

# Analysis, Design, Rating, and Drafting of Wood Bridge Superstructures

**W. Randall Thomas and J. A. Puckett, P.E., Ph.D.,** Department of Civil and Architectural Engineering, University of Wyoming

## Abstract

A program for engineering wood bridges is described. The program performs the analysis, design, and/or rating of wood bridge superstructures in accordance with the 15th Edition of the AASHTO Specification or the AASHTO-LRFD Specification. Bridges may be analyzed for design review and/or rating. In review/rating mode, the superstructure is analyzed and checked against all applicable specifications. In the design mode, minimal user input, such as span length, roadway width, bridge type, and materials are entered and the program reviews designs and issues details associated with standard plan sets. The appropriate design-specific information is included in a DXF format for inclusion into the final drawings. A Microsoft Windows-based user interface is used for the data definition. The data is passed to another module for engineering computations. The associated documentation is contained in both on-line and hard copy formats. The most commonly used bridge and deck systems are accommodated in either U.S. customary or SI units.

Keywords: bridge design, wood bridge, bridge design software

## Introduction

Computer programs have been used as an aid for bridge design for many years. Since the introduction of inexpensive personal computers, access to design programs has expanded greatly within all levels of the design community. In addition to general structural analysis programs, many design programs have been developed to assist with specific design applications involving both bridge superstructures and substructures. With few exceptions, these programs have been targeted at bridges constructed of steel and concrete. Very few computer design programs have included wood as a design material option.

In order to facilitate the design of wood bridges, computerized design programs for wood bridge superstructures must be developed that are comparable to those currently available for steel and concrete bridges. Otherwise, designers may unjustifiably select steel or concrete on the basis of the design tools available. These programs are needed not only in the traditional allowable stress design format, but also in the newly-introduced load and resistance factor design format. This paper presents the development of Madero (Spanish for timber beam), a complete computer package for timber bridge superstructures. Program functionality and design are discussed, along

with graphical user interface features, analysis methods, and implementation examples.

## Project Objectives

The main objective of the project is to develop a comprehensive computer program for the analysis, design, drafting, and rating of wood bridge superstructures. In developing this program, several intermediate objectives must be met. The first involves developing the analysis program required to define geometry and analyze simple- and continuous-span timber bridges. Such bridges are subjected to the loads prescribed in the *AASHTO 15th Edition of the Standard Specification for the Design of Highway Bridges* (AASHTO 1992), and in the *AASHTO First Edition of the LRFD Bridge Design Specifications* (AASHTO 1994). Hereafter, these are referred to as the ASD and LRFD specifications, respectively. Once the analysis routines are in place, the second step constitutes developing the design and review algorithms to perform the AASHTO checks, or where appropriate, design the section properties. The third intermediate objective is to develop the subroutines necessary to output possible material lay-ups and quantities. Finally, the procedures required to produce computer-aided-drafting (CAD) graphics must be developed so that common CAD systems can access the results.

## Timber Bridge Superstructures

### Descriptions

This project is concerned with timber bridge superstructures only. The five basic types are the beam, deck (slab), truss, arch, and suspension superstructures. This project is limited to beam and deck bridges. The beam and deck bridges may be further divided into the following categories: the stringer (beam) bridge, the longitudinal deck bridge, and the stress-laminated deck bridge. Even these three types can vary considerably in their use of different materials and geometries. To clarify the bridge configurations supported by this project, each specific type is described in the following sections.

### Analysis Methods and Assumptions

#### **Calculation of Load Effects for Beams -**

Analysis of beams is performed with the use of classical beam theory and the finite element method. Inherent assumptions are that behavior is linear elastic, plane cross sections remain plane, deformations are relatively small, beams are prismatic,

and beam supports are infinitely rigid. First, a stiffness model is generated with ten beam elements per span, and influence functions for shear, moment, and translation are determined. These functions are used to evaluate the load effects due to the component and wearing surface uniform dead loads. Next, the appropriate design vehicles are used to determine the critical live load actions at tenth points of each span.

