The choice of cut-off values is $1 \leq (C_{ab}, C_{fd}) \leq 10$ when using a scale factor of 10 for drawing the skeleton plot line. A larger cut-off value results in more skeleton plot lines drawn than a smaller value. Choosing a particular cut-off value is a matter of individual preference, but for skeleton plot crossdating, the values should obviously be consistent between the samples. The TREES and CROSSDATE programs indicate the values of these parameters on the skeleton plot graphs. The default length of a skeleton plot line is zero.

A skeleton plot line will be drawn for an index value if the following absolute difference check is true (I_i is the current index value, I_{min} is the minimum index value):

$$I_i < 1.0 - \frac{1.0 - I_{min}}{C_{ab}} \quad (3-2)$$

The line will be drawn to the following scaled length (scale is assumed to be 10):

$$L_i = 10.0 \left[1.0 - \left(\frac{I_i}{1.0 - \frac{1.0 - I_{min}}{C_{ab}}} \right) \right] \quad (3-3)$$

A skeleton plot line will also be drawn for an index value if the following first difference check is true (I_{i-1} is the previous index value, I_{\max} is the maximum index value):

$$I_i - I_{i-1} < \frac{I_{\min} - I_{\max}}{C_{fd}} \quad (3-4)$$

The line will be drawn to the following scaled length:

$$L_i = 10.0 \left[1.0 - \left(\frac{(I_i - I_{i-1}) - (I_{\min} - I_{\max})}{\frac{I_{\min} - I_{\max}}{C_{fd}} - (I_{\min} - I_{\max})} \right) \right] \quad (3-5)$$

The skeleton plotting process is shown diagrammatically in Figure 3-3. Figure 3-4 shows the result of matching the skeleton plot features of two samples. The smaller sample is actually extracted from the larger sample for this demonstration. Note how the index values match very closely between the samples except at the ends of the smaller sample. The cubic smoothing spline is dependent on the actual data end points as beginning and ending points.

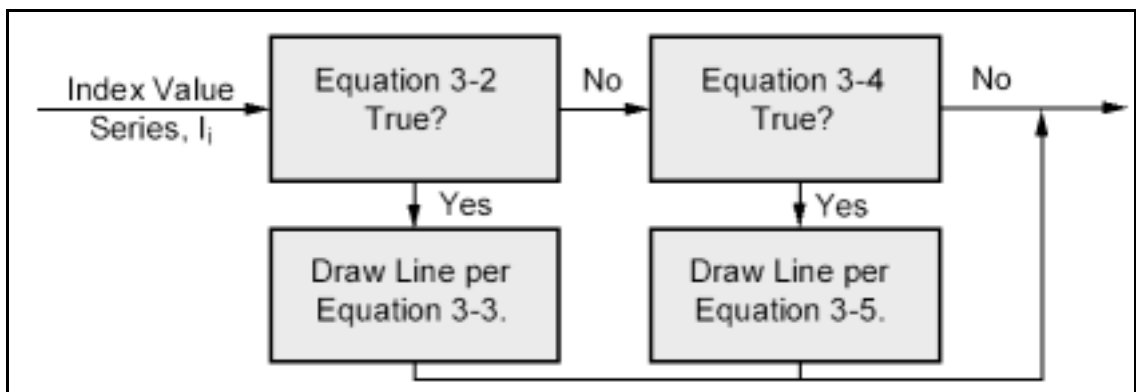


Figure 3-3 — Skeleton plot line drawing process

3.3 Correlation Pattern Matching

Once a tree-ring sample has been standardized and skeleton plotted, the next step is to establish the dates that correspond to each annual tree-ring. If the sample was taken

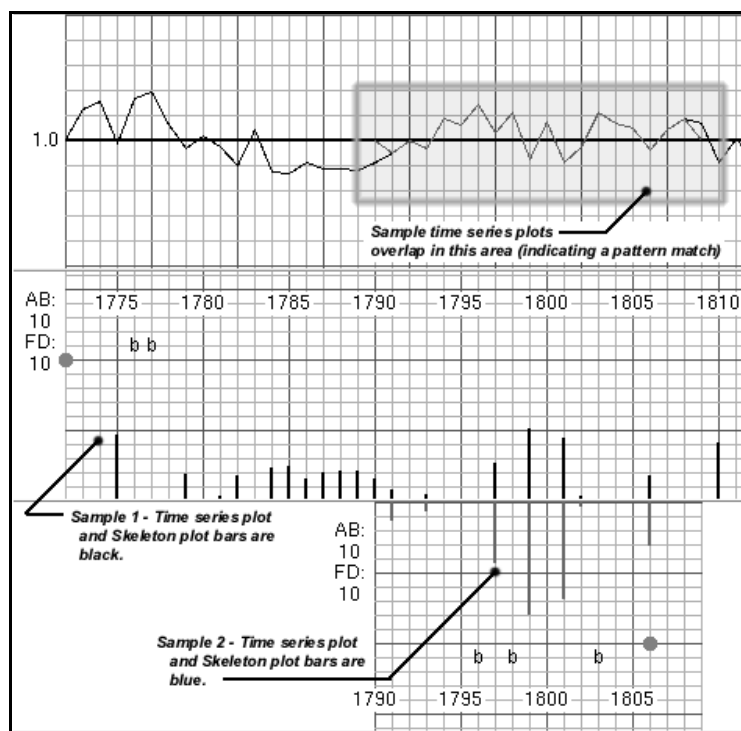


Figure 3-4 — Skeleton plot crossdating example

from a freshly cut tree, establishing its years of growth is a trivial matter since knowing the year for the last annual ring establishes the years for all the remaining rings by simply counting backwards. This is the beginning of building a master chronology as shown in Figure 1-1. However, samples are often found at a site where the dates of growth are unknown. To establish dating for these samples, it is necessary to match their growth pattern variations with samples of known growth dates. This process is known as *crossdating* and serves two mutually dependent purposes,

1. Samples with known dating are used to build *master chronologies*, which represent patterns of tree-ring growth variation over some time period, for a particular location, and due to a single climate factor.
2. The *master chronologies* are then used to establish dates of growth for samples with unknown dating. Once dated, these samples are often used to extend existing *master chronologies*.

There have been several methods used by dendrochronologists for crossdating tree-ring samples to create master chronologies. Prior to the widespread use of computers, samples were manually standardized and skeleton plotted as described in sections 3.1 and 3.2. An analyst could then simply slide the plots across each other to determine a point where either the time series graphs or the skeleton plots seemed to match. Indeed, tree-ring scientists still use this procedure to check the validity of more sophisticated computer-assisted models. This manual method of visually comparing sample representations by sliding them across each other is imitated graphically as part of the basic GUI in the CROSSDATE program.

Using visual comparison, it is difficult to quantify the degree to which two samples match. Therefore, the CROSSDATE program provides correlation pattern matching to aid analysts in quickly determining (1) the point where two samples have the greatest confidence of a match and (2) providing data quantifying how well the two samples match at any point of overlap. This is not intended to be a substitute for the

judgement of an experienced dendrochronologist; it is simply a tool for expediting the crossdating task.

Statistics and signal processing provide the mathematical basis of correlation pattern matching. The correlation coefficient, r , is a measure of the estimate of the linear relationship between two series variables X and Y [47]. The range for the correlation coefficient is $[-1, +1]$. A correlation coefficient of 0 indicates no linear relationship between X and Y . A correlation coefficient of $+1$ or -1 indicates a perfect linear relationship between the variables X and Y (e.g., all values lie on a single line). Other values may be interpreted as $100 \times r^2$ of the variations in Y may be accounted for by the linear relationship with the variable X . For two tree-ring index value series, a match could be indicated by a large positive correlation coefficient (i.e., close to $+1$).

Of course, comparing two series for a possible match involves more than one correlation coefficient. If the X series has N individual values, and the Y series has M individual values (numbered 0 to $N-1$ and 0 to $M-1$, respectively), there will then be a total of $(N + M - 1)$ different correlation coefficients for each possible overlapping match between the samples (Figure 3-5). Certainly, overlapping positions with high positive correlation coefficients are of most interest. It is also readily seen in Figure 3-5 that not all overlapping positions have the same number of overlapping values, L .

The question of whether the correlation coefficient arose by chance needs to be answered in order to determine which overlap position really represents the best match.

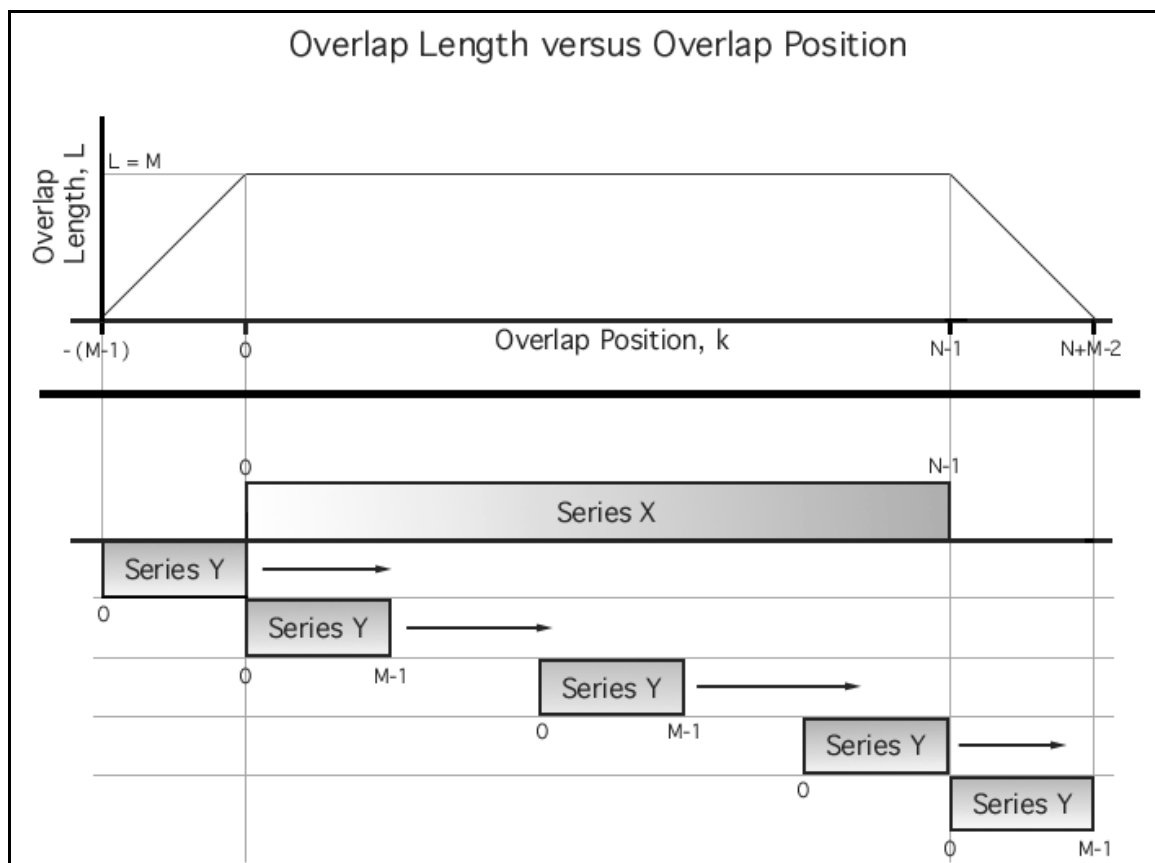


Figure 3-5 — Overlapping match positions between two series

Baillie and Pilcher [1] and Munro [33] suggest the calculation of the Student's t -statistic in order to measure the probability that the observed value of the correlation coefficient occurred by chance. Walpole and Myers [47] cast the same question as a statistical test of the null hypothesis that there is no linear association ($r = 0$) between the series variables at a level of significance, $A(t/v)$. This is known as a difference of means test. Equations 3-6 through 3-8 show the mathematical basis for the calculations to be performed. It should be noted that r and t are actually functions of overlap position and

are calculated for each possible overlap position, k , to quantify all the points of series correlation.

The correlation coefficient $r(k)$ is calculated for each overlap position, k , according to Equation 3-6 [33]:

$$r(k) = \frac{S_{xy}}{\sqrt{S_{xx} \cdot S_{yy}}}, \quad k = -(M-1), \dots, 0, \dots, N+M-2 \quad (3-6)$$

where S_{xx} , S_{yy} , and S_{xy} are the sample standard deviation values for the overlapping portions of the X, Y, and XY series respectively.

The hypothesis test of the t -statistic from Equation 3-7 for the level of significance obtained from Equation 3-8 assumes the data is bivariate-normally distributed (e.g., whitened). The natural log transformation in the following equation accomplishes this [47],

$$t(k) = \frac{\sqrt{L-3}}{2} \ln \left[\frac{1+r}{1-r} \right] = \sqrt{L-3} \tanh^{-1}(r), \quad k = -(M-1), \dots, 0, \dots, N+M-2 \quad [33] \quad (3-7)$$

with a level of significance given by,

$$A(t|v) = \frac{\Gamma[(v+1)/2]}{\Gamma[v/2]\sqrt{\pi v}} \int_{-t}^t \left(1 + \frac{x^2}{v} \right)^{-\frac{v+1}{2}} dx \quad (\text{CDF of Student's distribution [38]}) \quad (3-8)$$

where:

r is the correlation coefficient, $r(k)$, calculated in Equation 3-6.

L is the overlap length between the two series.

ν is the degrees of freedom for the hypothesis test significance ($\nu = L-2$).

Munro [33] explains autocorrelation (also known as "persistence") may exist among t values (calculated from Equation 3-7) that will tend to artificially inflate the significance of a particular match. There has been much work in autoregressive model fitting to correct this difficulty [8,33], however, the procedures are computationally expensive to produce a good result. Munro [33] suggests a simple correction procedure to test the best matching position. It is assumed that the distribution of t -values (calculated from Equation 3-7) from *non-matching* positions is gaussian with unknown mean and standard deviation, and the t -value from the *matching* position (if it exists) will be from a different normal distribution with a higher mean. The procedure determines the significance probability that the matching position is an outlier.

The sample mean and standard deviation are calculated for the entire population of t values using standard statistical methods. Each potential matching position, $t(k)$, is then normalized using Equation 3-9 to determine its normalized deviation from the mean t value, μ_t . Next, the normalized deviation, $T(k)$, is corrected using Equation 3-10 to produce a reasonable approximation to the true significance probability, P , for $T(k)$ using Equation 3-11.

$$T(k) = \frac{t(k) - \mu_t}{s_t} \quad (3-9)$$

$$T(k)' = \sqrt{\frac{Z(Z-2)T(k)^2}{(Z-1)^2 - ZT(k)^2}} \quad (3-10)$$

$$P = 1 - \left[\frac{Z}{2} \cdot A(T(k)'|Z-2) \right] \quad (3-11)$$

where:

μ_t and s_t are the mean and standard deviation for all $t(k)$ values..

Z is the total number of t values ($Z = N + M - 1$).

$A(T(k)'|Z-2)$ is the CDF of the Student t distribution (per Equation 3-8).

Computationally, S_{xx} , S_{yy} , and Equations 3-7, 3-8, 3-9, 3-10, and 3-11 may be calculated once for each value of k with a simple update to the series sum and series sum of squares values. However, S_{xy} cannot be calculated in this manner. The sum of the product of the X and Y series values for each overlap would otherwise need to be recalculated for each overlap. This requires on the order of $N \times M$ calculations per series pair, or N^2 calculations when $N \approx M$.

Recognizing that S_{xy} is a translated and scaled version of a convolution provides insight on how the calculation may be performed more efficiently. Figure 3-5 shows an operation similar to convolving X and Y , except that Y is not flipped in time, as in convolution. This relationship is stated mathematically in Equation 3-12 as a function of overlap position, k .

$$S_{xy}(k)^2 = \frac{\sum_{n=0}^{L-1} x(n)y(n+k)}{L-1} - \mu_x\mu_y, \quad k = -(M-1), \dots, 0, \dots, N+M-2 \quad (3-12)$$

where:

$x(n)$ and $y(n)$ are individual values of series X and Y at series position n.

L is the length of the overlap.

M is the length of the shorter series (Y is assumed to be the shorter series).

μ_x and μ_y are the means for the X and Y series respectively.

Equation 3-12 assumes μ_x and μ_y are constant for each series (e.g., stationary) [33]. This seems reasonable based on the detrending process described earlier.

In signal processing, the numerator of Equation 3-12 is known as the *cross-correlation function* [41]. Correlation in the time domain is equivalent to complex conjugate multiplication in the frequency domain using Fourier transforms. An FFT (Fast Fourier Transform) is known to be on the order of $N \times \log_2(N)$ calculations per series per transform where N is a power of 2. This process is quite efficient even for relatively small values of N . The process used to calculate the cross-covariance term, S_{xy}^2 , is shown in Figure 3-6.

It may be mathematically proven that the standard deviation will always be greater than or equal to zero, and the absolute value of the correlation coefficient, $|r|$, will always be less than or equal to one. However, in floating point arithmetic on a digital

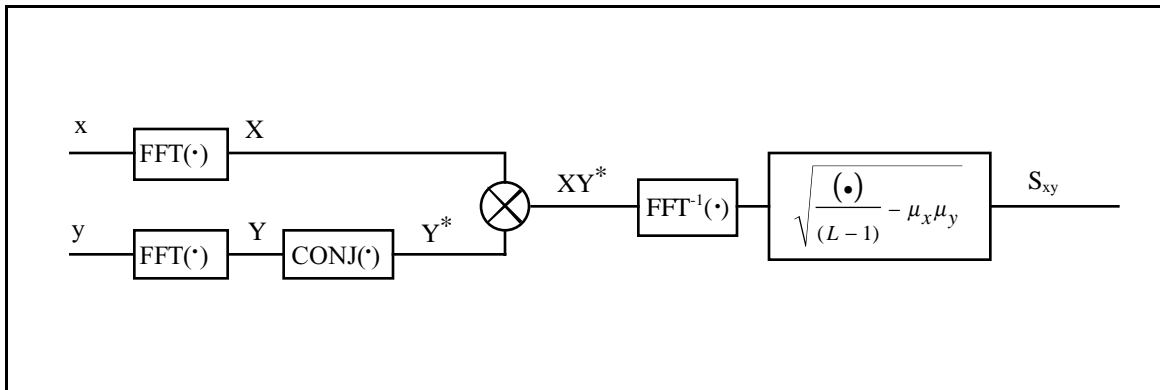


Figure 3-6 — Calculation of the cross-covariance term, S_{xy}

computer, round-off error may cause these conditions to be violated. Specifically, S_{xx} or S_{yy} may become negative when the arithmetic value is very close to zero. Very small values of S_{xx} and/or S_{yy} in the denominator of Equation 3-6, may cause $|r|$ to be greater than 1.0. These error conditions are caught by the CROSSDATE program, and exceptions are noted.

3.4 User Interaction — Correlation Tool

Correlation pattern matching is implemented in the CROSSDATE program using the correlation tool. *Correlate* → *Correlation Tool* presents the user with a dialog box (Figures 3-7 and 3-8) to select the samples to correlate. After selection, the program calculates the correlation and presents a separate graphic panel (Figure 3-9) which plots the correlation coefficient against each overlap position.

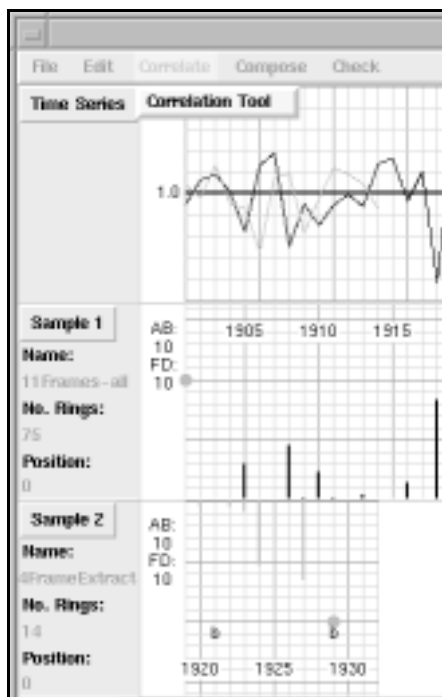


Figure 3-7 — Correlation tool menu

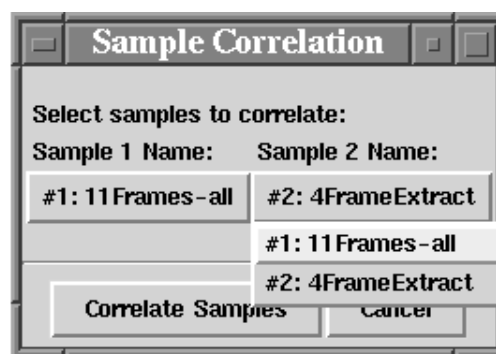


Figure 3-8 — Correlation tool sample selection dialog

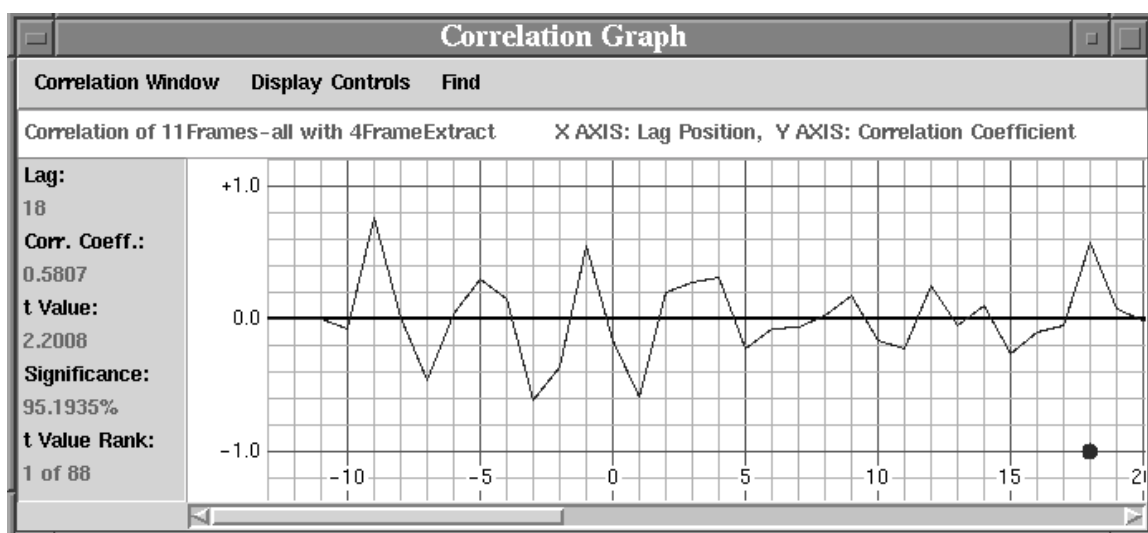


Figure 3-9 — Correlation tool graph panel

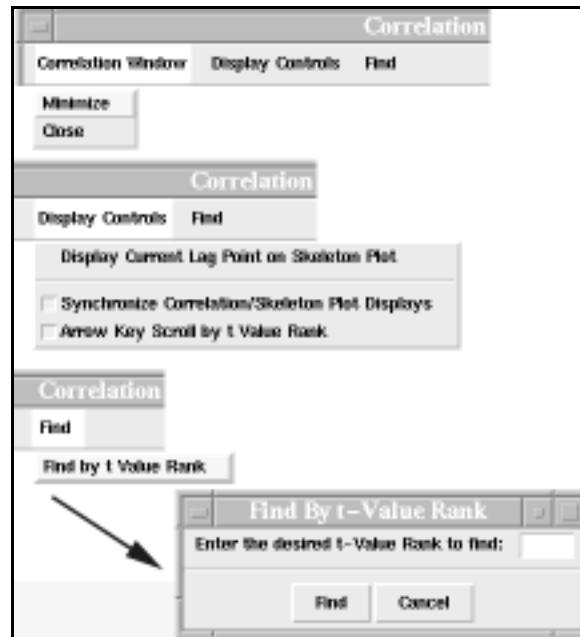


Figure 3-10 — Correlation tool menus and options

The correlation graphic panel (Figure 3-9) is divided into four sections —Menu bar, Sample names, Correlation Coefficient graph, and Information panel. The coefficient graph is drawn similar to the time series graph in the main screen. After calculating the correlation coefficient (Equation 3-9) for each possible lag position, the t-value (Equation 3-10) and its significance (Equation 3-11) are calculated to determine how well the two series match at each overlap position. The t-values are then sorted into descending order (highest value first). This ordering determines the t-value *rank* of the match position. The "best" match is the position with the highest t-value. When the correlation tool graph panel is first presented to the user, a blue highlight dot is placed on the lag (overlap position) with the highest t-value, and the graph is scrolled to display that position (in Figure 3-9, lag 18 has the highest t-value). The mouse and/or keyboard arrows may be

used to select other lag positions. The selection action moves the highlight dot, and displays the data corresponding to the selected lag in the information panel. It is interesting to note that higher correlation coefficients may occur in the transient portions of overlap, but they generally have a lower t-value due to the smaller number of overlapping data values. Additionally, there must be four or more points of overlap in order to characterize the lag. (See Equation 3-7, $L-3 > 0$.)

The correlation tool provides several control options in its menu bar. Since the correlation graph panel is a separate window from the CROSSDATE main screen, the option to minimize or close the window is provided under the Correlation Window menu button. The application does not store any correlation data once the correlation window is closed.

Options to synchronize the displays of the correlation graph panel and the application main screen are provided under Display Controls. The *Display Current Lag Point on Skeleton Plot* button automatically adjusts the skeleton plots on the main screen to correspond to the lag position currently highlighted in the correlation graph panel. The check box for *Synchronize Correlation/Skeleton Plot Displays* automatically synchronizes the correlation graph panel and main screen displays when checked. Changing the selected lag position on the correlation graph simultaneously moves the skeleton plot display, and moving the skeleton plot display moves the corresponding lag position on the correlation graph. The check box for *Arrow Key Scroll by t Value Rank* changes the

function of the keyboard arrow keys to scroll by t -value rank, rather than by lag position, when checked. In this mode, the arrow keys scroll the highlighted lag from rank n to $n-1$ or $n+1$ using the left or right arrow keys. The lag position of the rank is immediately scrolled into view in the correlation graph panel.

Find → *Find by t Value Rank* presents a dialog box for entering a specific rank to find. Pressing the *Find* button selects the lag corresponding to the desired rank and scrolls it into view in the correlation graph panel.

CHAPTER 4 — COMPOSING A MASTER CHRONOLOGY

4.1 Master Chronology Composition

After matching samples according to their growth patterns, it is then possible to begin composing a master chronology. According to [45], a master chronology is a dated composite chronology for a group of specimens. For this definition, *composite* means the arithmetic average of the component samples across the entire period of overlap. The numeric values averaged are the standardized sample index values discussed in Chapter 3. The samples used to compose a master chronology are known as its components. In order to date the master chronology, at least one ring date must be known from at least one component. One goal of the crossdating process is to reduce the data to a form that can be readily used and understood by others. The master chronology serves this purpose in dendrochronology.

It is important to note that the process of composition is a simple arithmetic average of all components with uniform weighting given to each component and not aggregate averages with non-uniform weighting:

$$MV(n) = \frac{\sum_{j=1}^N CV_j(n)}{N} \quad (4-1)$$

Where:

$MV(n)$ is the master chronology index value at year n

$CV_j(n)$ is the index value of component j at year n

N is the number of components used to create $MV(n)$ at time n .

In order to maintain the integrity of the simple arithmetic average it is necessary to maintain at least two pieces of Equation 4-1 for data storage and transmission. The master chronology index value, $MV(n)$, and the number of component samples, N , are commonly used for this purpose, and the International Tree-Ring Data Bank (ITRDB) standard format for the storage and exchange of master chronology data requires this information (See Appendix A for ITRDB data file formats).

A master chronology may include specimens from a widely varying scale of locations, for example, from a small local stand to an entire state. The specimens must have a homogeneous pattern of variation in common, where the component ring patterns resemble each other very closely. In other words, a single environmental factor must be dominant in tree-ring growth variation for the region. However, as the tree-ring sample location area increases, the variability of the dominant environmental factor will also increase, making it more difficult to crossdate individual samples.

This does not necessarily invalidate creating master chronologies for large geographic areas. Rather, scientists may study regional variations in climate across large regions by compiling chronologies in a hierarchical manner where successive levels within

the hierarchy represent larger sample location areas within the region. Table 4-1 shows an example of a ten level hierarchy covering a 1 million square kilometer region. The practical limit for climatic similarity for purposes of crossdating is 800 —1100 km (500 — 700 mi.) [45].

Level	Size km × km (mi. × mi.)	Area Km ² (mi ²)	Possible Number of Regions
0	1024 x 1024 (600 x 600)	1 million (400,000)	1
1	512 x 512 (300 x 300)	250,000 (100,000)	4
2	256 x 256 (150 x 150)	65,000 (25,000)	16
3	128 x 128 (75 x 75)	16,000 (6,000)	64
4	64 x 64 (40 x 40)	4,000 (1,600)	256
5	32 x 32 (20 x 20)	1,000 (400)	1,024
6	16 x 16 (10 x 10)	250 (100)	4,096
7	8 x 8 (5 x 5)	64 (25)	16,384
8	4 x 4 (3 x 3)	16 (6)	65,536
9	2 x 2 (1 x 1)	4 (2)	131,072

Table 4-1 Example master chronology sample location area hierarchy

Since climatic variation affects wide areas, and tree-ring samples are typically taken from smaller regions within wide areas, hierarchical composition of master chronologies is useful for regional climatic interpolation. First, the most accurate chronology dating is established based on composing samples from a single site or with close proximity to each other. Next, accurately-dated chronologies from the relatively small areas are combined to create chronologies for larger areas. Admittedly, the

chronologies for the larger areas are of limited value for dating individual samples. However, as the scale of location increases, the master chronology becomes a representation of the expectation of climatic variation across the region. The larger area chronologies serve as a benchmark for comparing climate variation among the smaller areas. This is obviously useful for studying long-term climate changes, but another purpose is predicting a chronology for an area where tree-ring scientists are unable to assign dates to samples. An estimated chronology is made using a spatiotemporal interpolation technique.

Table 4-1 shows that a small number of levels (ten) can represent a vast range of areas (4 km² to 1 million km²). A climatic variation model may be estimated by sampling master chronologies regionally at a given scale level. The key to this technique is hierarchical composition of master chronologies based on the area of the location sampled.

4.2 Master Chronology Data Structure

Composing master chronologies from individual samples and other master chronologies, and the ability to determine and review the specific samples from which a master chronology is composed for purposes of returning to the original wood sample are significant requirements of the CROSSDATE program. To meet these requirements, it is first necessary to define a master chronology data structure.

In the context of the CROSSDATE program, tree-ring samples and master

chronologies will be considered two versions of an object called `sample data`. In object-oriented software terminology, tree-ring sample objects and master chronology objects inherit from the root generic object — `sampledata`. This relationship is shown in Figure 4-1. All objects have session/file, skeleton plot, and export skeleton plot information in common. This is the source of time series plot and skeleton plot data for the TREES and the CROSSDATE programs.

The master chronology data structure includes header information to identify and describe the chronology and key data for each sample component. Key data fields for each sample component include the filename, the sample size (e.g., number of rings), and starting year position within the master chronology. The data structure also includes the component's file date attribute and type (e.g., TREES data file, CROSSDATE chronology, or ITRDB chronology). The basic graphic user interface for the CROSSDATE program allows for composition of up to three samples per master chronology file. The sample's status data member is used to indicate the presence or absence of sample component 1, 2 or 3. Appendix D contains specific details on the implementation of the master chronology data structure in the CROSSDATE program.

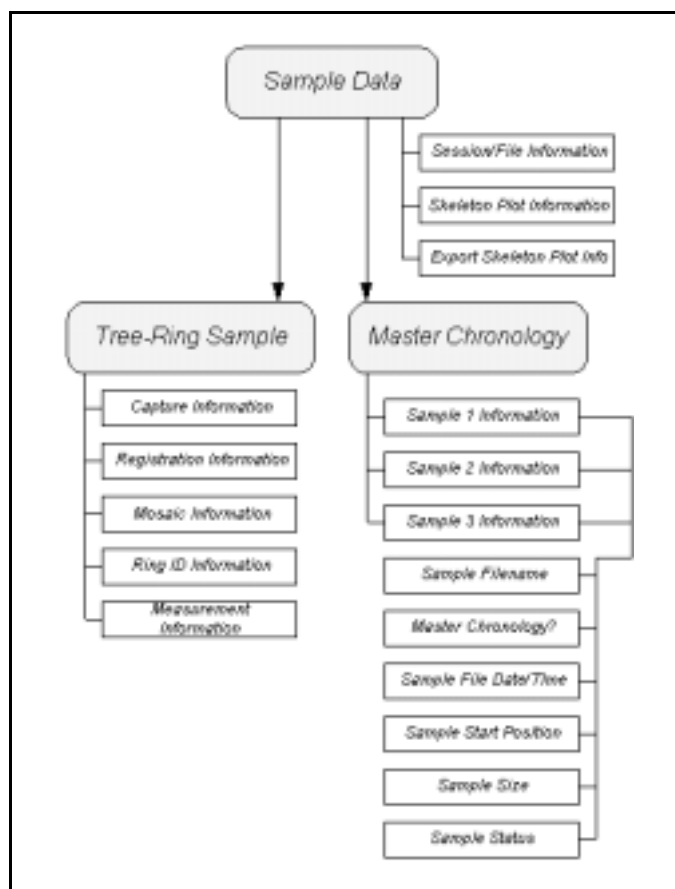


Figure 4-1 — Object data structures and inheritance

4.3 Hierarchical Composition of Master Chronology Components

Based on the definition of a master chronology [45], there is virtually no limit ⁽¹⁾ to the number of samples that may be used to compose a master chronology. However, for composition purposes, the CROSSDATE program allows for a chronology to be

(1) The ITRDB data format for master chronologies imposes a limit of 999 samples per composition.

composed from only up to three samples at a time because of display space limitations (Figure 2-2). This apparent dilemma is solved using hierarchical composition of tree-ring samples and master chronologies. This method allows chronologies to be composed from a virtually unlimited number of samples. This method readily supports the sample location area hierarchy described at the end of Section 4.1.

In this context, a hierarchy is a *parent* \rightarrow *child* relationship where a master chronology is the *parent* and tree-ring samples or other master chronologies are the *children*. This is a non-linear relationship where each component has zero or more successors [24]. In mathematical and computer science terms, this is known as a *tree* structure. In order to prevent confusion with a dendrochronologist's idea of tree structure, the term *virtual tree* will be used. Unlike real trees, the *root* is at the top of the virtual tree and serves as the point of access for all other parts of the virtual tree. A virtual tree's components are often referred to as *nodes*. Virtual tree components that do not have any successors are known as *leaves*. Tree-ring samples will be the leaves of this virtual tree structure since they cannot have successors in this hierarchy. The *level* of a node is its distance from the root (e.g., the number of predecessors including the root). This is also known as its *depth* in the virtual tree.

In this virtual tree structure, some absolute rules must be followed. First, the virtual tree structure must have one and only one root. Next, a parent node (master chronology) may become a child node to only one other parent node. A parent node may

not become a child node to itself at any level in the hierarchy. And finally, there must exist only one path from each node to each other node of the virtual tree structure. An additional requirement could be imposed specifying that leaf nodes (tree-ring samples) have one and only one parent node. While this is not absolutely necessary to the successful implementation of the virtual tree structure, it seems reasonable from a practical point of view that a tree-ring sample should only be included once in any chronology hierarchy.

In mathematical terms, the virtual tree structure is known as a *directed acyclic graph*. A graph is an ordered set of vertices (nodes) and edges (parent-child relations) that connect the vertices. A directed graph implies a *from-to* order of the vertices of an edge. In other words, the edge (a,b) is not the same as edge (b,a) between vertices a and b. A path is a sequence of edges required to travel from one vertex to another. In order for the graph to be acyclic, there must be only one path to each node in the virtual tree. A directed acyclic graph is also assumed to be a connected graph. This means that there must be at least one path to each node in hierarchy.

A structure satisfying the requirements of a directed acyclic graph allows for an algorithm known as Depth-First Search (DFS) to be used to visit all of the nodes of the graph [21]. The input is the root node of a directed acyclic graph. The output is some function performed on all of the component nodes. The DFS algorithm recursively invokes itself in the course of its execution. Figure 4-2 shows the pseudo-code for the

DFS algorithm (adapted from [21]).

<i>Algorithm DepthFirst</i>
<pre> procedure DepthFirst(root:node) { /* Performs some function on nodes in a directed acyclic graph in depth-first order from node root */ mark root visited; perform some function on root; for each child node n of root, do { if n has not been visited then DepthFirst(n); else STOP, cycle found } </pre>

Figure 4-2 – Depth-First Search algorithm

Figure 4-3 shows an example hierarchy of tree-ring samples and the result of performing DFS to get a listing of all the nodes. This procedure satisfies the requirement that each component of a master chronology is identifiable. Furthermore, it allows the CROSSDATE program to access each component of a master chronology while storing minimal component data. The disadvantage of this approach is the number of intermediate chronologies that must be created as "place-holders" in order to compose a single master chronology with a total of N samples (denoted C_i in Figure 4-3). For example, composing $N=9$ samples into one chronology, three samples at a time, requires an additional 3 intermediate chronologies, $N=27$ samples requires an additional 12

intermediate chronologies, $N=81$ samples requires an additional 39 intermediate chronologies, and so on. It can be shown that when a limitation exists on the number of chronologies that can be composed at one time the following relationships hold with respect to the number of nodes:

For N leaf nodes, combined C leaves at a time:

$$M = \lceil \log_c(N) \rceil \quad \text{Minimum number of levels} \quad (4-2)$$

$$T = \frac{C^{M+2} - 1}{C - 1} \quad \text{Maximum number of nodes} \quad (4-3)$$

$$I = T - N - 1 \quad \text{Maximum number of intermediate nodes} \quad (4-4)$$

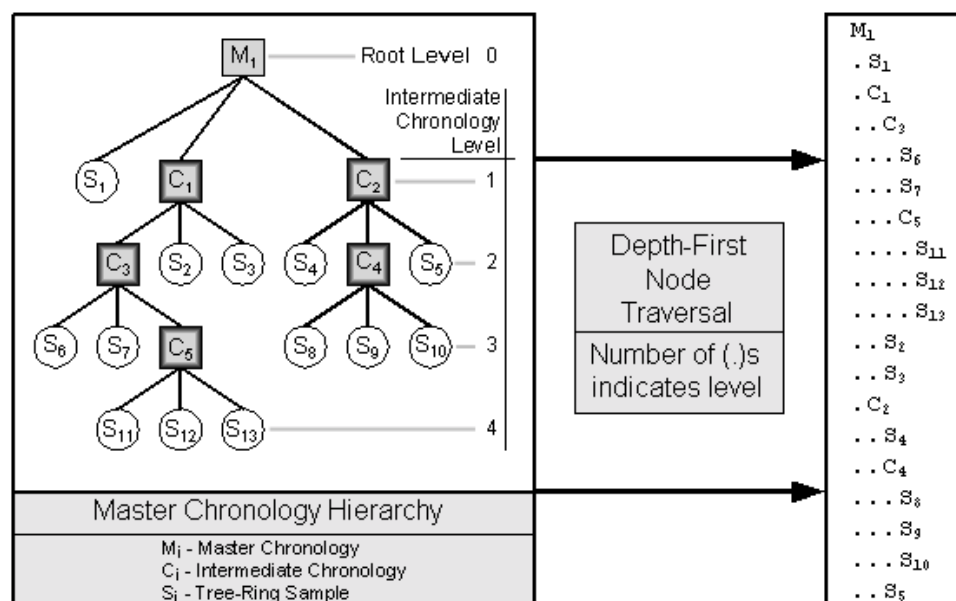


Figure 4-3 — Example Depth-First Traversal of a master chronology hierarchy

4.4 Importing and Exporting Data

The capability to exchange data in common formats with other tree-ring application programs is a specific requirement of the CROSSDATE program. Another fundamental requirement is the ability to determine and review the specific samples from which a master chronology is composed for purposes of returning to the original wood sample. For the CROSSDATE program, as long as the samples used in master chronology composition are the result of the TREES program, the ability to return to an image of the original wood sample is ensured. However, when master chronologies are composed with imported ITRDB data, no such images of the original component sample are available. Furthermore, the imported chronology provides no means for identifying the sample components used in its composition. Understanding that certain limitations are inherent in the data exchange process, it is still quite useful to exchange data with other application programs.