The stiffness method is well suited to beam analysis for several reasons. First, continuous spans are easily accommodated. The ability to analyze continuous spans is one of the justifications for developing Madero, since simple span analysis can be performed relatively quickly by hand. Another advantage is the flexibility of span end conditions. Boundary conditions are defaulted to pinned connections at all bridge supports, but the user has the option to modify them. Both translation and rotation can be released or fixed at any or all supports, and hinges (moment releases) can be placed at any tenth point along the spans. Lastly, the stiffness method allows for easy determination of critical load effects. Moment, shear, and translation are calculated for each tenth point and for each dead and live load. Because a critical combined loading effect could occur at any point along a given beam, all load effects for each tenth point are needed. Critical effects can then be determined by checking each load permutation.

**Load Combinations** - After the load effects due to each type of load are determined, the effects must be combined in a rational manner in order to investigate various load combinations and find the critical case. AASHTO-ASD, section 3.22, defines several *load groups* which are applicable to particular site conditions, loadings, or structure types. Of these, three groups commonly apply to timber bridges: Group I for standard highway loads equal to HS20 or greater, Group IA for loads less than HS20, and Group IB for overload vehicles. Each load group has a dead load factor, a live load factor, and a factor to adjust the allowable stress, so that the basic load combination equation can be written as

$$\text{Factored Load Effect} = \beta_{\text{dead}}D + g\beta_{\text{live}}(IM L) \quad (1)$$

where:

- $\beta$  are the load factors,
- $g$  is the live load distribution factor,
- $IM$  is the impact factor,
- $D$  is the dead load effect, and
- $L$  is the live load effect.

AASHTO permits the impact load to be zero for all timber structures when using allowable stress design, so the impact factor is set to unity by default. As with the distribution factors, however, the program user has the ability to custom define any of the above factors, including impact, when it is deemed appropriate. A complete list of factors and their values can be found in the ASD specification (AASHTO 1992).

The LRFD specification defines several *limit states* which serve to control the combination of loads in much the same way as the ASD load groups. For timber bridges, there are again three limit states which generally apply: the Strength I Limit State for normal vehicular use, the Service I Limit State for deflection considerations, and the Strength II Limit State for special design or permit vehicles. In addition to live load factors, the LRFD method also includes factors for importance, ductility, redundancy, and multiple presence of live load. Standard loading also calls for the simultaneous application of the design truck and a uniformly distributed lane load, each with separate impact factors. The general format for the LRFD load combination equation is

$$\text{Factored Load Effect} = \eta_{imp}\eta_{duc}\eta_{red} [\gamma_{dead}D + mg\gamma_{live}(IM_{truck}L_{truck} + IM_{lane}L_{lane})] \quad (2)$$

where:

- $\eta_{imp}$  is the importance factor,
- $\eta_{duc}$  is the ductility factor,
- $\eta_{red}$  is the redundancy factor,
- $\gamma$  are the load factors,
- $g$  is the live load distribution factor,
- $IM$  are the impact factors,
- $m$  is the multiple presence factor,
- $D$  is the dead load effect, and
- $L$  are the live load effects.

Again, all factors may be manually defined by the user, but standard AASHTO values are used by default. For a more complete list of factors and their values, the reader is referred to Section 3 of the LRFD specification (AASHTO 1994).

It was noted previously that Madero does not consider loads from sources other than structure weight and design vehicles. Evaluation of loads from pedestrians, earth pressure, and stream current may be required. Local conditions may also warrant the use of loads caused by snow, earthquakes, or other environmental effects.

## Program Design and Architecture

### Functional Requirements

As stated earlier, the main objective of this project is to develop a comprehensive computer program for the analysis, design, drafting and rating of wood bridge superstructures. In meeting this end, several program requirements must be considered to ensure a workable, maintainable, and user-friendly product. This section outlines the main functional requirements and describes their importance to the operation of the program.

**Structural Analysis and Review** - Structural analysis is a fundamental task for any bridge design software. Madero's analysis and review are performed by a FORTRAN program referred to as the *computational code* throughout this paper. In addition to the many required analysis procedures, which were previously described, Madero must have procedures for rating bridges. Calculation of design ratios and rating factors produce criteria for rating acceptable performance and making design decisions. Consideration must also be given to the ability of the analysis and review code to interact with the user. The main philosophy in developing analysis routines is that the program should automatically perform all required calculations per the appropriate specifications, yet allow considerable observation and control by the user. The following discussion centers around the review procedures and user control of the computational code.