The CROSSDATE program is able to access three types of files to be displayed as samples —(1) TREES data files, (2) CROSSDATE master chronology files, and (3) ITRDB master chronology files. See Figure 4-4 for the user interface for importing an ITRDB file. The CROSSDATE program subjects each file type to considerable error checking prior to accepting the data. For ITRDB files, the following checks are performed: unexpected end-of-file, improper start or end years (integer numbers, end year > start year), proper decade years (sequential from start to end), correct amount of

data based on the number of years in the chronology, and non-negative annual (*index value, number of sample*) data pairs with ranges: $0 \leq \text{index value} \leq 9900$ and $0 \leq \text{number of samples} \leq 999$.

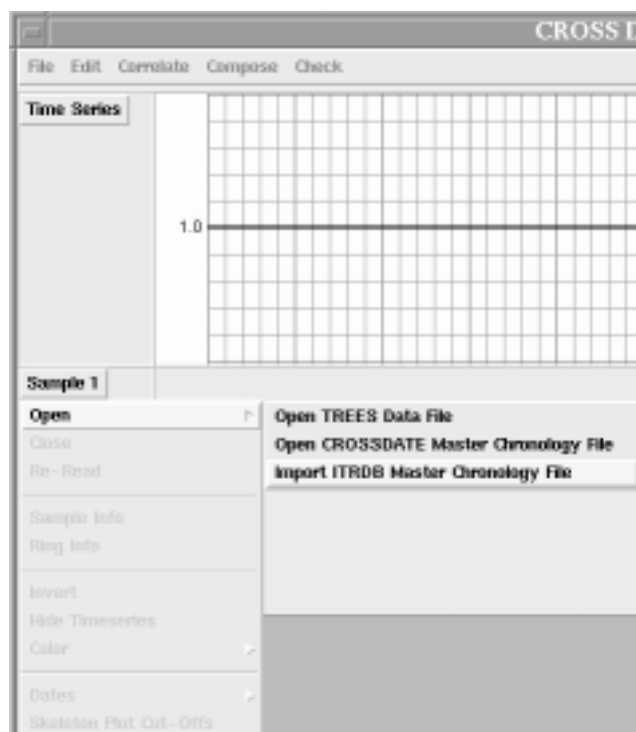


Figure 4-4 CROSSDATE sample open menu

The CROSSDATE program provides for exporting a master chronology in ITRDB format as the *File* → *Master Chronology Export* menu option from the Master Chronology display graphic user interface per Figure 4-12. The exported chronology file has no reference to its CROSSDATE master chronology file and no reference to any of its sample component files beyond the number of samples used to compose each year's index value.

4.5 User Interaction — Composition Tool

User interaction with the composition tool begins after the tree-ring samples have been positioned on the main screen at the positions where the analyst believes the best match exists. This will usually take place after correlation pattern matching has been performed as described in Chapter 3. After matching the samples, the analyst may reset and save a new start or end date as required at this point for TREES data file sample components and CROSSDATE master chronology components. The program will not allow the user to reset the start or end date of an imported ITRDB chronology component. This would be the responsibility of the program that created the ITRDB chronology.

The interaction process begins with selecting the *Compose* → *Compose Master Chronology* selection from the main screen menu (Figure 4-5). The user is then asked to select which samples are to be included in the chronology by checking the appropriate button next to the name of the sample (Figure 4-6). The program initially defaults to selecting all of the currently displayed samples. The program then asks the user to specify either the starting year or the ending year of the chronology (Figure 4-7), after which a dialog screen for entering master chronology header description information is presented (Figure 4-8). Character count limits and example entries are provided in the information dialog to aid the user. The user is then asked for a file name of the master chronology for saving to disk (Figure 4-9). Upon successfully writing the file to disk, the

user is asked if the master chronology and its components are to be displayed (Figure 4-10). A typical Master Chronology Component Display screen is shown in Figures 4-11 and 4-12.

A master chronology and its components may be displayed anytime by selecting the *Compose → Display Master Chronology* selection from the main screen menu (Figure 4-5). The user is then provided a (File Open) dialog box to select the master chronology file to open. The Master Chronology Component Display (MCCD) screen (Figure 4-11) is very similar to the main CROSSDATE screen with some exceptions:

1. The MCCD screen can display all of the components of a master chronology. It is not restricted to three samples, but allows window overlap.
2. The menubar has options that apply only to the master chronology in the display.
3. Sample components displayed in the MCCD are fixed in time lag position. To change the position, the user must recompose the chronology.
4. The Master Chronology section contains a sample depth plot indicating the number of samples used in composing each year of the master chronology.
5. The MCCD displays either the skeleton plot or the time series plot for each component, but not both. The program displays the component's time series plot by default. The user may overlay the sample's time series plot on the master chronology time series plot.

The following control options are provided for the master chronology display:

- *Master Chronology* → *Master Chronology Info*: Displays master chronology header information (Figure 4-8).
- *Master Chronology* → *Color*: Changes the color of master chronology graphic elements.

The following control options are provided for each component displayed (*n* is the component number):

- *Component n* → *Display Time Series Plot*: Displays the component's time series plot.
- *Component n* → *Display Skeleton Plot*: Displays the component's skeleton plot.
- *Component n* → *Component Information*: Displays the sample component's header information panel.
- *Component n* → *Overlay Time Series Plot on Master Chronology*: Displays the component's time series plot overlaid on the master chronology's time series plot.
- *Component n* → *Hide Time Series Plot Overlay*: Removes the display of the sample's time series plot from the master chronology's time series plot.
- *Component n* → *Invert Skeleton Plot*: Displays the component's skeleton plot inverted.
- *Component n* → *Color*: Changes the color of the component's graphic elements.

The following menu options are provided for the MCCD screen:

- *File* → *Save Master Chronology*: Saves the current master chronology.
- *File* → *Save Master Chronology As*: Saves the current master chronology under a different file name.
- *File* → *Export Master Chronology*: Creates an ITRDB export file for the current master chronology.
- *File* → *Delete Master Chronology*: Deletes a master chronology previously saved to disk. If it is the current master chronology, the MCCD screen is also closed.
- *File* → *Close Master Chronology Display*: Closes the MCCD screen after asking if saving the file is desired.
- *Edit* → *Master Chronology Info*: Change master chronology header description information (Figure 4-8).
- *Edit* → *Master Chronology Dates*: Change the master chronology start or end date (Figure 4-7).
- *Edit* → *Master Chronology Skeleton Plot Cutoffs*: Change the absolute difference and first difference cut-off values for the master chronology skeleton plot.
- *Display* → *Display All Components*: Display all of the components of a master chronology (default).

- *Display → Display Sample Components Only*: Display only the components of a master chronology that are TREES data file samples.
- *Display → Display Master Chronologies Only*: Display only the components of a master chronology that are CROSSDATE master chronologies or ITRDB import chronologies.
- *Display → Display No Components*: Do not display the components of a master chronology.
- *Display → Component List Report*: Prepare a text report listing all of the components of the master chronology by level with summary attributes (Figure 4-13). Report screen allows the user to save the report to disk.
- *Window → Minimize*: Iconify the MCCD window.
- *Window → Close*: Closes the MCCD screen without asking if a saving the file is desired.

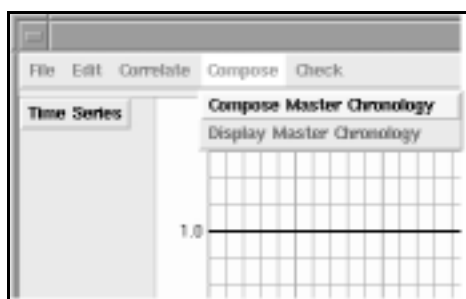


Figure 4-5 —CROSSDATE
Compose menu

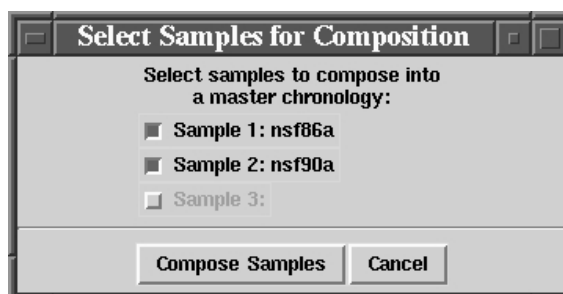


Figure 4-6 —CROSSDATE Compose
sample selection dialog box



Figure 4-7 — CROSSDATE set start or end year dialog and entry box

The following information is required for a master chronology:			
	Maximum Characters	Example	
Lead Investigator:	James B. Engle	63	Malcolm K. Hughes
Site ID:	AR2123	6	ABC123
Site Name:	A Mountain	52	Lake Tahoe
Species Code:	PCEN	4	PCEN
Species Name:	Pine	18	Engelmann Spruce
Elevation:	Pinus Sirevitis	6	1300M
Location (State/Country):	Arizona	13	Arizona
Latitude-Longitude:	+3202-11001	11	+5102-11303
Start Date:	05/14/00	8	05/03/00

Master Chronology Notes:

Save Notes: Test of master chronology composition
5-14-00

Reset Notes

Save Master Chronology Changes Cancel Master Chronology Changes

Figure 4-8 — CROSSDATE master chronology header information dialog box

Save Master Chronology File...

Directory: /win1/Notes/CROSSDATE_0000/headers/headers

- data
- data2
- input
- test
- XDATE
- fl.xml

File name: fl.xml Save

Files of type: CROSSDATE Master Chronology Files (*.xml) Cancel

Figure 4-9 — CROSSDATE master chronology file save dialog box

Display Master Chronology

Display Master Chronology Components?

Display Cancel

Figure 4-10 — CROSSDATE master chronology component display request dialog box

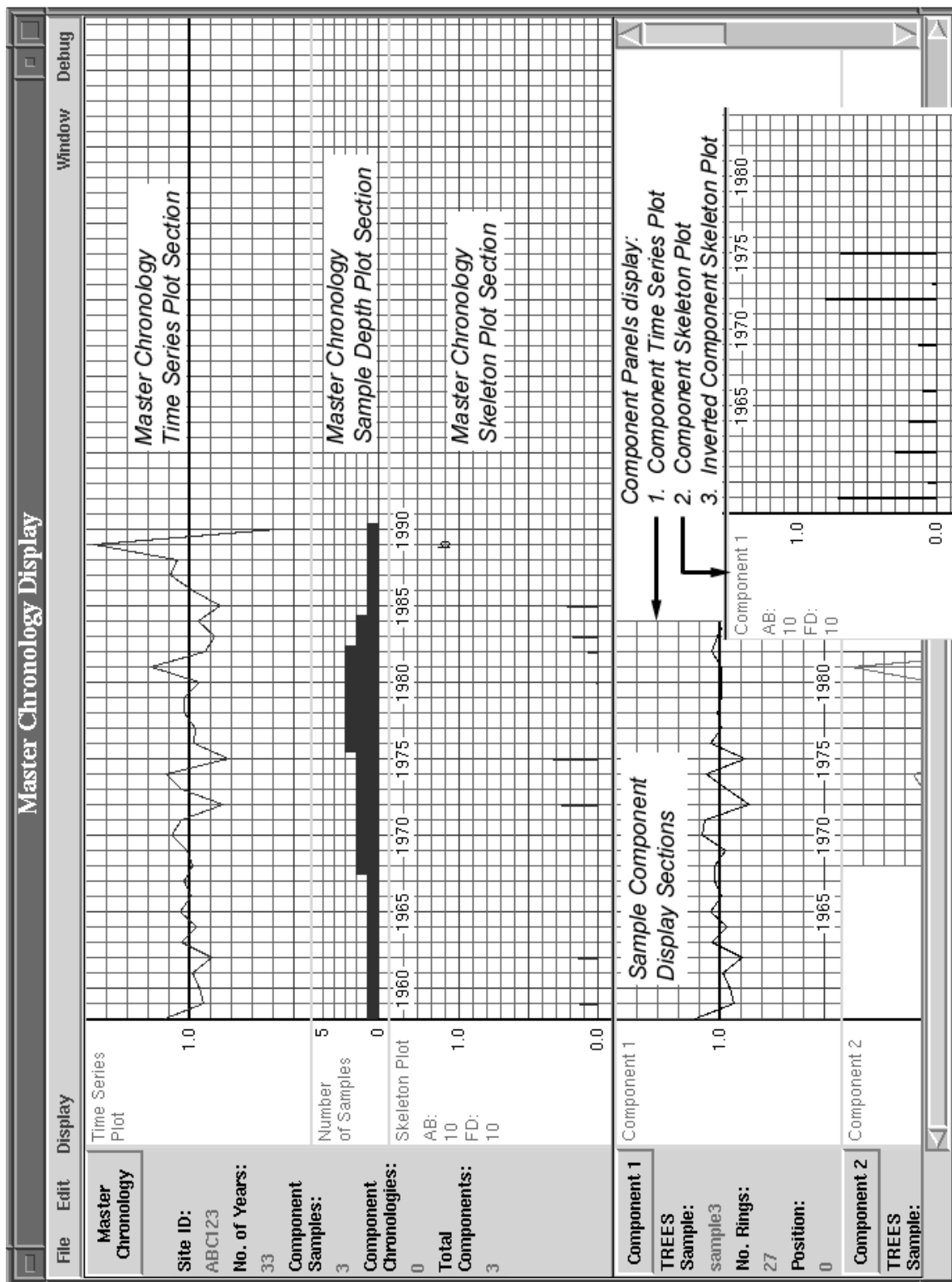


Figure 4-11 — CROSSDATE master chronology component display screen

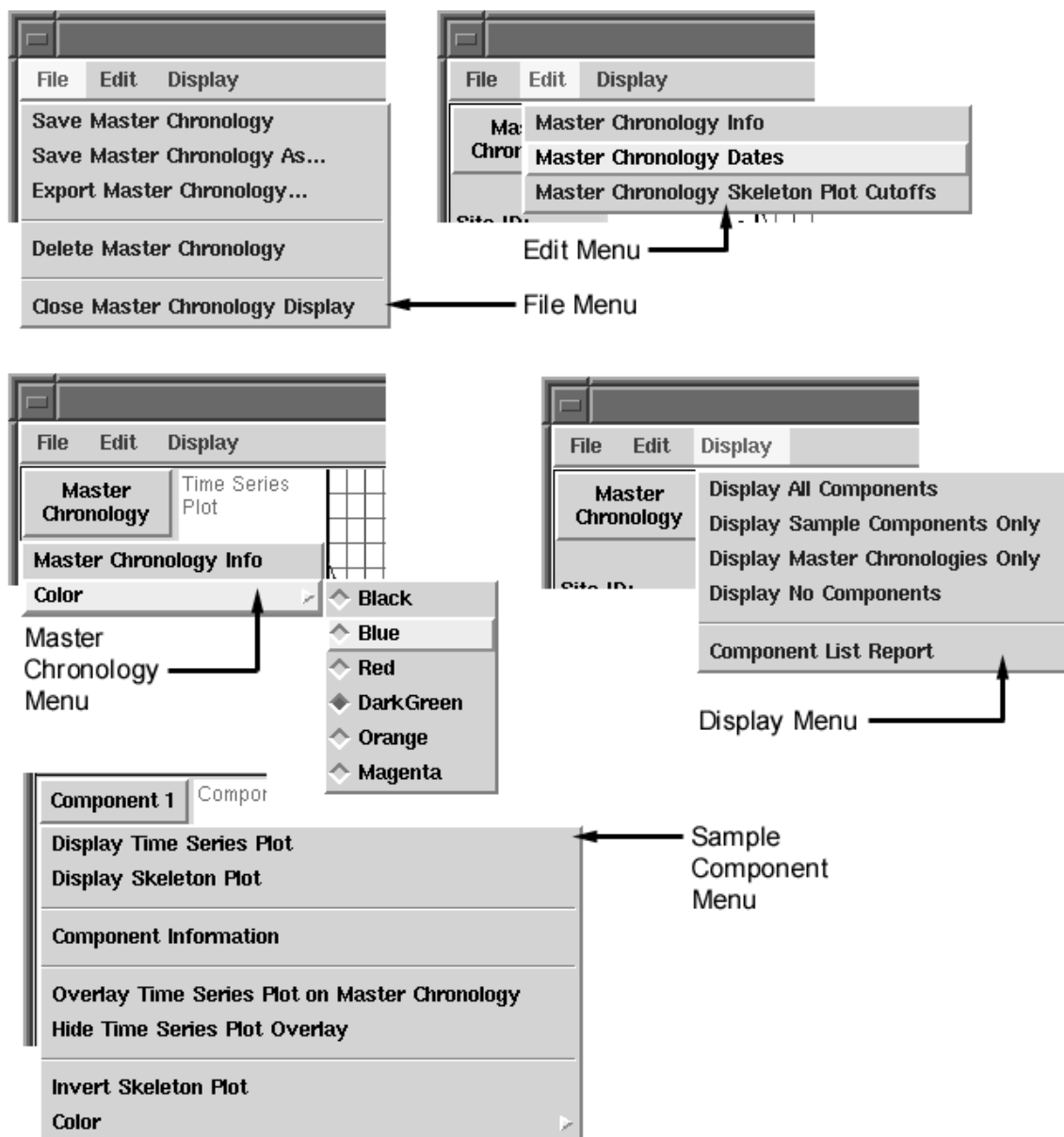


Figure 4-12 — CROSSDATE master chronology component display screen menus

Master Chronology Component Report

Component Report for Master Chronology - m1.xmf
Date: 05/23/00 Time:09:57AM

Master Chronology Information

Lead Investigator: James Engle
Site ID: WWW490
Site Name: Picacho Peak
Species Code: PCEN
Species Name: Pine
Elevation: 2500M
Location: Arizona
Latitude-Longitude: +3200-11000
Start Year: 1849
End Year: 2077
Number of Years: 229
File Start Date: 05/19/00
File Directory: /net/lewis/export/devel/engle/xdate/data3

Chronology Notes:

Master Chronology Component Information

No.	Level	File Name Directory	Type	Date/Time	Lag	Start Year	End Year	No. of Years
1	0	datafile6.tdf /net/lewis/export/devel/engle/xdate/data3	TDF Sample	04/05/00 11:14AM	6	1956	2030	74
2	0	m2.xmf /net/lewis/export/devel/engle/xdate/data3	XDATE Chron	05/19/00 11:10AM	0	1849	2021	173
3	1	datafile7.tdf /net/lewis/export/devel/engle/xdate/data3	TDF Sample	04/05/00 11:14AM	31	1932	2076	142
4	1	m4.xmf /net/lewis/export/devel/engle/xdate/data3	XDATE Chron	05/19/00 11:08AM	0	1849	1999	151

Figure 4-13 — CROSSDATE master chronology component list report

CHAPTER 5 — MASTER CHRONOLOGY QUALITY

CONTROL

5.1 COFECHA Method Overview

The process of creating master chronologies is often an overwhelming task due to the size of the data and the complexity of the process. The task becomes reasonable in size using computer-assisted methods. However, such methods simply automate the task and do not necessarily provide any perception of correctness. Dendrochronology experience is key in checking master chronologies and sample dating performed by automated methods. Tree-ring scientists often develop the ability to crossdate samples simply by visual inspection based on many years of experience with specific regions and species. Since the experts themselves are often unable to explain their ability to recognize and match tree-ring growth patterns, it is difficult to apply artificial intelligence methods to fully automate the crossdating process.

This is not to say that automated methods have no value in checking master chronology compositions. Many computer programs have been developed to perform this task — [19], [1][49], [12], [8], and [43] are examples of computer-assisted methods to crossdate tree-ring samples using various methods to check the results. One popular method for creating and checking master chronologies is program COFECHA [26,27,28]. Program COFECHA and many of the examples mentioned above were written before

graphical user interfaces were widely available. Program COFECHA is written in Fortran 77, takes raw tree-ring width measurements in text files as input, and produces a text report as output. According to [26], the main purpose of program COFECHA is the identification of data that should be re-examined for possible error. Philosophically, this means program COFECHA is a computer-assisted tool, rather than a fully automated process. The CROSSDATE program requires the capability to rigorously check a tentative chronology for correctness. Since program COFECHA matches the requirements for the CROSSDATE program so closely, many of its methods have been adapted to perform master chronology quality control checking in the CROSSDATE program. The key contribution of this work is that COFECHA methods are implemented in the CROSSDATE program with a highly visual graphic user interface.