The product of structural analysis and the load combination routines is a set of factored load effects. These load effects are converted to stresses and compared to factored resistances. (ASD factors are typically 1.0) This comparison yields a design ratio and a rating factor for each analysis point and load permutation. Design ratios are computed as

$$\text{Ratio} = \text{Resistance} / \text{Total Load Effect} \quad (3)$$

so that a ratio greater than unity indicates acceptable performance for the limit state and loading investigated, and a value less than unity indicates unacceptable performance. The design ratios are calculated for shear stress, flexural stress, bearing stress, and deflection, and can be employed by the user to decide how to modify bridge components for bridge design.

Rating factors are computed as

$$RF = \frac{\text{Resistance} - \text{Dead Load Effect}}{\text{Live Load Effect}} \quad (4)$$

A rating factor greater than unity indicates bridge live load capacity greater than that for which the bridge was analyzed. A rating factor below unity reveals deficiency in live load capacity of the bridge. Rating factors are computed for shear, flexure, and bearing. Deflection is not included since bridges are generally not rated based on this service criterion. Critical ratios and rating factors are determined for a bridge and reported to the user. Considerable detail of output can be achieved if desired, including all computations and output of every step outlined above.

User control is necessary so that AASHTO assumptions or default values may be overridden if desired. In general, the user has two options for changing default data. First, the user may select data from a list of typical values, such as defining a species and grade to describe a particular material. Madero then automatically retrieves material properties, such as bending strength or modulus of elasticity, for that material. Alternatively, the user may wish to directly define all material properties. Other parameters such as load factors or distribution factors can be defined in a similar manner, but by default are automatically calculated per the appropriate AASHTO specification.

Observation refers to the user's ability to track the entire analysis process of the computer. Results from any computer program should be checked by the designer to ensure that the proper input has been defined and that all computations are being performed as expected. Madero gives the user the power of selective observation by organizing output into several different categories. For each calculation performed by Madero, accompanying output can be written that defines input, calculations, and results for that operation. References to specifications are given where appropriate, so that the user can quickly find the correct design criteria and compare hand calculations or estimates to the Madero output.

**Specification Requirements** - Because two AASHTO bridge specifications are current, both needed to be included in the development of Madero. The user may elect to use either the *AASHTO 15th Edition of the Standard Specification for the Design of Highway Bridges* (AASHTO 1992), or the *AASHTO First Edition of the LRFD Bridge Design Specifications* (AASHTO 1994). Requirements of

both are described and contrasted later. It should be noted that both specifications are available in either a U. S. customary units version or a S. I. units version. The user has the option to use either system of units.

**Input Requirements** - The computational code receives input data by reading an ASCII input file. The user can create an input file two ways: directly with a text editor, or with the Windows-based user interface. The advantages of both methods are outlined in the following sections, along with a description of the help system.

**Windows-based Program to Produce ASCII Input** - Madero uses a Graphical User Interface (GUI) to define input data and produce an ASCII input file. A Windows-based GUI is desirable for several reasons. Standard Windows functions, such as editing file management, and access to help systems, have come to be expected in modem software. Use of these functions make Madero compatible with other Windows-based programs, and use a format that is familiar to most computer users. Windows also allows for a dynamic interface. For example, dialogs can assume default command parameters, or a tutorial can walk the user through a typical data definition session. From a development standpoint, a Windows-based GUI is attractive because other GUI's have already been developed for similar bridge engineering software. The C++ code used to develop the user interface for this project is in the public domain and is easily accessible to developers.

**On-line Help** - The help system is a Windows-based hyperdocument that puts all the information in the User's Guide on-line. In addition to providing instantaneous access to program documentation from the GUI, the help system also provides standard WinHelp features such as search, contents, and help history. Hyperlinks point to related topics and figures where appropriate. A special feature of Madero creates an easy way to reference input command definitions. When editing the command file in the GUI, the user can position the cursor within the desired command and press CTRL-F1 to bring up the help topic for that command. Details on the GUI and help system architecture are addressed later.