Program COFECHA overcomes many problems with other crossdating programs [26]:

1. A known error-free chronology for the site is not required.
2. The series being tested may have locally false and/or absent rings.
3. Many series may be tested together.
4. A series may be reviewed in segments (blocks) for errors, rather than as a whole.

Program COFECHA takes dated tree-ring measurements as input. Each set of measurements is taken from a single wood sample and is referred to as a series. The

program applies a filter to remove low-frequency variation from the series measurements and then (optionally) applies a log transformation to enhance the effect of variability among small rings. Taking the mean of all the filtered and transformed series results in the master chronology. This is virtually identical to the CROSSDATE master chronology composition process up to this point.

Program COFECHA then tests each filtered and transformed series against the master chronology by removing the series under consideration from the master chronology composition. Correlation statistics are then computed between short segments of the series and corresponding segments of the master chronology. For each segment, the program checks that the correlation is positive and highly significant, and that it is higher when matched as dated than when shifted forward or back from that point. Statistical outliers in the component samples are noted after the filtering and transformation operations. Statistical outliers and segment correlation may only be determined where the master chronology has been composed from two or more series. The procedure is more useful when the master chronology has been composed from many more than two series.

Program COFECHA is intended to enhance data quality control by thoroughly examining all the sample components of a master chronology. The program is an independent confirmation of the accuracy of previous dating and measurement efforts. The program does not have a specific accept or reject criteria for correctly crossdating a

sample. Criteria to accept or reject a series or a portion of a series for inclusion in a master chronology is established by the dendrochronologist and is outside the scope of program COFECHA.

5.2 COFECHA Method Details

Program COFECHA takes sample input from ITRDB format raw data files (file extension .rwl — see Appendix A). Raw datafiles primarily contain dated tree-ring width measurements. Program COFECHA produces a printed output in (up to) eight sections:

- Part 1: Title page, options selected, summary, absent rings by series.
- Part 2: Histogram of time spans.
- Part 3: Master series with sample depth and absent rings by year.
- Part 4: Bar plot of master dating series.
- Part 5: Correlation by segment of each series with master series.
- Part 6: Potential problems: low correlation, divergent year-to-year changes, absent rings, outliers.
- Part 7: Descriptive statistics.
- Part 8: [Optional] Best Fit for undated ring width measurement series.

An example of program COFECHA output is included in Appendix E. The following discussion is a summary of [28] with emphasis on methods and procedures.

Part 1 of program COFECHA output provides a summary of the checking tasks

performed, absent rings noted by series, and parameter values set at run time. The format for raw data files allows for designating specific rings as absent or missing by specifying a tree-ring width value of 999. Program COFECHA allows the user to omit absent rings from correlation tasks. Part 2 shows the date period covered by each component series. Part 3 shows the normalized master series printed in tabular form. The normalized index value and sample depth is printed for each year of the master series. The user may optionally perform autoregressive modeling and a log transform on the master series to enhance the crossdating match.

Part 4 is a tabular bar plot of the normalized master series. Unlike a skeleton plot, the bar plot shows longer bars for wider rings. For each year of the master series, a line is drawn followed by a lower case character, an @, or an upper case character. A lower case letter indicates the ring is narrower than the mean, and an upper case letter indicates the ring is larger than the mean. The @ symbol indicates the ring is close to the mean. The position of the character in the alphabet is indicative of how many quarter standard deviations the relative ring width is from the mean. For example, b would indicate $-2x$ ($s/4$) and D would indicate $+4x$ ($s/4$), where s is the standard deviation of the master series ring width for a specific year.

Parts 5 and 6 represent the results of program COFECHA segment testing. Testing is performed by temporarily removing each component series from the master dating series to avoid biasing the results by partly testing a series against itself. Each

component series is split into ~50-year segments with ~25-year overlaps for testing. For example, suppose a 67-year series is dated from years 1921 to 1987. The series would be segmented as shown in Figure 5-1.

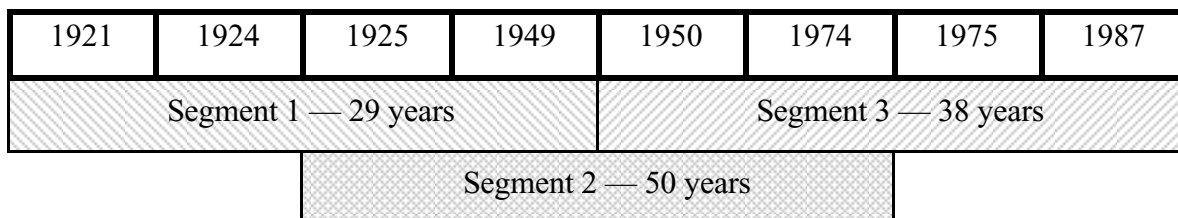


Figure 5-1 — Example series segmentation for program COFECHA

Program COFECHA allows the user to vary the segment length and overlap length parameters at run time. Each segment of each component series is tested against the corresponding portion of the master series for correlation. Part 5 is a tabular result of the correlation coefficient for each segment of each component series with the master series. Segments that correlate below a 99% confidence level are flagged with an A . Segments that correlate better at another lag position within a —10to +10 year window are flagged with a B .

Part 6 is a very detailed numeric description of potential problems with particular components in the master dating series. Only exceptional data is printed. The purpose of the detailed data is to direct the analyst to specific ring measurements that may be the source of problems found with the crossdating. Data is presented in the following five categories:

- A. If a better correlation is found within —10years to +10 years of a

segment's dating, then the correlation values for each lag position are printed with the highest value flagged.

- B. For the entire series and especially those segments where a better correlation is found at another lag position, the specific rings that raise or lower the correlation the most are identified along with their effect on the correlation.
- C. Segments are checked for year-to-year variations that are significantly in excess (± 4 standard deviations) of other sample components mean year-to-year variations for the same pair of years.
- D. Locally absent rings are listed and compared with other series.
- E. Rings that are statistical outliers are listed. A statistical outlier is a ring whose index value is $+3$ standard deviations larger than the mean or -4.5 standard deviations smaller than the mean.

Part 7 is a summary table of descriptive statistics of the ring measurement series processed. An analyst may optionally include a second data file with undated ring measurement series. Program COFECHA will attempt to indicate where the best fit occurs with the master dating series for the undated samples in part 8.

The key features of program COFECHA are its ability to perform multivariate statistical analysis and segment correlation. As a quality control tool, the objectives of program COFECHA are (1) to identify rings within a series component that are most

likely improperly measured and (2) to identify series components that are most likely improperly dated. These are the most common sources of error in crossdating. It is assumed that if the correlation between a series component and its master chronology is highly positive and significant, then the correlation of segments of the component series should also be highly positive and significant.

5.3 Master Chronology Quality Control Check Implementation

The CROSSDATE program implements most of the functionality of program COFECHA. The key difference between the programs is the highly visual nature of the CROSSDATE program. For example, parts 2, 3, and 4 of program COFECHA are combined into a single time series plot which may be displayed in basic or normalized units in the main screen of the CROSSDATE program. This is typical of the increase in effectiveness and reduction in data size that comes from a visual interface. The graphic user interface (GUI) also reduces the dependence on some numeric procedures for checking series components. For example, year-to-year ring width changes that are very different from the mean year-to-year ring width changes for other series are instantly obvious from an overlay plot without calculating any statistics.

The CROSSDATE program implements the key features of program COFECHA along with other checking procedures that are necessary within the CROSSDATE program framework. The following checks are performed for each sample component in

the CROSSDATE program:

- Date Checking.
 - Series component date alignment with the master chronology.
 - Series file date check.
- Identify ring widths that are statistical outliers.
- Series component overall correlation fit with the master chronology.
- Series component correlation fit with master chronology by segment.
 - Identify segments with better correlation fit at another lag position.
 - Identify segments with low correlation fit as dated.

Additionally, the correlation tool described in Chapter 3 is an interactive implementation of the Best Fit for undated series (Part 8 of program COFECHA).

5.3.1 Sample Component Date Checking

The CROSSDATE program does not depend on the user setting the start and end dates for a sample in the same way as program COFECHA. For program COFECHA start and end dates are used to determine sample-to-sample and sample-to-master series overlap positions. The CROSSDATE program explicitly keeps track of overlap positions and allows the user to set start or end dates at any time. For purposes of checking a master chronology, the dates of the chronology are assumed to be correct. Each sample component's start and end date is compared to its corresponding segment in

the master chronology for matching dates. CROSSDATE identifies sample components whose start and end dates are not aligned with the master chronology.

When designing a database, *normalization* is the process of assigning attributes to data objects, with the goal of eliminating data redundancies. This is the fundamental process that enables relational database linking. In CROSSDATE, the master chronology file provides references (e.g., *links*) to the actual files that contain the tree-ring data used to compose the master chronology.

The master chronology file stores only a small amount of information about each sample component that it references (See Figure 4-1 and Appendix D for attributes). One attribute stored in the master chronology file is the file date stamp for each sample component file. This is a way of identifying the version of the sample component used in composing the master chronology. Since it is possible to update a sample component file after a master chronology has been composed, the CROSSDATE program checks whether the file date of the sample component is equal to the value recorded in the master chronology file and notes discrepancies.

A discrepancy in a file date may or may not be a serious situation. For example, editing a sample (adding or removing tree-rings) in the TREES program will result in a new file date for the sample. This action could potentially invalidate its parent master chronology composition. This would be a serious situation that would likely require re-composing the master chronology. On the other hand, although simply changing an

analyst note or a start/end date will also result in a new file date for a sample, this would not be a serious situation, since there would be no effect on the master chronology. In any case, a file date check failure will alert the analyst to these situations.

5.3.2 Sample Component Statistical Outliers

The process of composing a master chronology is an averaging of several sample component index values at a particular year. Implicit in this process is the idea that the values are similar, otherwise a match might not really exist at the point in question. In looking for statistical outliers, it is assumed that the index values for each sample component for a given year are really the same, and that the small differences in numeric values are the result of an additive white gaussian noise process in measurement. The index values for a given year are then instances of a stationary process whose statistical mean and variance are unbiased estimators of the actual mean and variance. The arithmetic mean and standard deviation may then be calculated for each year of the master chronology using standard methods [47]. Comparing each sample index value with a threshold identifies a statistical outlier. For the CROSSDATE program, the threshold is ± 3 standard deviations from the mean for the given year.

5.3.3 Sample Component Overall Correlation Fit

The correlation between two sequences should be positive and highly significant

for a pattern match to be credible. However, strong correlation alone is not sufficient to establish a match. Like program COFECHA, the CROSSDATE program subtracts each component from the master chronology before performing component specific testing. This removes the bias that could result from partly comparing a sample component against itself. The functionality of the correlation tool described in Chapter 3 is re-used extensively for checking the overall correlation fit between the sample component and the remainder of the master chronology.

There are four possible outcomes to the overall correlation fit check:

1. The best fit between the sample and the master chronology is as dated and the correlation is highly positive and significant. (Sample OK as dated)
2. The best fit between the sample and the master chronology is as dated, but the correlation is below the threshold for significance. (Low correlation)
3. The best fit between the sample and the master chronology does not occur at the position dated within -10 to $+10$ years. (Better fit found)
4. The sample component may be a key component in the master chronology composition. Removing it from the chronology leaves nothing to correlate with. (No data - unable to correlate)

The CROSSDATE program identifies these situations after performing an overall correlation fit check. The CROSSDATE program will not attempt to perform correlation fit checking if the corresponding portion of the master chronology containing the sample

component has any gaps due to missing data. If the master chronology has any negative index values after subtracting the sample component, no correlation fit checking will be performed. This situation is likely due to a ring addition/deletion in the sample component file since the master chronology was composed.

5.3.4 Sample Component Segment Correlation Fit

Segment correlation is performed in the same manner as described above for the overall sample component. The sample is segmented as illustrated in Figure 5-1, and the results are identified by one of the four possible outcomes. According to [27], 50-year segment lengths provide sufficient degrees of freedom so that there are few segments where very high or very low correlation occurs by chance, and the chance correlation at high significance is low enough to prevent false alarms. Yet 50 years is short enough to allow detection of dating errors of a few years in length. In segmenting a sample component, it is assumed that the component fits the master chronology very well, but there may be measurement errors that exist. Segmenting the component allows for identifying segments with possible measurement errors. Better fit or low correlation flags alert the dendrochronologist to the type of error that might exist in the segment (e.g., missing ring, false ring, or improper measurement).

5.4 User Interaction — Check Tool

User interaction with the Check tool begins by selecting the master chronology to check as shown in Figure 5-2. A progress gauge indicates the percentage of completion of the component check task. After the sample components have been checked the results are displayed in the master chronology QC check display shown in Figure 5-3. The master chronology QC check display is similar to the master chronology component display (MCCD) described in Chapter 4. The master chronology is displayed in the upper half of the screen and each of its sample components may be scrolled into view in the lower half of the screen. Menu button controls are provided for the master chronology and each of the sample components as shown in Figure 5-5. The functions are described in section 4.5 and shown in Figure 4-12 with the exception that sample components will additionally display a QC check plot (default). However, the analyst may choose instead to display a time series plot, a skeleton plot, or an inverted skeleton plot in the same panel. The main menu command options are significantly different between the QC check display and the MCCD.

Component display is limited to TREES data file samples. Checking is not performed for master chronology components that are also master chronologies or imported chronologies. For purposes of checking a master chronology, intermediate chronologies are simply place holders in the hierarchy. Actual samples are the only components relevant for checking. Similarly, it is not possible to unpack an imported

chronology into its component samples, so no attempt is made to perform checking on an imported chronology component.

Imported chronologies have an additional impact on statistical outlier tests for a sample component. Since individual sample values are unknown within the imported chronology, it must be assumed that each component index value is equal to the mean index value of the chronology. This effectively causes the standard deviation to be zero for the sample components in the import chronology. The result of this reduction in standard deviation will be more false positive statistical outliers flagged in sample component checking. Fortunately, the error is an annoyance rather than a complete loss in validity.

The QC check plot panel depends significantly on color to convey the results of the check (Figure 5-4). Red is used to indicate when a problem exists. Green is used to indicate when no problem was found. Blue is used when a specific check could not be performed. A pale yellow background with light gray gridlines is used to indicate the time span of the component series. Medium gridlines are used for five-year gridlines, and heavy gridlines are used for twenty-five year gridlines to delineate segment time boundaries. The color selections may be reconfigured inside the program source code in case the color selections are not satisfactory to the analyst.

The QC check plot panel includes a descriptor label and bar plot area for date checks, chronology years, statistical outliers, overall fit, better segment fit, and low

segment correlation. To draw the analyst's attention to problem areas, the descriptor label for each section is drawn in red, green, or blue as indicated by the result of the check. The background of the date check boxes is changed to indicate the result of the check (e.g., pass or fail). For the date alignment check, the background of the Chron[ology] Year bar is changed to match the result of the check. Index values in the sample component that are statistical outliers are flagged with a red marker on the Outlier bar plot. Positions where the sample component has a better overall correlation fit with the master chronology are also flagged with a red marker in the Overall Fit bar plot area. The Overall Fit bar also includes text information for the overall correlation coefficient and its significance.

Segment boundaries for the sample component are shown in the bar plot areas for Better Segment Fit and Low Segment Correlation. Two horizontally adjacent plot bars are used for these sections to show the segment overlap. The background color of the segment bar is changed to reflect the result of the segment check. If there is a better fit found for the segment, the position of the best fit, its correlation coefficient, and its significance are displayed as text information inside the segment bar. The correlation of the segment with the corresponding segment in the master chronology along with its significance is displayed inside the Low Correlation bar independent of the result of the low correlation check. The significance threshold is used for determining the result of the low correlation check. End segments less than ten years in length are not tested for

correlation with the master chronology.

The following menu options are provided for the master chronology QC check display screen:

- *QC Display* → *Component QC Check Report*: Display a text report listing numeric results for the sample component correlation and statistical checks (Figure 5-6). Report screen allows the user to save the report to disk.
- *QC Display* → *Component List Report*: Display a text report listing all of the components of the master chronology by level with summary attributes (Figure 4-13). Report screen allows the user to save the report to disk.
- *QC Display* → *Close Window*: Closes the master chronology QC check display screen.
- *Edit* → *Change Significance Level and Recheck Components*: Allows the user to change the value of the significance test threshold and recheck all components against the new value. The value must be in the non-inclusive range of (0,1).
- *Window* → *Minimize*: Iconify the master chronology QC check display window.
- *Window* → *Close*: Closes the master chronology QC check display screen.