**ASCII Standalone** - Although the FORTRAN computational code could be called directly by the C++ user interface, Madero uses an ASCII input file as an intermediate step between them. The input file contains a series of commands and command parameters which Madero recognizes and interprets to

define its input data. There are several reasons for using such an input file. First of all, development of the input command language necessitates a well-documented interface that can be easily linked to other systems. That is, any GUI that can convert input to Madero's input language can communicate with Madero. This opens possibilities for linkage to future interfaces and also to other bridge software systems. Second, existing code sets a precedent for developing a GUI that produces ASCII input. Modifying existing code takes much less time than starting a new system from scratch. Lastly, ASCII input allows the user to run the computational code directly from DOS. This is advantageous if the user's system does not have Windows, or if a large number of input files are desired to be run with batch execution.

**Output Requirements** - Described previously, the user may selectively request output for certain analysis calculations. This philosophy is reflected throughout the entire output file generated by the computational code. A basic level of output is standard, with options to expand into greater detail. At a minimum, echoed input, a basic bridge description, and summary reports on member actions, stresses, and rating factors are included in the output file. However, if the user wishes, a detailed bridge description, dead load calculations, distribution factors, influence lines, timber adjustment factors, and detailed reports on member actions, stresses, and rating factors can be output as well. The user also controls which design locations are output. Each beam span, for example, is divided into ten analysis points; results for any or all of the analysis points can be output at the user's discretion.

**Maintainability** - Maintainability refers to the ease with which the program can be modified and updated in the future. In order for Madero to be a valuable design aid, it needs to operate according to the most recent AASHTO specifications. While every effort has been made to include the newest design requirements, these requirements will most certainly change as new research findings point to better design methods and specifications are periodically updated. An emphasis on modular design generally creates a more manageable piece of software.

An effort was made throughout the planning and programming processes to keep subroutines small and functions as general as possible. Such modularization has several benefits. First, and perhaps most importantly, the program is divided into definite steps so that the logic is readily apparent. A clear algorithm

is necessary for the programmer to initially organize the program, and it makes for easy addition or modification as the code develops. For example, the procedures created for the ASD version made a basic framework that could be duplicated and edited to produce the LRFD routines. The final code is also more readable, so that for future modifications it can be more easily understood and subsequently edited. Another important concern is the adaptation of code from within Madero to other applications. The exclusive use of local variables makes each subroutine an independent unit, so that it may be extracted for use in another code. A program that performs timber design, for example, could take advantage of Madero's routines for handling material properties, timber adjustment factors, and associated input commands.

The philosophy of modular design applies to data structures equally as well as it does to subroutine structures. The program requires a relatively large amount of data to fully describe a bridge, including its geometric properties, material properties, loading properties, and analysis results. To efficiently transfer this data between subroutines, several large data structures were created. A data structure may conveniently be passed from one subroutine to another in the form of a single variable name, but the main advantage lies in the way each structure is defined. A given data structure definition appears only in one file, and that file is referenced by all subroutines which require the use of that particular data structure. When a new variable must be added, or when a data structure must be otherwise modified, only the single definition needs to be changed. All references to the definition are automatically updated.

**Linkage to Other Rating and Design Systems** -

As stated earlier in the introduction to this paper, very few computer programs are available to assist in designing timber bridges. Large software systems, such as those being developed for comprehensive bridge rating, will require timber analysis as well as the more traditional concrete and steel. Therefore, due consideration should be given to the ease with which Madero can be linked to other rating and design software systems. The computational code can act as an independent analysis tool, given the proper format of input data. In fact, any external program could use Madero's analysis routines if its data were translated into the Madero command language for input. Output data from Madero could also be converted to the desired format.