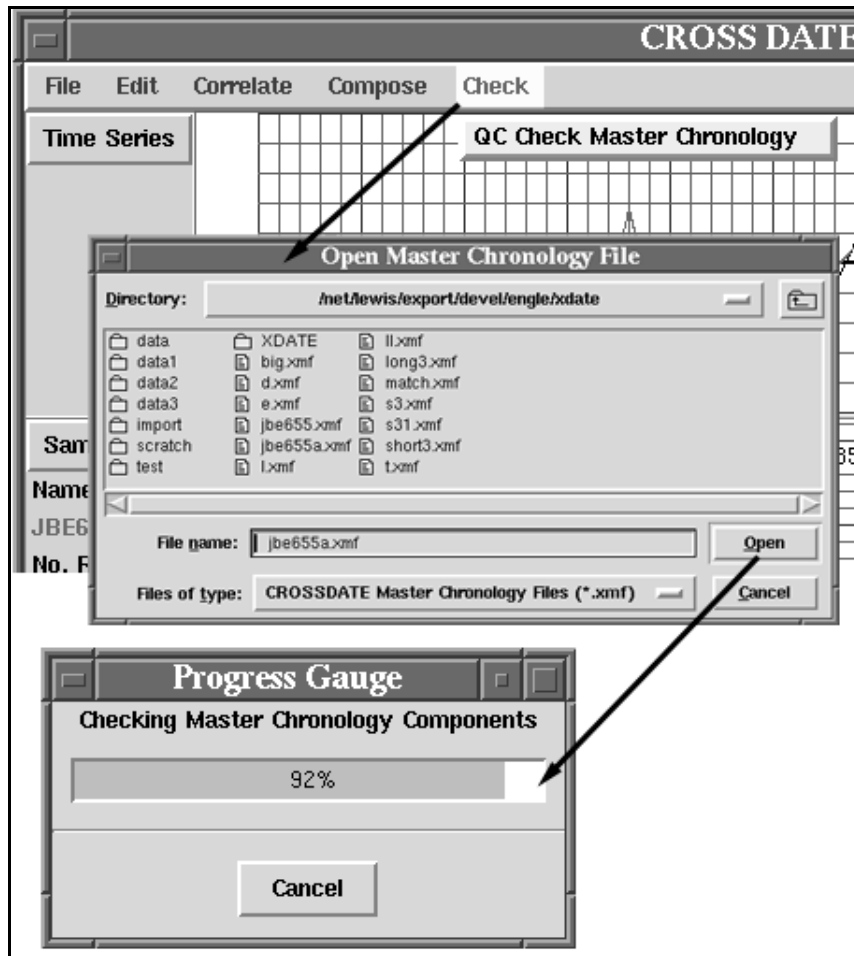


Figure 5-2 — CROSSDATE master chronology QC check process

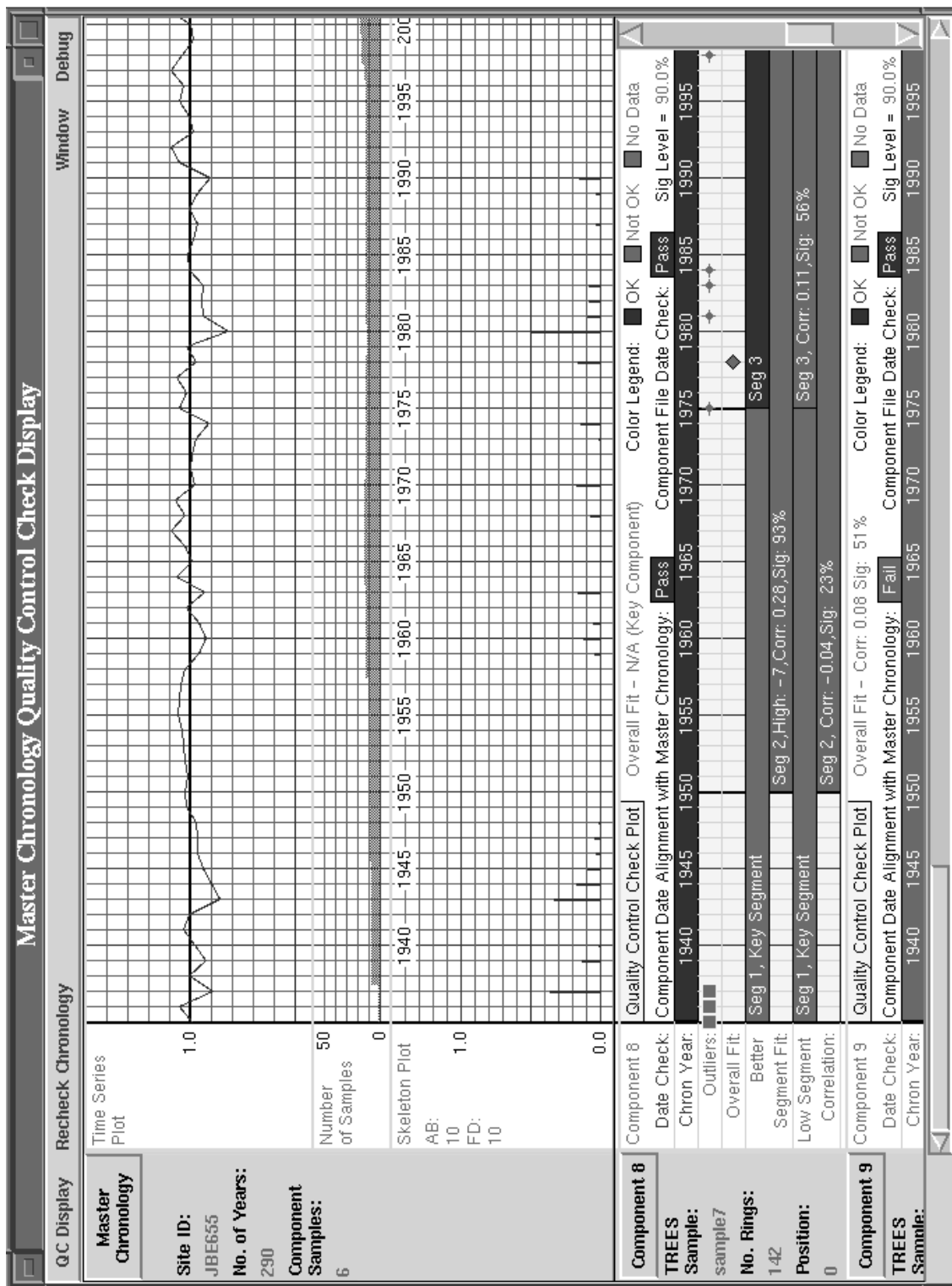


Figure 5-3 — CROSSDATE master chronology QC check display screen

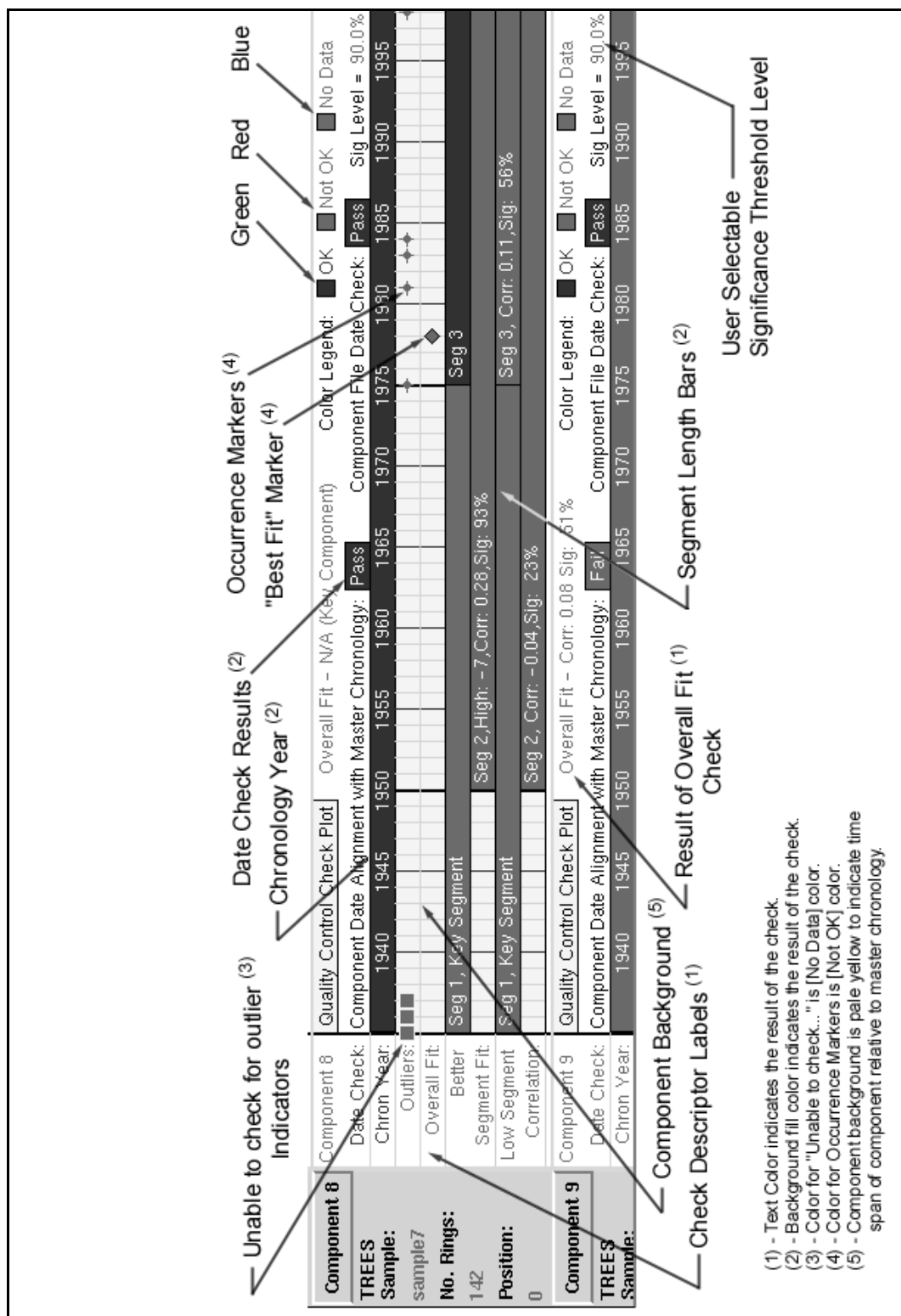


Figure 5-4 — CROSSDATE sample component QC check panel

- (1) - Text Color indicates the result of the check.
- (2) - Background fill color indicates the result of the check.
- (3) - Color for "Unable to check..." is [No Data] color.
- (4) - Color for Occurrence Markers is [Not OK] color.
- (5) - Component background is pale yellow to indicate time span of component relative to master chronology.

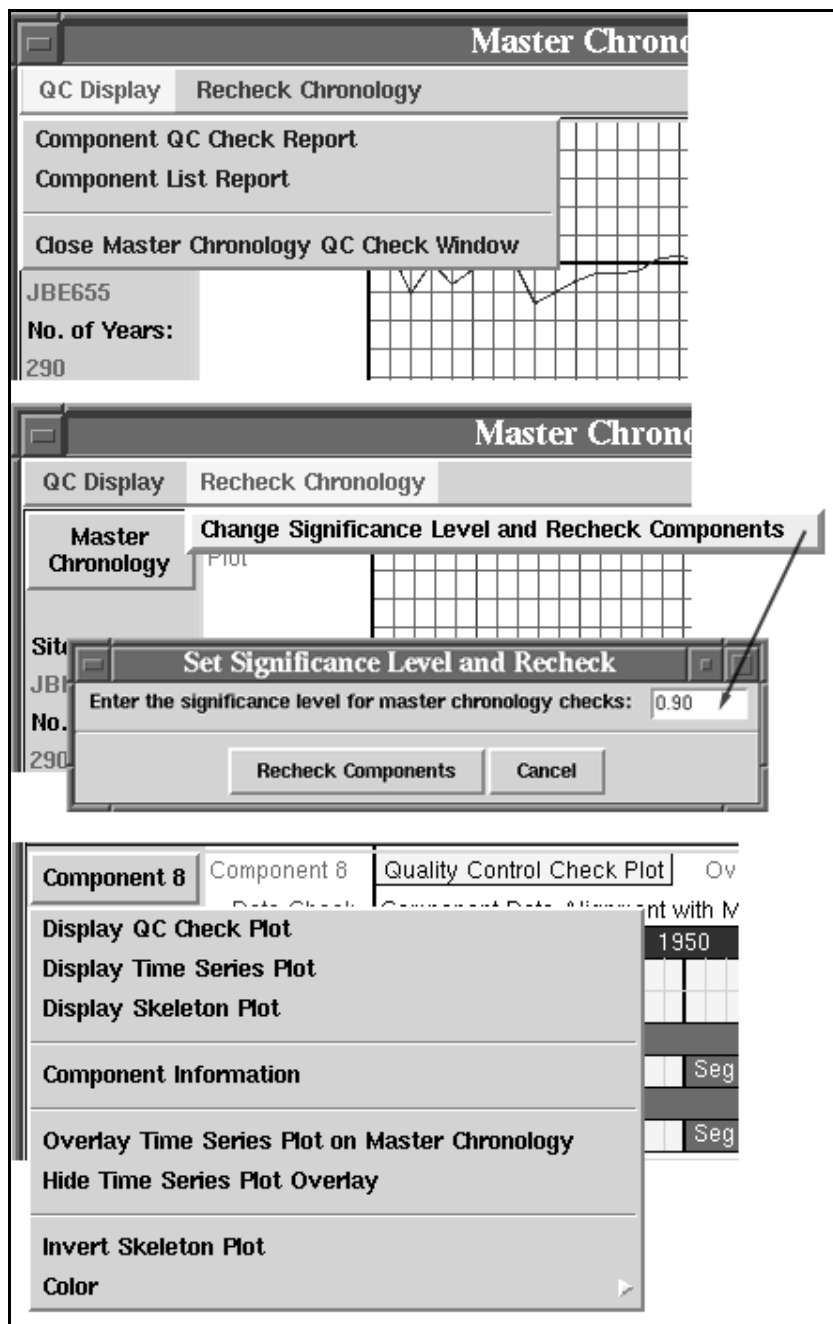


Figure 5-5 — CROSSDATE master chronology QC check menu options

```

Component QC Check Report for Master Chronology - t.xmlf
Date: 05/22/00 Time:10:00AM

Master Chronology Information
-----

(same as Component Check Report - See Figure 4-13)

Component QC Check Information -- Significance Level = 90%

COMPONENT 3 -
File Name:      nsf96a-4.tdf
Directory:      /net/lewis/export/devel/engle/xdate/data2
Lag Position:   22      Chronology Start Year: 1838 End Year:      1986
                  Sample Start Year:      0 End Year:      104

DATE CHECK:
Sample Dates Aligned with Master Chronology: Fail
Sample Component is offset: -1838 years from the chronology
File Date Check: Fail
Expected: 05/22/00 09:34:25 Actual: 05/22/00 09:54:07

OUTLIERS:
(Outlier Dates are SAMPLE Years)
Sample      Sample      Sample      Sample      Sample      Sample
Year      SD      Year      SD      Year      SD      Year      SD      Year      SD      Year      SD
-----
1857 +3.10  1884 -3.08  1901 +3.33  1904 -3.20
4 outliers found in Component 3

OVERALL FIT:
(Overall Fit Dates are CHRONOLOGY Years)
Chron      Corr      t      Sig
Year      Coeff      Value      %
-----
As Dated:  1860  0.250  2.53  99
Best Overall Fit: 1852  0.269  2.78  99

Better Overall Fit Positions than as dated:
Chron      Corr      t      Sig      Chron      Corr      t      Sig
Year      Coeff      Value      %      Year      Coeff      Value      %
-----
1852  0.269  2.78  99  1882  0.251  2.60  99
2 positions with better fit found for Component 3

SEGMENT FIT:
(Segment Fit Dates are SAMPLE Years)

Sample Sample *---As Dated---* *---Better Fit---*
Seg Start End Corr t Sig Lag Corr t Sig
No Year Year Coeff Value % Pos Coeff Value %
-----
1 1860 1899 0.188 1.17 75 Low Correlation as dated
   40 yrs Lag Positions with better fit:
2 1875 1924 0.106 0.74 54 -10 0.348 1.92 94 Best Fit
   50 yrs Low Correlation, best fit as dated
3 1900 1949 0.274 1.95 94
   50 yrs Lag Positions with better fit:
4 1925 1964 0.403 2.60 99 -8 0.399 2.67 99 Best Fit
   40 yrs Best Fit as dated

```

Figure 5-6 — CROSSDATE master chronology QC check report example

CHAPTER 6 — SYSTEM INTEGRATION AND TESTING

6.1 Module Testing and Integration

The CROSSDATE program consists of approximately 170 Tcl/tk procedures and 30 C functions. Each procedure or function may be considered a module for this discussion. The initial release of the source is over 10,000 lines of code contained in 20 files. For comparison, this is roughly one-half the size of the TREES program and twice the size of program COFECHA. As discussed in Chapter 2, the CROSSDATE program was developed iteratively using rapid prototyping. Development proceeded sequentially over the following four major areas:

1. Basic input/output and graphic user interface (Chapter 2).
2. Correlation tool (Chapter 3).
3. Master chronology composition tool (Chapter 4).
4. Master chronology quality control check tool (Chapter 5).

This task sequence matches the steps a dendrochronologist would take in dating samples and composing master chronologies. The development tasks of (1) requirements refinement and definition, (2) software design, (3) implementation and test, and (4) final review were applied to each of the four major areas iteratively. Module testing was an integral part of implementation throughout the development process.

The source code for the CROSSDATE program is split into portions that focus

either on data display or on data management. Data display is accomplished entirely using Tcl/tk procedures. Data management is split between Tcl/tk procedures and C

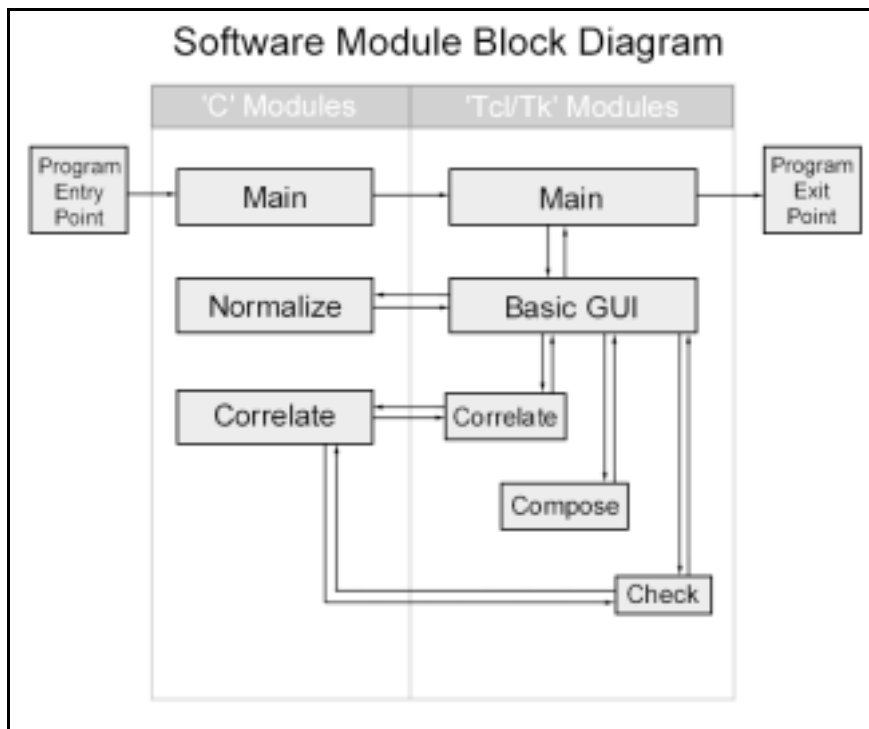


Figure 6-1, CROSSDATE software module block diagram

functions. Figure 6-1 delineates the flow of control between modules programmed in the C language and modules written as Tcl/Tk scripts. Since Tcl/tk is an interpreted language that does not require compilation, testing and debugging are comparatively quick tasks, especially for data display functions. Feedback is highly visual and practically instantaneous upon completion of a module. The Tcl/tk interpreter provides excellent diagnostics that aid in debugging code and making corrections. This instantaneous feedback is quite useful in trying different approaches for graphic user interface (GUI)

design. Since the CROSSDATE program provides a highly visual environment, it was tested during development on a wide variety of graphic hardware for proper display, for example, 8 and 24 bit video on Sun workstations and PCI and AGP video cards on PCs using Linux.

Each data management module typically performs functions related to input/output (I/O) or numeric processing. In addition to data acquisition and storage, I/O modules were designed to emphasize trapping error conditions and preventing their propagation throughout the program. Data sets were specifically created to test each conceivable error condition. A Debug Window tool was created for the purpose of observing module I/O behavior during runtime. Each I/O module was tested for proper data acquisition and storage using the Debug Window during development.

Modules that perform numeric processing functions were tested in several ways. First, black box testing was used to test for correct results. Black box testing means that a specific output is expected from a specific input independent of the process inside the black box. For the CROSSDATE program, the results of numeric procedures were compared with the results of known valid MATLAB functions for correctness. Next, white box testing was used to test for correct module operation. White box testing focuses on the operation of the module. This method was used extensively to test the various execution paths within the numeric processing modules. Finally, performance testing was used for testing memory size and timing considerations.

Module and system integration can be a major undertaking for many software applications that involve a team of developers. Since the CROSSDATE program was conceived and implemented by one author, module integration was not a major issue. Testing the overall operation of each major portion of the program was easily accomplished using the same methods as in module testing. Furthermore, the dependencies between the major portions of the CROSSDATE program are not highly complex.

6.2 System Validation and Verification

In software development, system verification is the process of testing the system against its specification. System validation is the process of confirming that the system matches the user's expectations [37]. The CROSSDATE program has been designed to perform each of the requirements listed in Section 2.1, and it has also been tested to perform each of the requirements listed. In this sense the program has been verified in terms of its specification. However, this is not a complete response to the question of system validation.

The requirements for the CROSSDATE program were prepared from observations of the manual task performed by dendrochronologists in crossdating tree-ring samples. An interactive, visual tool for performing crossdating tasks from digital images of tree-ring samples has not been done before in the field of dendrochronology. It

is therefore difficult to say exactly what user expectations would be for such a program. This is often the case in software application development.

For example, the notion of a computerized spreadsheet was developed in exactly the same way. Individuals performed numeric analysis for many years manually on large pieces of paper known as spreadsheets using adding machines and calculators. The dependencies among the numbers made it a tedious task to make changes. A computerized spreadsheet significantly reduced the effort of making changes. Computerized spreadsheets are currently taken for granted as a numeric tool, and users have very specific expectations of what these products should do. However, it would have been quite difficult to answer what user expectations of a computerized spreadsheet would be before having first used one.

The CROSSDATE program will be validated ultimately by user interaction in a similar fashion. One measure of a successful system validation is the level of interest generated by the topic of improving the product. If the system is found to be useful by the dendrochronology community in reducing the effort of crossdating tree-ring samples, there will be continuing discussion regarding ways in which to improve it. If the system is not viewed as useful, there will be more interest in generating alternative systems to perform the task. The question of system validation can only be answered over time.

6.3 User Help and Assistance

An informal and highly un-scientific method of testing new software usefulness has been the five-minute test . If a user is able to do something productive with a new software product within five minutes without reading any documentation, the product is said to pass the five-minute test. It is always a software designer s goal for the operation of the product to be intuitively obvious , as suggested by the five-minute test. However, it is very poor practice to assume that the application is so easy to use that no help or documentation is needed.

Much of the operation and philosophy of the CROSSDATE program described in this thesis has been translated into a user manual as an HTML (HyperText Markup Language) document that accompanies the software source code. This will require the user s computer to have an HTML viewer program that is separate from the CROSSDATE program. Such programs are freely available as internet browsers from various companies. Figure 6-2 is an illustration of the user interaction sequence to invoke on-line help for the CROSSDATE program. The user manual is written under the assumption that the reader has a background in dendrochronology and basic UNIX workstation usage. The complete source code is also distributed with the CROSSDATE application. Significant attention to source code documentation has been provided to promote future modifications and enhancements.

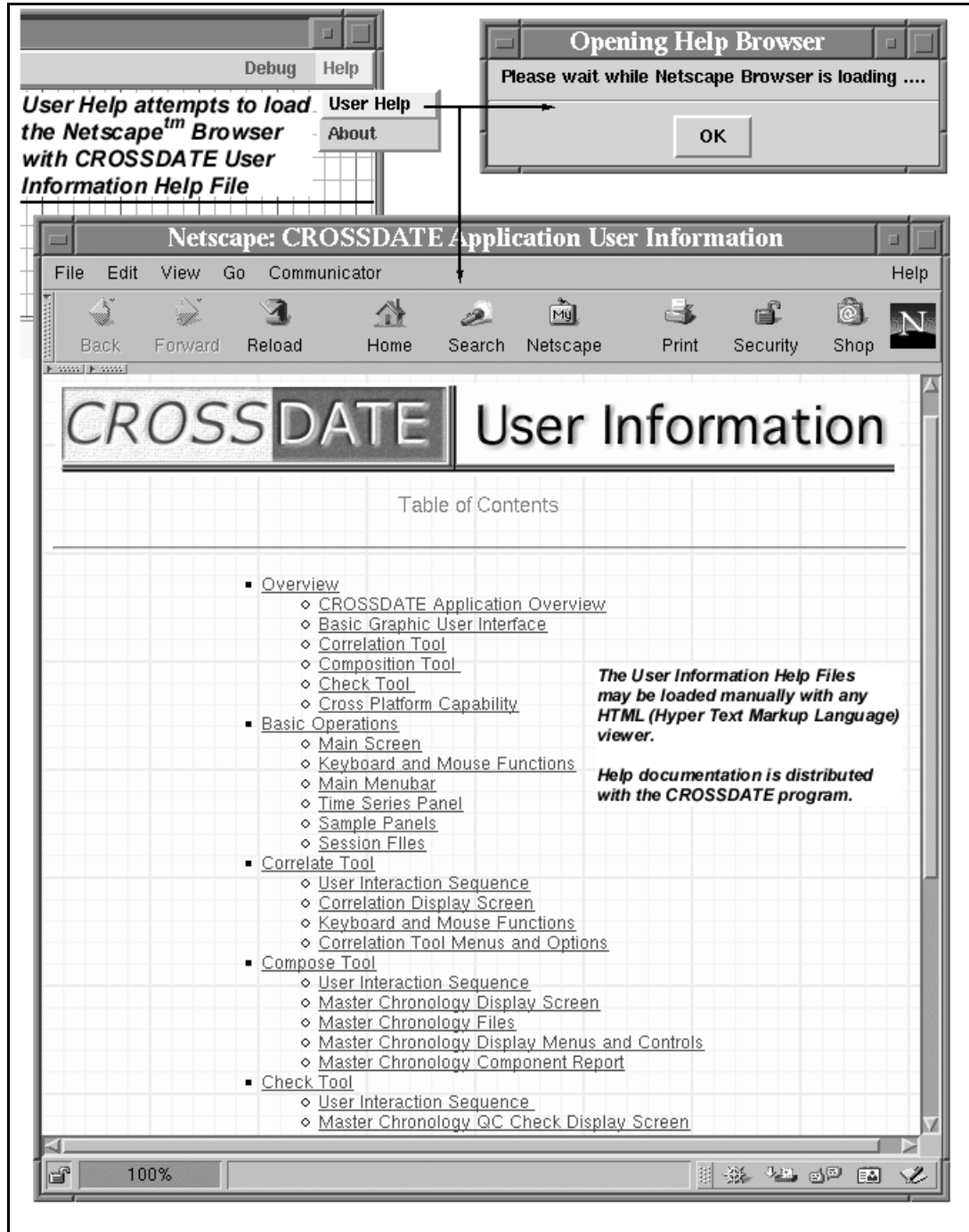


Figure 6-2, CROSSDATE user help interaction sequence

6.4 Software Configuration Control

In order to test or modify the CROSSDATE program, it is first necessary to know what version is being discussed. Testing and modification cannot be done sensibly unless the version of each item relevant to the test or modification is known and is confirmed to be the correct version for the task. The CROSSDATE program has been developed iteratively and has thus gone through several changes prior to initial release. It is reasonable to assume that the program will go through many more revisions during its life cycle.

Changes are therefore an integral part of the development process and require a minimum level of record keeping for change traceability. Traceability is the ability to establish an audit trail of relevant information [37]. In the context of testing, this can be interpreted as the ability to trace an error to its source. In the context of program modification, this can be interpreted as the ability to trace a modification to its requirement specification.

The CROSSDATE program is under software configuration control using the CVS (Concurrent Versions System) repository from the GNU project of the Free Software Foundation. CVS maintains a history of all source code changes, time stamps each change, and records the user name of the person who made it. When a change is being committed to the repository, an opportunity to add a brief text note explaining the change is presented.

The CVS repository helps developers answer the following questions:

- § Who made a given change?
- § When did they make it?
- § Why did they make it?
- § What other changes did they make at the same time?

CVS is widely used by developers for tracking source code changes.

CHAPTER 7 — SUMMARY AND CONCLUSIONS

7.1 Summary

Dendrochronology provides critical support to various areas of scientific research in establishing occurrence dates of environmental events over very long periods of time. Like many scientific research areas, a large portion of an experienced dendrochronologist's effort is spent tediously measuring and comparing tree-ring samples manually. The TREES program was created to improve the accuracy and the efficiency of the measurement activity using image processing and computer vision techniques. This thesis describes the CROSSDATE program, which was created to improve the accuracy and the efficiency of the comparison activity after measurements have been completed.

The CROSSDATE program is an interactive graphic application to aid dendrochronologists in:

- Establishing dating for tree-ring samples.
- Composing master chronologies from tree-ring samples.

The unique contribution of the CROSSDATE program is the graphic framework for performing tree-ring sample comparison activities. The CROSSDATE program implements methods familiar to dendrochronologists for performing the tasks of correlation, composition, and quality control checking in a computer-assisted environment.

Up to three tree-ring samples measured by the TREES program may be viewed and compared on the basic graphic screen at a time. Tree-ring samples may consist of hundreds of ring width measurements. Matching large samples for dating can be an overwhelming and time-consuming task. The correlation tool provides a quick and effective numeric tool for finding the best match position between two tree-ring series. The tool is based on statistical covariance theory and is efficiently implemented using Fast Fourier Transforms (FFT).

Once the samples have been pattern matched in time, a composition tool is provided to create a master chronology from component samples or other chronology files. The three-sample composition limit is bypassed through the use of hierarchical composition. Using a virtual tree data structure for sample components, there is practically no limit to the number of components that may be used to compose a master chronology. The CROSSDATE program imports and exports standard ITRDB format chronology files.

Dendrochronologists have used program COFECHA for many years as a tool for validating a master chronology from its components. The CROSSDATE program implements most of the functionality of program COFECHA in a highly visual, graphic environment. The fit of a sample component in the master chronology is tested by temporarily removing it from the chronology and checking its correlation with the remainder of the chronology. This is performed for the overall sample versus the overall

chronology, as well as 50-year segments of the sample versus the corresponding segment of the master chronology. The master chronology is also checked year-by-year for sample component ring widths that are statistical outliers. The CROSSDATE program additionally performs sample and file date checking for each sample component of the master chronology. The results of all tests are displayed for the user on screen using color to convey the results of each test. The analyst may also produce a text report detailing the results of each sample component test.

The following performance was observed using a Sun workstation (Ultra 10, 350Mhz, 256Mb RAM) as a benchmark for testing:

- Initial RAM requirement - 4 MB
- Correlate two 400-year tree-ring samples - 600 msec.
- Correlate one 7,980-year sample with a 400-year sample - 3000 msec.
- QC Check a 16 sample master chronology - 2950 msec and increased RAM required to 10 MB.

The CROSSDATE program has been developed for Sun Solaris and Linux operating systems on PC and UNIX Workstation hardware platforms. User help and assistance is provided through hypertext help files that accompany the program distribution. Information about the TREES and CROSSDATE programs may be found on the internet at the locations shown in Table 7-1.

Topic	Location
TREES Project	http://www.ece.arizona.edu/~dial/trees
TREES User Manual	http://www.ece.arizona.edu/~dial/treesdoc
CROSSDATE User Manual	http://www.ece.arizona.edu/~dial/xdatedoc

Table 7-1, Internet locations for TREES project references

7.2 Conclusions

The dendrochronology tasks of sample acquisition, measurement, and dating through growth pattern matching have changed little from their scientific origins. However, the tools used to accomplish these tasks have changed greatly during this time. As companion programs, TREES and CROSSDATE provide a computer-assisted environment for performing dendrochronology analysis. But, more importantly, these programs provide a digital setting for evaluating, implementing and utilizing new tools for accomplishing dendrochronology tasks. The object-oriented modular design of the TREES and CROSSDATE programs allow for seamlessly replacing key components as new methods become available without a major change in system architecture.

The CROSSDATE program uses methods that are very familiar to dendrochronologists for comparing tree-ring samples and composing master chronologies.

The program's graphic environment seeks to imitate manual tree-ring series comparison tools as much as possible. The effectiveness of these methods has been proven over time as the scientific basis of dendrochronology. In its initial release, dendrochronologists appear very comfortable with the look, feel, and operation of the CROSSDATE program.

At a minimum, it may be concluded that the CROSSDATE system satisfies its sample comparison and chronology composition objectives. By freeing tree-ring scientists from the tedium of tree-ring measurement and comparison, it is reasonable to expect the development of new and improved dendrochronology methods to perform these tasks. When this happens, the framework for implementing and displaying new procedures will already be in place in the CROSSDATE program.

7.3 Recommendations for Future Improvements

The CROSSDATE program is the result of a future improvement recommendation from the TREES program [6,7,34]. The following areas are under consideration for future improvements to the CROSSDATE program:

- Number of Composition Samples

The composition process is currently limited to viewing and composing three samples at a time into a master chronology. This results in a large number of intermediate chronologies to keep track of during composition for even a moderate number of sample components. A future modification might include revising the basic GUI to allow for more than three samples to be viewed and displayed at one time.

- Interactive Links between the CROSSDATE and TREES programs

The TREES and CROSSDATE programs are designed to complement each other and operate on the same base data at the same time. However, there is no interactive link between the programs at this point. Possible future modifications might include displaying a master chronology time series plot and/or skeleton plot inside the TREES program for dating purposes, and an interactive link that will display a particular tree ring image from TREES using a selection in the CROSSDATE program.

- Graphic Display Performance

One of the interesting results of performance testing on the CROSSDATE program was that building the display panel for the results of a test takes much more time than performing the actual numeric test. As mentioned in Section 7.1, the correlation math procedure between a 7,980-year sample and a 400-year sample took 3000 msec to perform, but roughly another minute was required to complete the graphic display. This is, of course, an extreme example. The graphic lag for tree-ring samples smaller than 1,000 years in length is not noticeable on the workstation platform.

There are current contributions to Tcl/Tk programming that allow scripts to be compiled to improve execution time. There is a possibility that compiling the CROSSDATE Tcl/Tk code might improve graphic display time. This could be investigated as a future improvement.

APPENDIX A — ITRDB DATA FILE FORMATS

The following data file format information has been provided by the International Tree-Ring Data Bank (ITRDB) and is reproduced here with permission. The ITRDB is maintained by the NOAA Paleoclimatology Program and World Data Center-A for Paleoclimatology. Please see <http://www.ngdc.noaa.gov/paleo/treering.html> for further information.

A.1 Format for Tree-Ring Data Files

Raw Data Files (File Extension .rwl)

These are measurements in increments of .01mm of the thickness of tree ring width for each year. Each file consists of all the measurements for a given site. Fifty or more Core ID numbers and data series may comprise one (site) file. Missing ring value code is 999. The 10 values following the decade are the 10 annual measurements for the 10 years of that decade. First and last decade rows for each core may contain less than 10 values. The format is:

Core ID Number	columns 1-6
Decade	columns 9-12
Data Values	columns 13-73, 6 columns/measurement, 10(I6)
Optional Site ID	columns 74-78

A.2 Format for Processed Data Files

Site Chronologies (File Extension .crn)

These are the normalized averages from a stand of trees, representing percentage of mean growth observed for each year over the entire stand. Site Chronologies are used in climate analysis, and it is these values which are displayed by the ITRDB software. Data are stored as 4-digit numbers, with a value of 1000 representing mean growth and 900 representing 90% of mean growth, etc. There is only one time series per file, in contrast

to the raw data files. Missing ring value code is 9990. Site information is stored in the first 3 records of the file.

Format for chronology header records:

Record #1:	Site ID	columns 1-6	6 characters	
	Site Name	columns 10-61	52 characters	
	Species Code	columns 62-65	4 characters	
	(Optional) ID#'s	columns 66-		
Record #2:	Site ID	columns 1-6	6 characters	
	State or Country	columns 10-22	13 characters	
	Species	columns 23-40	18 characters	
	Elevation	columns 41-46	6 characters	
	Lat-Long	columns 47-57	11 characters	
	1st & last Year	columns 68-76	9 characters	(*)
Record #3:	Site ID	columns 1-6	6 characters	
	Lead Investigator	columns 10-72	63 characters	
	Comp. date	columns 73-80	8 characters	

Chronology Data, Records 4 and up

Site ID#	column 1-6	6 characters	(*)
Decade	column 7-10	4 characters	
Index Value-Sample Number*			
pairs of values	column 11-80	10(I4+I3) characters	
TRL ID# (Optional)	column 82-88	7 characters	

*Index Value-Sample Number Pair consists of:

Index Values, columns 11-14,18-21,25-28,32-35,etc

Number of samples used in calculating chronology, columns 15-17,22-24,29-31,36-38,etc

Chronology Statistics, Last Record (Optional):

Site ID#	column 1-6	6 characters
Number of Years	column 8-10	3 characters
First Order Autocorrelation	column 13-16	4 characters
Standard Deviation	column 19-22	4 characters
Mean Sensitivity	column 25-28	4 characters

Mean Index Value	column 29-35	7 characters
Sum of Indices	column 37-44	8 characters
Sum of Squares of Indices	column 46-53	8 characters
Max# of series	column 62-63	2 characters

(*) — BC Dates:

A minus sign is used for BC dates. The presence of a minus sign in cols. 67 and 72 for the first or last year (Record #2) indicate a BC date. When a decade year is BC, a minus sign is placed in col. 5, overwriting the sixth character of the Site ID#. At present, all BC dates are actually off by one year, since there was actually no year 0.

APPENDIX B — TREES DATA FILE STRUCTURE AND FORMAT

Type	Variable Name or Comment	Data Type	Notes
Header	%trees_session*data~file:v0.0a1%		.tdf filemarker
File Comment	# DO NOT MODIFY THIS FILE!		
File Comment	# It was generated automatically by TREES v0.0a0		
File Comment	#		
File Comment	#Session info		
Array Variable	session		
Index	analyst_name	string	
Index	basename	string	
Index	datafile	string-filename	
Index	directory	string-full_directory_path	
Index	last_change_date	string-date-%m/%d/%y	
Index	last_save_date	string-date-{%l:%M %p %b %d, %Y}	
Index	needsave	boolean-yes/no	
Index	notes	list-string	
Index	sample_label	string	
Index	start_date	string-date-%m/%d/%y	
Index	status	string	
File Comment	#Capture info		
Array Variable	capture		
Index	frame0,estx	integer	
Index	frame0,esty	integer	
Index	frame0,file	string-filename	
Index	frame0,stagex	integer	
Index	frame0,stagey	integer	
	For N frames numbered as 1..N:		
Index	frame1,estx	integer	
Index	frame1,esty	integer	
Index	frame1,file	string-filename	
	...		
Index	frameN,estx	integer	
Index	frameN,esty	integer	
Index	frameN,file	string-filename	
Index	framecount	integer	(This is N, number of frames)
Index	frameheight	integer	
Index	framewidth	integer	
Index	illumination	string-filename	
Index	magnification	float	
Index	markcount	string	
Index	status	string	
Index	type	string	
File Comment	#Registration info		
Array Variable	registration		

Type	Variable Name or Comment	Data Type	Notes
Index	biasgain-adjust	boolean	
	For N frames numbered as 0..N-1:		
Index	frame0,bias	float	
Index	frame0,gain	float	
Index	frame0,gmax	float	
Index	frame0,gmin	float	
Index	frame0,x	integer	
Index	frame0,y	integer	
...	...		
Index	frameN,bias	float	
Index	frameN,gain	float	
Index	frameN,gmax	float	
Index	frameN,gmin	float	
Index	frameN,x	integer	
Index	frameN,y	integer	
Index	illum-adjust	boolean-0/1	
Index	status	string	
File Comment	#Mosaic info		
Array Variable	mosaic		
Index	factor	float	
Index	file,fullresol,index	string-filename	
Index	file,fullresol,mosaic	string-filename	
Index	file,fullresol,mosaic_columns	string-filename	
Index	file,index	string-filename	
Index	file,mosaic	string-filename	
Index	file,mosaic_columns	string-filename	
Index	file,smallmosaic	string-filename	
Index	frame0,glob_bias	float	
Index	frame0,glob_gain	float	
Index	frame0,glob_x	integer	
Index	frame0,glob_y	integer	
Index	frame1,glob_bias	float	
Index	frame1,glob_gain	float	
Index	frame1,glob_x	integer	
Index	frame1,glob_y	integer	
Index	frame2,glob_bias	float	
Index	frame2,glob_gain	float	
Index	frame2,glob_x	integer	
Index	frame2,glob_y	integer	
Index	gmin	float	
Index	indexfile	string-filename	
Index	mosaicfile	string-filename	
Index	mosaicheight	integer	
Index	mosaicwidth	integer	
Index	stage_offset,x	integer	
Index	stage_offset,y	integer	
Index	status	string	

Type	Variable Name or Comment	Data Type	Notes
File Comment	#Ring identification info		
Array Variable	findrings		
Index	file,chainmap	string-filename	
Index	file,linkmap	string-filename	
Index	file,ringmap	string-filename	
Index	file,smallchainmap	string-filename	
Index	file,smalllabeledrings	string-filename	
Index	file,smalllinkmap	string-filename	
Index	file,smallringmap	string-filename	
Index	loaded,smallchainmap	string-filename	
Index	ring,resolution	integer	
Index	status	string	
File Comment	#Measurement info		
Array Variable	measurements		
Index	file,ringwidths	string-full_path+filename	
Index	file,smallringwidths	string-full_path+filename	
Index	ringwidths	list-float	
Index	status	string	
File Comment	#Skeleton plot info		
Array Variable	skelplot		
Index	endyear	integer	
Index	notes	list-string	
Index	seriesname	string	
Index	startyear	integer	
Index	status	string	
File Comment	#Export Skeleton plot info		
Array Variable	plotinfo		
Index	abCutoff	integer	
Index	bone	list-float	
Index	diffCutoff	integer	
Index	indexval	list-float	
Index	LBoundary	list-string	
Index	RBoundary	list-string	
Index	size	integer	
Index	status	string	
Index	typelist	list-string	
Index	width	list-float	
Index	year	List-integer	

APPENDIX C — CROSSDATE SESSION FILE STRUCTURE AND FORMAT

Type	Variable Name or Comment	Data Type	Notes
Header	xdate_session*data~file:v0.0		.xds filemarker
File Comment	# DO NOT MODIFY THIS FILE!		
File Comment	# It was generated automatically by CROSSDATE v0.0a0		
File Comment	#		
File Comment	#Session info		
Array Variable	session		
Index	analyst	string	
Index	dir	string-full_pathname	
Index	file	string-filename	
Index	filename	string-full_pathname+filename	
Index	last_save	string-date-{%a, %d%M%y, %H:%M:%S}	
Index	notes	list-string	
Index	sample1,color	string	
Index	sample1,count,current	integer	
Index	sample1,display	boolean-0/1	
Index	sample1,filename	string-full_pathname+filename	
Index	sample1,invert	boolean-0/1	
Index	sample1,position	integer	
Index	sample1,status	string	
Index	sample2,color	string	
Index	sample2,count,current	integer	
Index	sample2,display	boolean-0/1	
Index	sample2,filename	string-full_pathname+filename	
Index	sample2,invert	boolean-0/1	
Index	sample2,position	integer	
Index	sample2,status	string	
Index	sample3,color	string	
Index	sample3,count,current	integer	
Index	sample3,display	boolean-0/1	
Index	sample3,filename	string-full_pathname+filename	
Index	sample3,invert	boolean-0/1	
Index	sample3,position	integer	
Index	sample3,status	string	
Index	start	string-date-{%a, %d%M%y, %H:%M:%S}	
Index	status	string	