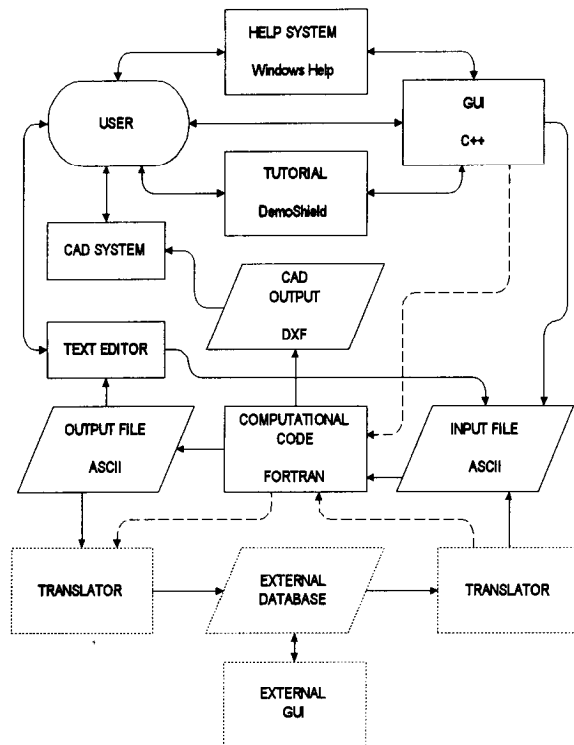
**On-line Tutorial** - In order to aid the user in learning how the Madero GUI works, an on-line tutorial is accessible from the help menu. The tutorial steps through a typical stringer bridge rating example, allowing the observer to see exactly how to enter command parameters and perform other necessary functions from the GUI. The tutorial is a DemoShield@4 demonstration, and works by opening up Madero and running mouse macros that interact with the program just as a bridge designer would. Notes and explanations appear as the demonstration unfolds, and the observer is allowed to control the pace.

### Overview of the Architecture

Madero consists of several subsystems designed to facilitate user interaction, structural analysis, and easy referencing of user documentation. The system has a global architecture, in that the subsystems fit together and relate to one another in a certain fashion. Each subsystem also has a local architecture which describes how it performs its main functions. Obviously there exist many levels of complexity, each of which could be thoroughly described and flowcharted, down to the last subroutine. This section does not intend such an exhaustive explanation. Instead, the general arrangement of the program is presented, so that the relationship between its functionality and its structure may become apparent.

**Overall Architecture** - From the broadest perspective, Madero is a linkage of several different major subsystems. As shown in Figure 1, the relationships among these subsystems are by no means linear. In fact, there are two-way communication channels and alternate paths between many modules. Figure 1 is divided into two major portions: the solid elements represent subsystems that are required for Madero to work, and the dotted elements demonstrate how an external program could link to Madero's computational code. For both, the solid arrows point along conventional paths, and the dashed arrows point to alternate routes that could be developed in the future.

The user has direct access to the GUI, tutorial, help system, and text editor. Two observations can be made by the relationships shown among these modules. First, note that the help system, tutorial, and GUI have direct access to each other, so that the user has the freedom to move back and forth between them with ease while defining the input data. Second,



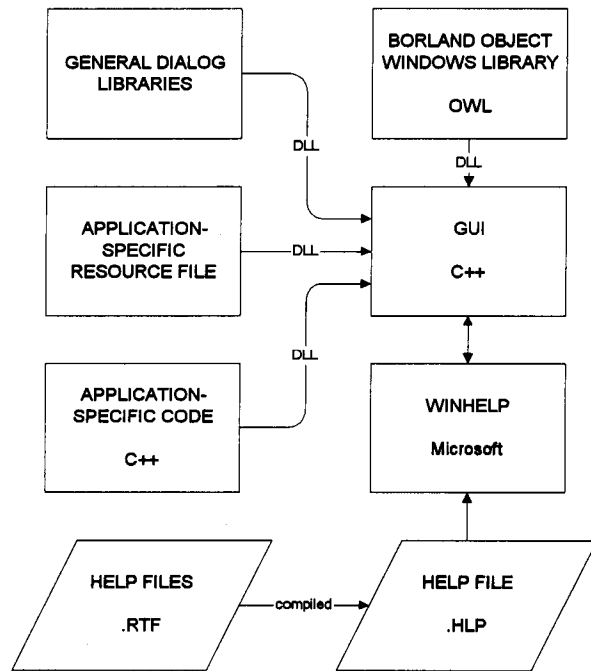
**Figure 1 - Overall architecture.**

notice how both the GUI and the text editor point to the input file. These two paths show the two methods for creating an ASCII input file. From the input file, the path goes through the computational code and into the output file. A trip to the text editor views the output and completes the loop required for a single bridge analysis. The dashed arrow drawn from the GUI directly to the computational code skips over the input file. This represents a possible future path which passes an argument list of data structures from the C++ GUI to the FORTRAN computational code without the use of the Madero command language. The FORTRAN is in fact being directly called by the C++ GUI to create a seamless and more efficient execution of the computational code.