APPENDIX D — CROSSDATE MASTER CHRONOLOGY FILE STRUCTURE AND FORMAT

Type	Variable Name or Comment	Data Type	Notes
Header	xdate_master*data~file:v0.0		.xmf filemarker
File Comment	# DO NOT MODIFY THIS FILE!		
File Comment	# It was generated automatically by CROSSDATE v0.0a0		
File Comment	#		
File Comment	#Session info		(Same as TREES Data File)
Array Variable	msession		
Index	analyst_name	String	
Index	basename	String	
Index	datafile	string-filename	
Index	directory	String-full_directory_path	
Index	last_change_date	String-date-%m/%d/%y	
Index	last_save_date	String-date-{%I:%M %p %b %d, %Y}	
Index	needsave	Boolean-yes/no	
Index	notes	List-string	
Index	sample_label	String	
Index	start_date	String-date-%m/%d/%y	
Index	status	String	
File Comment	#Skeleton Plot info		(Same as TREES Data File)
Array Variable	skelplot		
Index	endyear	Integer	
Index	notes	list-string	
Index	seriesname	string	
Index	startyear	integer	
Index	status	string	
File Comment	#Export Skeleton plot info		(Same as TREES Data File)
Array Variable	plotdata		
Index	abCutoff	integer	
Index	bone	list-float	
Index	diffCutoff	integer	
Index	indexval	list-float	
Index	LBoundary	list-string	NA for Master Chronology
Index	RBoundary	list-string	NA for Master Chronology
Index	size	integer	
Index	status	String	
Index	typelist	list-string	
Index	width	list-float	0 for Master Chronology
Index	year	List-integer	
File Comment	#Master Sample Component info		
Array Variable	master		
Index	elevation	integer	ITRDB Export Reqmt.
Index	indexval,num	list-integer	No. of samples
Index	indexval,sum	list-float	Sum of samples indexvals

Type	Variable Name or Comment	Data Type	Notes
Index	latLong	string	ITRDB Export Reqmt.
Index	sample1,filename	string-full_pathname+filename	
Index	sample1,type	integer	0=TREES Data File 1=XDATE Master Chronology 2=ITRDB Import Chronology last mod date/time for filename
Index	sample1,mtime	long integer	
Index	sample1,position	integer	
Index	sample1,size	integer	
Index	sample1,status	string-boolean-COMplete/NONE	
Index	sample2,filename	string-full_pathname+filename	
Index	sample2,type	integer	0=TREES Data File 1=XDATE Master Chronology 2=ITRDB Import Chronology last mod date/time for filename
Index	sample2,mtime	long integer	
Index	sample2,position	integer	
Index	sample2,size	integer	
Index	sample2,status	string-boolean-COMplete/NONE	
Index	sample3,filename	string-full_pathname+filename	
Index	sample3,type	integer	0=TREES Data File 1=XDATE Master Chronology 2=ITRDB Import Chronology last mod date/time for filename
Index	sample3,mtime	long integer	
Index	sample3,position	integer	
Index	sample3,size	integer	
Index	sample3,status	string-boolean-COMplete/NONE	
Index	siteID	string	ITRDB Export Reqmt.
Index	siteName	string	ITRDB Export Reqmt.
Index	species	string	ITRDB Export Reqmt.
Index	speciesCode	string	ITRDB Export Reqmt.
Index	state	string	ITRDB Export Reqmt.
Index	status	string	

```
[ ] Dendrochronology Program Library
[ ]
[ ] P R O G R A M      C O F E C H A
```

```
Run BLRW2 Program COF 11:35 Tue 02 May 2000 Page 1
```

```
Version 6.00P 23865
```

QUALITY CONTROL AND DATING CHECK OF TREE-RING MEASUREMENTS

Title of run: BAKER LAKE MONTANA LARCH RING WIDTHS

File of DATED series: BALTRW.RWL

CONTENTS:

- Part 1: Title page, options selected, summary, absent rings by series
- Part 2: Histogram of time spans
- Part 3: Master series with sample depth and absent rings by year
- Part 4: Bar plot of Master Dating Series
- Part 5: Correlation by segment of each series with Master
- Part 6: Potential problems: low correlation, divergent year-to-year changes, absent rings, outliers
- Part 7: Descriptive statistics

RUN CONTROL OPTIONS SELECTED

VALUE

- 1 Cubic smoothing spline 50% wavelength cutoff for filtering
32 years
- 2 Segments examined are 50 years lagged successively by 25 years
- 3 Autoregressive model applied A Residuals are used in master dating series and testing
- 4 Series transformed to logarithms Y Each series log-transformed for master dating series and testing
- 5 CORRELATION is Pearson (parametric, quantitative)
Critical correlation, 99% confidence level .3281
- 6 Master dating series saved N
- 7 Ring measurements listed N
- 8 Parts printed 1234567
- 9 Absent rings are omitted from master series and segment correlations (Y)

Time span of Master dating series is 987 to 1997 1011 years
Continuous time span is 987 to 1997 1011 years
Portion with two or more series is 1209 to 1997 789 years

>> BAL092 1775 absent in 1 of 28 series, but is not usually narrow: master index is -.049

```
*****
*C* Number of dated series 41 *C*
*O* Master series 987 1997 1011 yrs *O*
*F* Total rings in all series 14475 *F*
*E* Total dated rings checked 14253 *E*
*C* Series intercorrelation .679 *C*
*H* Average mean sensitivity .301 *H*
*A* Segments, possible problems 13 *A*
** Mean length of series 353.0 **
*****
```

1000	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000	Ident	Seq	Beg year	End year	Yrs
.	BAL021	1	1788	1997	210
.	BAL022	2	1783	1997	215
.	BAL031	3	1895	1997	103
.	BAL051	4	1848	1997	150
.	BAL052	5	1811	1997	187
.	BAL061	6	1814	1997	184
.	BAL081	7	1846	1997	152
.	BAL091	8	987	1997	1011
.	BAL092	9	1295	1997	703
.	BAL311	10	1815	1997	183
.	BAL313	11	1743	1997	255
.	BAL332	12	1500	1997	498
.	BAL35A	13	1399	1997	599
.	BAL35B	14	1627	1997	371
.	BAL36A	15	1751	1997	247
.	BAL37A	16	1445	1997	553
.	BAL38A	17	1209	1797	589
.	BAL391	18	1665	1997	333
.	BAL39A	19	1390	1997	608
.	BAL39B	20	1469	1997	529
.	BAL406	21	1449	1997	549
.	BAL40A	22	1750	1997	248
.	BAL41A	23	1663	1997	335
.	BAL41B	24	1743	1997	255
.	BAL426	25	1435	1970	536
.	BAL42A	26	1511	1997	487
.	BAL43A	27	1885	1997	113
.	BAL43B	28	1450	1721	272
.	BAL501	29	1773	1997	225
.	BAL502	30	1662	1997	336
.	BAL511	31	1537	1997	461
.	BAL531	32	1637	1997	361
.	BAL532	33	1654	1997	344
.	BAL672	34	1632	1997	366
.	BAL862	35	1885	1997	113
.	BL301A	36	1853	1997	145
.	BL301B	37	1443	1795	353
.	BL512A	38	1527	1997	471
.	BL512B	39	1527	1997	471
.	BL671A	40	1861	1997	137
.	BL671B	41	1572	1788	217

Part 2 — Histogram of Time Spans

Year	Value	No Ab	Year	Value	No Ab	Year	Value	No Ab	Year	Value	No Ab	Year	Value	No Ab	Year	Value	No Ab
1100	-1.987	1	1150	.469	1	1200	-3.560	1	1250	.983	2	1300	-.437	3	1350	-1.177	3
1101	.155	1	1151	.098	1	1201	.988	1	1251	.386	2	1301	.397	3	1351	-.534	3
1102	-.106	1	1152	.327	1	1202	1.352	1	1252	-1.133	2	1302	.380	3	1352	1.395	3
1103	-1.142	1	1153	.553	1	1203	-.142	1	1253	.606	2	1303	.683	3	1353	.038	3
1104	.654	1	1154	-.136	1	1204	.075	1	1254	.514	2	1304	-.547	3	1354	.012	3
1105	-.461	1	1155	1.947	1	1205	1.586	1	1255	1.396	2	1305	-.641	3	1355	1.073	3
1106	-4.168	1	1156	.069	1	1206	.251	1	1256	-5.01	2	1306	.874	3	1356	1.541	3
1107	-.698	1	1157	.625	1	1207	-.586	1	1257	1.032	2	1307	.068	3	1357	-.992	3
1108	-.179	1	1158	-5.605	1	1208	1.498	1	1258	-.263	2	1308	-.529	3	1358	1.257	3
1109	.133	1	1159	1.094	1	1209	-1.463	2	1259	-.649	2	1309	.232	3	1359	-.218	3
1110	1.911	1	1160	1.321	1	1210	-.086	2	1260	1.075	2	1310	-.184	3	1360	-.442	3
1111	.676	1	1161	.471	1	1211	1.027	2	1261	.698	2	1311	.991	3	1361	-1.705	3
1112	.766	1	1162	-.219	1	1212	-1.198	2	1262	-1.664	2	1312	-.601	3	1362	.417	3
1113	1.035	1	1163	-.264	1	1213	.351	2	1263	-.365	2	1313	1.200	3	1363	.415	3
1114	1.452	1	1164	.364	1	1214	.754	2	1264	-.605	2	1314	1.230	3	1364	.566	3
1115	1.005	1	1165	1.189	1	1215	.045	2	1265	-.424	2	1315	.081	3	1365	1.434	3
1116	.172	1	1166	-.790	1	1216	-.172	2	1266	-1.109	2	1316	1.311	3	1366	-.085	3
1117	1.376	1	1167	-2.823	1	1217	1.316	2	1267	-.145	2	1317	.264	3	1367	-.224	3
1118	-2.545	1	1168	.174	1	1218	.808	2	1268	.308	2	1318	-1.684	3	1368	-.067	3
1119	-.550	1	1169	-2.425	1	1219	1.276	2	1269	-.672	2	1319	-.571	3	1369	-2.971	3
1120	-1.910	1	1170	-.663	1	1220	-.184	2	1270	.430	2	1320	-.042	3	1370	.874	3
1121	-.482	1	1171	.897	1	1221	-2.060	2	1271	1.768	2	1321	-1.133	3	1371	1.026	3
1122	-.715	1	1172	.563	1	1222	.336	2	1272	.198	2	1322	.728	3	1372	1.029	3
1123	-.420	1	1173	.524	1	1223	.646	2	1273	1.058	2	1323	.672	3	1373	-.671	3
1124	-.125	1	1174	1.308	1	1224	-.140	2	1274	1.948	2	1324	-.007	3	1374	-.575	3
1125	.635	1	1175	1.045	1	1225	.789	2	1275	.717	2	1325	-2.565	3	1375	-1.114	3
1126	-.597	1	1176	.788	1	1226	.104	2	1276	.854	2	1326	1.342	3	1376	-.870	3
1127	.501	1	1177	.541	1	1227	-.356	2	1277	-.901	2	1327	.715	3	1377	-.547	3
1128	2.133	1	1178	-.333	1	1228	-.103	2	1278	-1.609	2	1328	.381	3	1378	-.421	3
1129	-.145	1	1179	1.866	1	1229	-.054	2	1279	.625	2	1329	.693	3	1379	-.454	3
1130	.435	1	1180	1.004	1	1230	-2.010	2	1280	.439	2	1330	-.658	3	1380	-.132	3
1131	.259	1	1181	-.028	1	1231	2.169	2	1281	-2.050	2	1331	.713	3	1381	.044	3
1132	.797	1	1182	-.567	1	1232	.091	2	1282	-1.170	2	1332	-.394	3	1382	-.025	3
1133	1.073	1	1183	-.784	1	1233	-.462	2	1283	-.485	2	1333	1.378	3	1383	-.201	3
1134	1.330	1	1184	.887	1	1234	-2.832	2	1284	-.375	2	1334	.307	3	1384	.545	3
1135	-.191	1	1185	-1.260	1	1235	.614	2	1285	.015	2	1335	-1.659	3	1385	-.311	3
1136	-.674	1	1186	1.797	1	1236	-1.121	2	1286	.311	2	1336	-.415	3	1386	.458	3
1137	-1.944	1	1187	-1.399	1	1237	.144	2	1287	-.264	2	1337	.075	3	1387	1.403	3
1138	-3.737	1	1188	.168	1	1238	.272	2	1288	.378	2	1338	.559	3	1388	1.179	3
1139	-.281	1	1189	-1.623	1	1239	-1.117	2	1289	-.494	2	1339	.295	3	1389	1.004	3

Part 3 — Master series with sample depth and absent rings by year

Year Rel value	Year Rel value	Year Rel value	Year Rel value	Year Rel value	Year Rel value	Year Rel value	Year Rel value
1200n	1250-----D	1300--b	1350-e	1400---a	1450-----@	1500-----C	1550-d
1201-----D	1251-----B	1301-----B	1351--b	1401-----F	1451-d	1501-----A	1551-----A
1202-----E	1252-e	1302-----B	1352-----F	1402-----C	1452--b	1502-c	1552-----E
1203----a	1253-----B	1303-----C	1353-----@	1403--b	1453-----A	1503f	1553-----@
1204----@	1254-----B	1304--b	1354-----@	1404----A	1454----A	1504--b	1554-----B
1205-----F	1255-----F	1305--c	1355-----D	1405f	1455-----B	1505g	1555--b
1206-----A	1256--b	1306-----C	1356-----F	1406-----C	1456-----@	1506--a	1556-----B
1207--b	1257-----D	1307-----@	1357-d	1407-----B	1457-----A	1507-----B	1557--c
1208-----F	1258--a	1308--b	1358-----E	1408--a	1458v	1508-----B	1558-----B
1209f	1259--c	1309-----A	1359--a	1409-d	1459--b	1509-----B	1559-----B
-----	-----	-----	-----	-----	-----	-----	-----
1210----@	1260-----D	1310---a	1360--b	1410---a	1460-----D	1510-----E	1560-----A
1211-----D	1261-----C	1311-----D	1361g	1411-----B	1461----@	1511---a	1561-d
1212-e	1262g	1312--b	1362-----B	1412-d	1462-----C	1512-----E	1562--b
1213-----A	1263--a	1313-----E	1363-----B	1413-----D	1463-----B	1513-----E	1563----A
1214-----C	1264--b	1314-----E	1364-----B	1414-----A	1464-----C	1514--b	1564-----B
1215----@	1265--b	1315-----@	1365-----F	1415-----A	1465-----B	1515-----A	1565h
1216----a	1266-d	1316-----E	1366-----@	1416-----@	1466-----E	1516-----@	1566--a
1217-----E	1267--a	1317-----A	1367--a	1417s	1467-----B	1517-----F	1567-----B
1218-----C	1268----A	1318g	1368----@	1418-----E	1468--a	1518f	1568-----B
1219-----E	1269--c	1319--b	1369l	1419-----B	1469-----C	1519----A	1569--b
-----	-----	-----	-----	-----	-----	-----	-----
1220----a	1270-----B	1320----@	1370-----C	1420-----B	1470-----D	1520p	1570-----A
1221h	1271-----G	1321-e	1371-----D	1421-----C	1471-----C	1521----@	1571----A
1222-----A	1272-----A	1322-----C	1372-----D	1422-----A	1472n	1522----@	1572-----B
1223-----C	1273-----D	1323-----C	1373--c	1423-----@	1473-----A	1523-----A	1573--a
1224----a	1274-----H	1324----@	1374--b	1424--a	1474-----A	1524--b	1574-----B
1225-----C	1275-----C	1325j	1375-d	1425--c	1475--a	1525--b	1575-----C
1226-----@	1276-----C	1326-----E	1376-c	1426-----E	1476-d	1526--a	1576-----G
1227--a	1277-d	1327-----C	1377--b	1427--a	1477-d	1527-----D	1577--a
1228----@	1278f	1328-----B	1378--b	1428-----A	1478-----B	1528--a	1578----@
1229----@	1279-----B	1329-----C	1379--b	1429--a	1479-----D	1529h	1579----A
-----	-----	-----	-----	-----	-----	-----	-----
1230h	1280-----B	1330--c	1380--a	1430----@	1480----@	1530-----C	1580-----A
1231-----I	1281h	1331-----C	1381-----@	1431--b	1481-----C	1531-----F	1581----@
1232-----@	1282-e	1332--b	1382----@	1432----@	1482-----D	1532-----E	1582-d
1233--b	1283--b	1333-----F	1383--a	1433-----F	1483--c	1533-----C	1583--a
1234k	1284--b	1334-----A	1384-----B	1434--a	1484f	1534-----A	1584-----@
1235-----B	1285-----@	1335g	1385--a	1435--b	1485--b	1535-----C	1585--c
1236-d	1286-----A	1336--b	1386-----B	1436-----C	1486g	1536--b	1586-e
1237-----A	1287--a	1337-----@	1387-----F	1437-----D	1487-----D	1537----A	1587-----@
1238-----A	1288-----B	1338-----B	1388-----E	1438----@	1488-----A	1538--b	1588-----C
1239-d	1289--b	1339-----A	1389-----D	1439-----D	1489-----C	1539--a	1589-----C

Part 4 — Bar plot of master dating series

Correlations of 50-year dated segments, lagged 25 years

Flags: A = correlation under .3281 but highest as dated; B = correlation higher at other than dated position

Seq	Series	Time_span	1200	1225	1250	1275	1300	1325	1350	1375	1400	1425	1450	1475	1500	1525	1550	1575	1600	1625	1650	1675
			1249	1274	1299	1324	1349	1374	1399	1424	1449	1474	1499	1524	1549	1574	1599	1624	1649	1674	1699	1724
8	BAL091	987 1997	.45	.48	.41	.31A	.49	.73	.65	.61	.71	.68	.65	.81	.87	.59	.50	.78	.78	.70	.64	.74
9	BAL092	1295 1997				.56	.55	.66	.65	.72	.77	.69	.71	.70	.71	.61	.67	.75	.81	.83	.72	.68
12	BAL332	1500 1997													.66	.66	.59	.54	.69	.74	.61	.57
13	BAL35A	1399 1997								.80	.84	.85	.74	.74	.85	.80	.66	.75	.84	.83	.72	.79
14	BAL35B	1627 1997																		.83	.69	.70
16	BAL37A	1445 1997									.81	.81	.85	.78	.57	.52	.73	.84	.82	.77	.85	
17	BAL38A	1209 1797	.45	.48	.39	.37	.60	.63	.46	.63	.74	.82	.87	.78	.78	.57	.57	.61	.59	.71	.62	.64
18	BAL391	1665 1997																			.72	.78
19	BAL39A	1390 1997								.67	.66	.55	.70	.83	.86	.77	.68	.71	.74	.71	.73	.74
20	BAL39B	1469 1997											.77	.78	.86	.55	.54	.73	.73	.67	.74	.69
21	BAL406	1449 1997									.51	.48	.62	.75	.64	.50	.68	.69	.58	.41	.27A	
23	BAL41A	1663 1997																			.47	.44
25	BAL426	1435 1970									.80	.66	.82	.78	.61	.58	.76	.76	.67	.58	.54	
26	BAL42A	1511 1997													.73	.68	.60	.69	.85	.72	.59	.62
28	BAL43B	1450 1721										.57	.68	.73	.68	.54	.62	.76	.72	.63	.67	
30	BAL502	1662 1997																			.70	.83
31	BAL511	1537 1997														.57	.67	.81	.82	.77	.67	.63
32	BAL531	1637 1997																		.77	.72	.74
33	BAL532	1654 1997																			.81	.76
34	BAL672	1632 1997																			.45	.50
37	BL301B	1443 1795									.63	.60	.76	.81	.58	.59	.76	.74	.73	.65	.60	
38	BL512A	1527 1997													.61	.61	.83	.78	.74	.74	.70	
39	BL512B	1527 1997													.70	.64	.77	.83	.73	.71	.67	
41	BL671B	1572 1788														.56	.57	.55	.58	.66	.71	
Av	segment	correlation	.45	.48	.40	.41	.55	.67	.59	.69	.74	.70	.69	.76	.78	.64	.59	.71	.75	.72	.66	.67