The external program links to the computational code in the same manner as the Madero modules: through the input and output files. As explained previously, the external database need only be translated into the Madero command language to be compatible with the computational code. Such a connection could be used to link Madero with other bridge rating and design systems, such as AASHTO Virtus™ or AASHTO Opus™. The dashed arrows show a direct link between the translators and the computational code. As was proposed with the Madero system, an external

system could bypass the input and output files if the appropriate data structures were defined and directly passed into the FORTRAN.

Figure 2 shows the major components of the graphical user interface and help system. Both systems take advantage of existing code or utilities to keep new software development to a minimum.



**Figure 2 - GUI and help system architecture.**

The GUI uses a system of dynamic links to create a complete environment for data input. Written in C++, the GUI is comprised of four main components: the application-specific code, application-specific resource, general dialog libraries, and Borland object windows library. The dialog libraries were developed under a previous project sponsored by the U. S. Department of Transportation, and contain code especially suited to dialog development. The Borland object windows library contributes general development tools. Particular to Madero are the resource file and C++ code written for this project, which were patterned after existing GUI code for similar bridge software.

The help system is a close companion of the GUI in that it documents the command language and provides a general guide for using the GUI. Data files for the help system were created on Microsoft Word and saved in rich text format (RTF). These files were then compiled into help file format (HLP) with the aid of

the Microsoft help compiler. The GUI accesses the help system through Microsoft Windows Help, which is the standard for Windows-based applications. Since the Madero help files are in standard format, they can be opened from the help system in any Windows application.

**Validation Procedures** - To ensure the most error-free code possible, Madero has been subjected to many levels of testing throughout the software development. The computational code requires the most rigorous tests because of the numerous areas of coding where computations could deviate from those intended. Individual processes were checked for acceptable performance as they were developed. For example, the routines for loading material property data into data structures were tested by creating a special output file for debugging. Once adequate processes were established as passable, testing began on bridge rating for specific bridge types. An effort has been made to cover all program functions over the course of the entire verification problem set. Each problem was compared to a known solution so that results could be validated. All verification problems matched known solutions for the majority of computations. In a few instances some discrepancies were observed, but these were carefully explained, typically with reference to modifications in the AASHTO specifications.

The GUI and help systems were likewise tested for acceptable performance. In the GUI, the dialog for each command and its accompanying output string had to be verified as complying with the Madero command language. For the help system, all hyper links had to be checked to make sure paths were correct. These tests, assuming that every part of the system was checked, confirm that Madero is capable of bridge analysis through the GUI. However, much testing remains in the areas of user-friendliness and program robustness. The validation process is an ongoing one, and the program will not be in a release state until extensive beta testing has been performed by testers independent of the program development. Certainly many points brought up by beta testers, but not considered by the developers, will require modifications to the code in the future.

### **Approximate Transverse Deck Analysis for LRFD**

Unlike the ASD specification, the LRFD specification requires transverse deck strips to be modeled as continuous beams. Two complications result. First, the critical positioning of the wheel loads is not readily

apparent. For a simple span, the critical moment from a single wheel load occurs when the load is placed at midspan. The same holds true for continuous spans. However, the single-wheel analysis case never occurs in continuous beam modeling because both wheel loads on an axle will always contribute to the load effect. In other words, no matter where a load is placed, it will affect the moment experienced by every part of the deck. Critical loading then, is a function of bridge and truck geometry and is best determined by trial and error. Second, each wheel load needs to be distributed across a 20-inch footprint. Where small spans are concerned, the effects of wheel load distribution over the footprint can be significant.

### Development of Analysis Functions

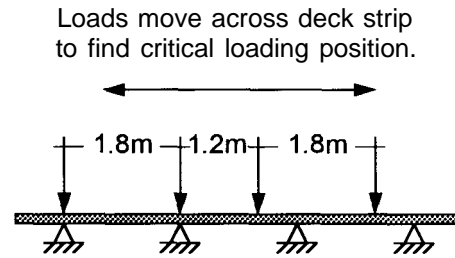
The goal was to create approximate expressions for maximum and minimum moment and deflection that are applicable to geometries and loadings commonly encountered in stringer type timber bridges. To meet this end, several variational parameters were investigated, including number of supporting beams, deck span, and number of lanes loaded.

The approximate expressions for moment and deflection are applicable only to deck spans between 0.9 and 2.1 meters (typical for timber). In addition, the following geometric restrictions apply:

- the number of stringers is no less than 3,
- stringers are spaced evenly, and
- the overhanging part of the deck, measured from the centerline of the outside stringer to the face of the curb, is less than or equal to either (0.625 times the deck span minus 0.5 meters), or (0.3 meters), whichever is greater.

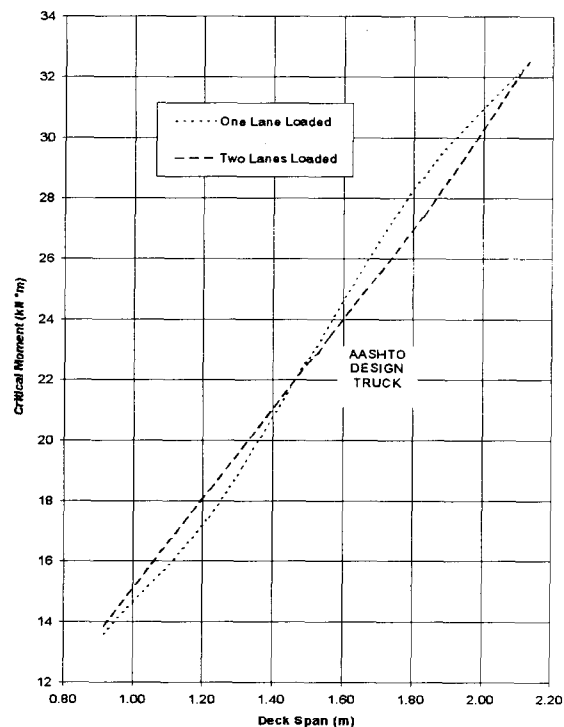
The overhang limitation is intended to keep the critical design section within the main deck spans and prevent the overhang from controlling. The criteria are adapted from USDOT qualifications for concrete decks (National Highway Institute 1995).

Each model consists of a deck strip supported by three or four beams and loaded with standard design truck axle weights, as illustrated by Figure 3. Rotational and translational restraints on the outside supports are removed to represent the overhang at each end of the deck. Live loads move across the deck strip, and load effects are computed at the tenth points of each span.



**Figure 3 - Continuous deck strip analysis model**

In order to create conservative analysis functions, mathematical expressions were developed that fit the results of the finite element analyses. The plots shown in Figures 4 and 5 summarize the critical moment and deflection values from the analyses. Results from either the 3- or 4- beam models are shown, whichever is more critical. Moment is considered at all points along the deck. Deflection, however, is checked only within the main deck spans and is ignored in the overhangs. Where the critical moment or deflection is negative, the absolute value is used. In other words, Figures 4 and 5 show the extreme load effects as a function of span, whether positive or negative, plotted on a positive scale.



**Figure 4 - Critical moment for transverse timber decks - LRFD.**



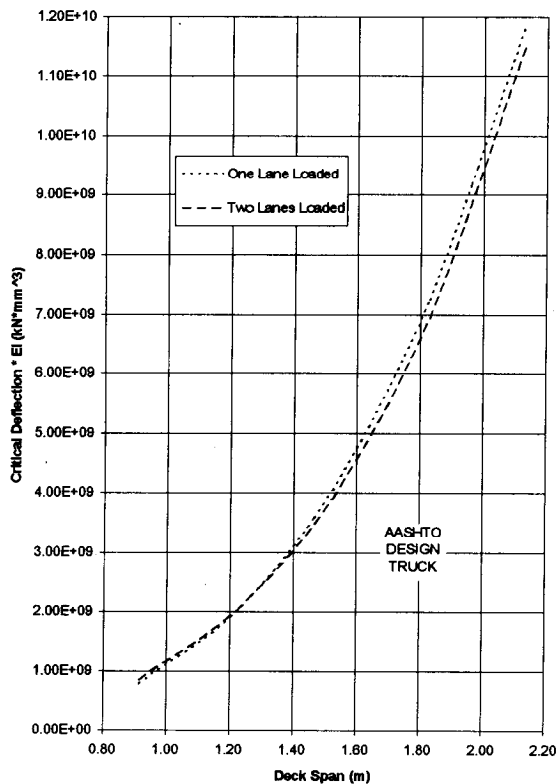


Figure 5 - Critical deflection for transverse timber decks - LRFD.