Part 5 — Correlation by segment of each series with master

For each series with potential problems the following diagnostics may appear:

[A] Correlations with master dating series of flagged 50-year segments of series filtered with 32-year spline, at every point from ten years earlier (-10) to ten years later (+10) than dated

[B] Effect of those data values which most lower or raise correlation with master series

[C] Year-to-year changes very different from the mean change in other series

[D] Absent rings (zero values)

[E] Values which are statistical outliers from mean for the year

=====

BAL021 1788 to 1997 210 years Series 1

[B] Entire series, effect on correlation (.656) is:

Lower 1809 -.026 1993 -.010 1840 -.009 1792 -.007 Higher 1832 .040 1915 .024 1877 .019 1838 .011

[E] Outliers 7 3.0 SD above or -4.5 SD below mean for year

1809 +6.0 SD; 1818 +3.2 SD; 1903 +3.3 SD; 1907 +3.1 SD; 1925 -5.0 SD; 1956 +3.6 SD; 1993 +3.6 SD

=====

BAL022 1783 to 1997 215 years Series 2

[A] Segment High -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +10

1925 1974 -10 .44*-.07 -.07 -.31 -.04 -.02 .13 -.19 .10 .03 .41|.00 -.01 .20 -.06 .18 -.15 -.10 -.16 .21 -.05

[B] Entire series, effect on correlation (.502) is:

Lower 1809 -.017 1993 -.016 1840 -.015 1893 -.014 Higher 1803 .025 1877 .023 1832 .022 1915 .018

1925 to 1974 segment:

Lower 1949 -.026 1928 -.019 1925 -.019 1944 -.018 Higher 1933 .029 1948 .023 1927 .020 1968 .019

[E] Outliers 11 3.0 SD above or -4.5 SD below mean for year

1789 -5.1 SD; 1809 +5.0 SD; 1810 +3.5 SD; 1840 -5.1 SD; 1843 +4.1 SD; 1864 +3.1 SD; 1893 -6.6 SD;

1903 +3.3 SD; 1925 -7.1 SD; 1956 +3.0 SD; 1993 +5.0 SD

=====

Part 6 — Potential problems: low correlation, divergent year-to-year changes, outliers

Seq	Series	Interval	No. Years	No. Segmt	No. Flags	Corr with Master	//----- Mean msmt	----- Max msmt	----- Std dev	----- Auto corr	----- Mean sens	----- Max value	----- Std dev	----- Auto corr	AR ()
1	BAL021	1788 1997	210	8	0	.656	.69	1.88	.402	.877	.233	2.56	.367	-.012	1
2	BAL022	1783 1997	215	8	1	.502	.84	2.62	.451	.836	.251	2.47	.377	-.018	1
3	BAL031	1895 1997	103	4	1	.369	.74	2.10	.350	.453	.356	2.63	.566	-.030	1
4	BAL051	1848 1997	150	6	0	.551	.26	.45	.072	.494	.231	2.48	.442	.001	1
5	BAL052	1811 1997	187	7	0	.569	.54	1.25	.161	.545	.219	2.64	.350	-.034	4
6	BAL061	1814 1997	184	7	0	.562	.30	.58	.082	.366	.243	2.52	.392	-.012	1
7	BAL081	1846 1997	152	6	5	.356	.41	.79	.103	.554	.194	2.67	.383	-.038	3
8	BAL091	987 1997	1011	31	1	.670	.21	.55	.088	.766	.239	2.50	.261	-.018	2
9	BAL092	1295 1997	703	28	0	.677	.22	.73	.094	.704	.262	2.50	.232	-.024	1
10	BAL311	1815 1997	183	7	0	.622	.26	.63	.114	.388	.421	2.60	.475	-.022	1
11	BAL313	1743 1997	255	10	0	.681	.19	.45	.087	.558	.396	2.60	.322	.011	1
12	BAL332	1500 1997	498	19	0	.661	.19	.54	.077	.730	.228	2.45	.269	-.022	2
13	BAL35A	1399 1997	599	24	0	.802	.29	.81	.134	.739	.270	2.50	.308	-.018	2
14	BAL35B	1627 1997	371	14	0	.807	.23	.54	.075	.526	.257	2.54	.345	.004	2
15	BAL36A	1751 1997	247	9	0	.738	.90	1.74	.271	.520	.255	2.43	.323	-.020	1
16	BAL37A	1445 1997	553	22	0	.776	.24	1.05	.140	.704	.352	2.55	.307	.004	3
17	BAL38A	1209 1797	589	23	0	.626	.33	1.33	.213	.842	.311	2.62	.348	-.018	1
18	BAL391	1665 1997	333	13	0	.738	.19	.55	.085	.543	.377	2.54	.338	-.004	1
19	BAL39A	1390 1997	608	24	0	.784	.18	.66	.094	.690	.376	2.47	.308	-.015	1
20	BAL39B	1469 1997	529	21	0	.692	.14	.37	.063	.524	.375	2.62	.377	-.024	1
21	BAL406	1449 1997	549	22	1	.641	.21	.48	.076	.579	.271	2.54	.284	-.018	1
22	BAL40A	1750 1997	248	9	0	.753	.19	.44	.064	.480	.265	2.49	.275	-.012	1
23	BAL41A	1663 1997	335	13	0	.627	.15	.39	.066	.507	.349	2.52	.332	-.016	1
24	BAL41B	1743 1997	255	10	0	.650	.18	.49	.071	.481	.310	2.55	.366	-.008	1
25	BAL426	1435 1970	536	21	0	.718	.23	.64	.095	.448	.368	2.72	.373	-.011	1
26	BAL42A	1511 1997	487	19	0	.691	.23	.57	.086	.399	.356	2.64	.386	-.011	1
27	BAL43A	1885 1997	113	4	0	.537	.45	1.23	.231	.702	.318	2.70	.554	.029	1
28	BAL43B	1450 1721	272	10	0	.727	.25	.58	.105	.330	.402	2.53	.357	-.028	1
29	BAL501	1773 1997	225	9	0	.784	.19	.58	.109	.750	.336	2.52	.351	-.016	1
30	BAL502	1662 1997	336	13	0	.746	.15	.40	.064	.538	.358	2.50	.312	-.014	1
31	BAL511	1537 1997	461	18	0	.667	.29	.68	.117	.707	.246	2.58	.379	-.029	1
32	BAL531	1637 1997	361	14	0	.715	.37	.94	.165	.706	.284	2.42	.285	-.021	1
33	BAL532	1654 1997	344	13	0	.738	.42	1.03	.177	.690	.265	2.41	.287	-.036	1
34	BAL672	1632 1997	366	14	2	.484	.19	.37	.065	.296	.335	2.49	.310	-.010	2
35	BAL862	1885 1997	113	4	2	.406	.38	.95	.198	.830	.257	2.41	.380	.050	1
36	BL301A	1853 1997	145	5	0	.666	.31	.80	.120	.686	.256	2.41	.291	-.026	1
37	BL301B	1443 1795	353	14	0	.715	.18	.43	.072	.576	.317	2.66	.328	.010	1
38	BL512A	1527 1997	471	18	0	.681	.24	.84	.129	.835	.257	2.71	.362	-.021	1
39	BL512B	1527 1997	471	18	0	.730	.25	.74	.118	.743	.292	2.55	.327	-.024	2
40	BL671A	1861 1997	137	5	0	.532	.21	.45	.056	.323	.252	2.77	.464	.020	1
41	BL671B	1572 1788	217	9	0	.605	.19	.51	.079	.500	.317	2.70	.408	-.024	1
Total or mean:			14475	553	13	.679	.27	2.62	.119	.623	.301	2.77	.332	-.015	--

-- = [COFECHA BLRW2COF] = --

APPENDIX F — MATHEMATICAL BASIS FOR SMOOTHING

SPLINE ALGORITHM

Cubic splines are typically used to interpolate new values between adjacent (discrete) measurements. The assumption is that the measured points are known without error. Reinsch [39] takes the approach that a cubic smoothing spline is a least squares fit to experimental data for extracting the underlying function in a piecewise fashion. Like the cubic spline, the cubic smoothing spline has second derivative continuity at segment boundaries.

The smoothing spline algorithm for the TREES program is based on a special case of Reinsch's model for equally spaced data and minimizes the total squared curvature of the spline,

$$\int_{x_0}^x [g''(x)]^2 dx \quad (\text{F-1})$$

under the constraint

$$\sum_{i=0}^n \left[\frac{g(x_i) - y_i}{\delta y_i} \right]^2 \leq S \quad (\text{F-2})$$

where:

y_i is the ringwidth series measurements.

δy_i is a series of weights for individual data points.

S is a scaling parameter.

$g(x_i)$ is the spline function.

This is solved by introducing a Lagrangian multiplier, p , and an auxiliary variable, z , and minimizing the functional:

$$\int_{x_0}^x \left[\frac{d^2 g(x)}{dx^2} \right]^2 dx + p \left\{ \sum_{i=0}^n \left[\frac{g(x_i) - y_i}{\delta y_i} \right]^2 + z^2 - S \right\} \quad (\text{F-3})$$

The solution is a cubic spline (with $\delta y_i = 1$ and $S = 0$ for convenience):

$$g_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3, \quad x_i \leq x < x_{i+1} \quad (\text{F-4})$$

such that $g(x)$, $dg(x)/dx$, and $d^2g(x)/dx^2$ are continuous at x_i for $i = 1$ to $n-1$.

The Lagrangian multiplier is found through Fourier transformation of the matrix operations in the computation of the spline for a specified frequency, f :

$$p = \frac{6(\cos 2\pi f - 1)^2}{(\cos 2\pi f - 2)} \quad (\text{F-5})$$

The cubic spline values are found by solving simultaneous equations for the coefficient c_i , then using c_i to solve for a_i , b_i , and d_i as follows:

$$c_i: \quad (\mathbf{Q}^T \mathbf{Q} + p \mathbf{T}) \mathbf{c} = p \mathbf{Q}^T \mathbf{y} \quad i = 1, \dots, n-1 \quad c_0 = c_n = 0 \quad (\text{F-6})$$

$$a_i: \quad \mathbf{y} - p^{-1} \mathbf{Q} \mathbf{c} \quad i = 0, \dots, n \quad (\text{F-7})$$

$$d_i: \quad (c_{i+1} - c_i)/3 \quad i = 0, \dots, n-1 \quad (\text{F-8})$$

$$b_i: \quad (a_{i+1} - a_i) - c_i - d_i \quad i = 0, \dots, n-1 \quad (\text{F-9})$$

$$\mathbf{y}: \quad \text{the ring width series data vector } \{y_0, y_1, \dots, y_n\}^T$$

$$\mathbf{T}: \quad \text{a positive-definite tridiagonal matrix } (n-1) \times (n-1): \quad (\text{F-10})$$

$$\begin{bmatrix} 4/3 & 1/3 & & & & \\ & 1/3 & 4/3 & & & \\ & & 1/3 & 4/3 & & \\ & & & 1/3 & 4/3 & \\ & & & & 0 & \\ & & & & & 0 \\ & & & & & & 1/3 & 4/3 \end{bmatrix}$$

REFERENCES

- [1] M.G.L. Baillie and J.R. Pilcher, "*A Simple Crossdating Program for Tree-Ring Research*". *Tree-Ring Bulletin*, Vol. 33, 1973, pp. 7-14.
- [2] Julius S. Bendat, Allen G. Piersol, *Engineering Applications of Correlation and Spectral Analysis, Second Edition*. John Wiley, New York, 1993.
- [3] Petr Beckmann, *Probability in Communication Engineering*. Harcourt, Brace, and World, Inc., New York, 1967.
- [4] George E. P. Box and Gwilyn M. Jenkins, *Time Series Analysis, Forecasting and Control*. Holden-Day, San Francisco, 1976.
- [5] C.H. Chen, *Digital Waveform Processing and Recognition*. CRC Press, Boca Raton, Florida, 1982.
- [6] W. Steven Conner, *A Computer Vision Based Tree Ring Analysis and Dating System*. Master s Thesis, The University of Arizona, Department of Electrical and Computer Engineering, 1999.
- [7] W. Steven Conner, Robert A. Schowengerdt, Martin Munro, Malcolm K. Hughes, *Engineering Design of an Image Acquisition and Analysis System for Dendrochronology* . *Optical Engineering*, Vol. 39, No. 2, February 2000, pp. 453-463.
- [8] Edward R. Cook, *A Time Series Approach to Tree-Ring Standardization*. Ph.D. Dissertation, The University of Arizona, School of Renewable Natural Resources, 1985.
- [9] Edward R. Cook and Kenneth Peters, *The Cubic Smoothing Spline as a Digital Filter* . Technical Report No. CU-1-81/TR1, Lamont-Doherty Geological Observatory of Columbia University, Tree-Ring Laboratory, Palisades, New York, October 1981.
- [10] Edward R. Cook and Kenneth Peters, *The Smoothing Spline: A New Approach to Standardizing Forest Interior Tree-Ring Width Series for Dendroclimatic Studies* . *Tree-Ring Bulletin*, The University of Arizona, Vol. 41, 1981, pp. 45-53.
- [11] E. R. Cook and L.A. Kairiukstis, *Methods of Dendrochronology*. Kluwer Academic Publishers, Dordrecht, Netherlands, 1990.
- [12] John Philip Cropper, *Tree-Ring Skeleton Plotting By Computer* . *Tree-Ring Bulletin*, Vol. 39, 1979, pp. 47-54.

REFERENCES - CONTINUED

- [13] A. E. Douglass, *Precision of Ring Dating in Tree-Ring Chronologies* . University of Arizona Bulletin, Laboratory for Tree-Ring Research Bulletin No. 3, Vol. 17, No. 3, July 1, 1946.
- [14] A. E. Douglass, *Researches in Dendrochronology* . Bulletin of the University of Utah, Vol. 37, No. 2, July 12, 1946.
- [15] A. E. Douglass, *Some Aspects of the Use of the Annual Rings of Trees in Climatic Study* , The Scientific Monthly, Vol. 15, No. 1, July 1922, pp. 1-22.
- [16] A. E. Douglass, *The Annual Rings of Trees* , The Scientific Monthly, Vol. 37, December 1933, pp. 481-495.
- [17] A. E. Douglass, *Tree Rings and Chronology* . University of Arizona Bulletin, Physical Science Bulletin No. 1, Vol. 8, No. 4, October 1, 1937.
- [18] A. E. Douglass, *Climatic Cycles and Tree-Growth: A Study of the Annual Rings of Trees in Relation to Climate and Solar Activity*. Wheldon and Wesley, Ltd., New York, 1971. (Originally published in 1919 by The Carnegie Institution of Washington).
- [19] Harold C. Fritts, *Computer Programs for Tree-Ring Research* . Tree-Ring Bulletin, Vol. 25, No. 3-4, 1963, pp. 2-7.
- [20] Harold C. Fritts, *Tree Rings and Climate*. Academic Press, London, 1987, pp. 567.
- [21] Judith L. Gersting, *Mathematical Structures for Computer Science, Third Edition*, Computer Science Press, an imprint of W. H. Freeman and Company, New York, 1993.
- [22] Regent Guay, Rejean Gagnon, and Hubert Morin, *MacDENDRO™, A New Automatic and Interactive Tree Ring Measurement System Based on Image Processing* . Tree Rings and Environment, Proceedings of the International Dendrochronological Symposium, Ystad, South Sweden, 3-9 September 1990.
- [23] Mark Harrison, Michael McLennan, *Effective Tcl/Tk Programming*. Addison Wesley, Reading, Mass., 1998.
- [24] Mark R. Headington, David D. Riley, *Data Abstraction and Structures Using C++*, D.C. Heath and Company, Lexington, Massachusetts, 1994.
- [25] Herbert L. Hirsch, *Statistical Signal Characterization*. Artech House, Boston, 1992.

REFERENCES - CONTINUED

- [26] Richard L. Holmes, *Computer-Assisted Quality Control in Tree-Ring Dating and Measurement* . Tree-Ring Bulletin, The University of Arizona, Vol. 43, 1983, pp. 69-78.
- [27] Richard L. Holmes and Harold C. Fritts, *Procedures used in the chronology development work including users manuals for computer programs COFECHA and ARSTAN* . Laboratory for Tree-Ring Research, University of Arizona, 1986.
- [28] Richard L. Holmes, *Users Manual for Program COFECHA* . Laboratory for Tree-Ring Research, University of Arizona, February 1999.
- [29] Gwilyn M. Jenkins, Donald G. Watts, *Spectral Analysis and Its Applications*. Holden-Day, San Francisco, 1968.
- [30] Steven M. Kay, *Fundamentals of Statistical Signal Processing. Detection Theory*, Prentice-Hall, Upper Saddle River, New Jersey, 1998.
- [31] Ronald N. Kickert, Barbara Herren-Gemmill, Rodney Arkley, and Robert E. Thomson, *CURVES and ALIGN: Some New Computerized Methods for the Analysis of Tree-Ring Data* . Tree-Ring Bulletin, The University of Arizona, Vol. 43, 1983, pp. 79-88.
- [32] Glenford J. Myers, *The Art of Software Testing*, John Wiley and Sons, New York, 1979.
- [33] Martin A. R. Munro, "*An Improved Algorithm for Crossdating Tree-Ring Series*". Tree-Ring Bulletin, Vol. 44, 1984, pp. 17-27.
- [34] M.A.R. Munro, M.K. Hughes, R. Schowengerdt, W.S. Conner, J.B. Engle, G. Gopalan, P.R. Sheppard, J.M. Burns, *TREES: A System for Computer-assisted Dendrochronology* , presented at the International Conference on Dendrochronology in the Third Millennium, 1-7 April, 2000, Mendoza, Argentina.
- [35] Henri J. Nussbaumer, *Fast Fourier Transform and Convolution Algorithms*. Springer-Verlag, Berlin, 1981.
- [36] Robert K. Otnes, Loren Enochson, *Digital Time Series Analysis*. John Wiley, New York, 1972.
- [37] Martyn A. Ould and Charles Unwin, *Testing in Software Development*. Cambridge University Press for the British Computer Society, Cambridge, England, 1986.

REFERENCES - CONTINUED

- [38] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, *Numerical Recipes in C, The Art of Scientific Computing, Second Edition*. Cambridge University Press, Cambridge, UK, 1992.
- [39] Christian H. Reinsch, *Smoothing by Spline Functions*. Numerische Mathematik, Vol. 10, 1967, pp. 177-183.
- [40] Marc Roper, *Software Testing*. McGraw-Hill Book Company Europe, London, England, 1994.
- [41] Richard Schiava, *Introduction to Applied Statistical Signal Analysis*. Aksen Associates, Inc. Publishers, Homewood, Illinois, 1991.
- [42] Jesper Schou and Erik Rytter, *Dendrochronological Dating Using Scanning and Imageprocessing*. Tree Rings and Environment, Proceedings of the International Dendrochronological Symposium, Ystad, South Sweden, 3-9 September 1990.
- [43] Paul Sheppard, *Crossdating Tree Rings Using Skeleton Plotting*. An interactive Java-based website presentation, The Laboratory for Tree-Ring Research, The University of Arizona, 1999.
URL: <http://tree.ltrr.arizona.edu/skeletonplot/introcrossdate.htm>
- [44] M.D. Srinath, P.K. Rajasekaran, R. Viswanathan, *Introduction to Statistical Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1996.
- [45] Marvin A. Stokes, Terah L. Smiley, *An Introduction to Tree-Ring Dating*. University of Arizona Press, Tucson, Arizona, 1996.
- [46] Ramzi Touchan, Peter Ffolliott, and Malcolm Hughes, *Mitigating Risks to Conservation and Sustainable Use of Water and Other Natural Resources*, Technical Training Course, Laboratory of Tree Ring Research, University of Arizona, 1999.
- [47] Ronald E. Walpole, Raymond H. Myers, *Probability and Statistics for Engineers and Scientists*, Second Edition. Macmillan Publishing, New York, 1978.
- [48] Brent B. Welch, *Practical Programming in Tcl and Tk, Second Edition*. Prentice-Hall, Upper Saddle River, NJ, 1997.
- [49] Wayne M. Wendland, *An Objective Method to Identify Missing or False Rings*. Tree-Ring Bulletin, Vol. 35, 1975, pp. 41-47.
- [50] T. M. L. Wigley, P. D. Jones, K. R. Briffa, *Cross-dating Methods in Dendrochronology*. Journal of Archaeological Science, Vol. 14, 1987, pp. 51-64.