#### Patch Factor for Flexure

A *patch factor* is used to compensate for the difference between the effects of the distributed load and the assumed point loading in the development of the approximate analysis equations. The patch factor is based on the relationship between the moment due to a point load on a simple span and the moment due to a distributed load on a simple span. While this method is exact for simple spans, it is approximate for continuous spans. The patch factor is defined as

$$PF = 1 - \frac{b_t}{2s_s} \quad (5)$$

where

$PF$  = patch factor,  
 $b_t$  = tire width, and  
 $s_s$  = stringer spacing

and it is applied to all moment calculations.

Equation 5 can be used in conjunction with Figures 4 and 5 to create a design aid for LRFD transverse

timber deck design. Direct finite element analysis is avoided while still meeting the requirements of the LRFD specification. Applicability is limited to decks loaded with the AASHTO design truck that conform to the geometric restrictions described earlier. A detailed description of design aid development and discussion of the approximate methods can be found in (Thomas 1996).

#### Closing Remarks

The development of Madero has realized several important accomplishments. Most central to its purpose, the program is a working design tool for common types of timber bridges. Madero performs checks on critical shear, bending moment, bearing, deflection, and deck connections for simple- and continuous-span bridge components. The bridge designer has the flexibility to use ASD or LRFD design specifications in either U.S. customary or SI units. Another significant accomplishment is the program's ability to produce a generous amount of output upon request. Detailed output provides a means for the bridge designer to verify all calculations and understand any assumptions made by the program. As a final contribution, a set of design aids (described elsewhere, Thomas 1996) was created through the development of approximate analysis techniques for transverse decks. The continuous strip analysis required by the LRFD specification is more complicated than the simple-span analysis permitted in previous editions of the AASHTO specification. The design aids provide a quick and simple source for transverse deck analysis of standard design truck loads, while still meeting AASHTO requirements.

#### References

- AASHTO. 1994. AASHTO LRFD Bridge Design Specifications, 1st ed. Washington, D. C.: American Association of State Highway and Transportation Officials.
- AASHTO. 1992. Standard Specifications for Highway Bridges, 15th ed. Washington, D. C.: American Association of State Highway and Transportation Officials.
- AASHTO. 1991. Guide Specifications for the Design of Stress-Laminated Wood Decks. Washington, D. C.: American Association of State Highway and Transportation Officials.

AF&PA/AWC. 1994. Wood Products Metric Planning Package. Washington, D. C.: American Forest & Paper Association / American Wood Council.

NHI. 1995. Load and Resistance Factor Design for Highway Bridges, National Highway Institute Course No. 13061, Participant Notebook. Washington, D. C.: U. S. Department of Transportation, Federal Highway Administration. 11- 12.

Ritter, M. A. 1990. Timber Bridges, Design, Construction, Inspection and Maintenance. EM7700-B. Washington, D. C.: United States Department of Agriculture, Forest Service.

Thomas, W. R. 1996. Computer Program for the Analysis, Design, Rating, and Drafting of Wood Bridge Superstructures. Master's Thesis, University of Wyoming, Laramie, Wyoming.

### **Acknowledgments**

This development project was funded through a contract between the USDA Forest Products Laboratory and the Department of Civil and Architectural Engineering at the University of Wyoming. Their assistance is greatly appreciated.

In: Ritter, M.A.; Duwadi, S.R.; Lee, P.D.H., ed(s). National conference on wood transportation structures; 1996 October 23-25; Madison, WI. Gen. Tech. Rep. FPL- GTR-94. Madison, WI: U.S. Department of Agriculture, Forest Service, Forest Products Laboratory.