Validation of Bridge Engineering Computations

Assuring Bridge Software Users

MARK MLYNARSKI AND JAY PUCKETT

M. Mlynarski is Project Manager, Bridge Software Group, Michael Baker Jr., Inc., Coraopolis, Pennsylvania, and J. Puckett is Professor of Civil Engineering, University of Wyoming and President of BridgeTech, Inc.

Bridge engineers rely on automated computations to increase productivity and therefore need procedures for assuring error-free software. The large number of bridge types, geometric configurations, materials, and loadings creates a challenge for software developers and end users to provide this assurance.

Whether a spreadsheet or a three-dimensional finite element analysis and design tool, automation of bridge computations is an integral part of the design engineer's daily routine. Bridge designers are and will be using software to design bridges based on new specifications.

In theory, this software should be error-free—yet perfect software remains elusive. The number of bridge types, geometric configurations, materials, and loadings creates a large domain of solutions.

Research conducted under NCHRP Project 12-50 provides a standardized process that is useful for a host of applications in bridge engineering, specification development, and software development and maintenance.

Problem

The quality and quantity of new bridge software and specification testing is unknown, creating a barrier to acceptance. Bridge owners lack the resources for lengthy validations of bridge software. Software developers and specification writers need a standardized process to validate and report results.

The large number of bridge types, geometric configurations, materials, and loadings creates a challenge for software developers and users. Only limited independent validation of the software can be performed.

The *AASHTO LRFD Bridge Design Specifications* require verifying computer programs against the results of closed-form solutions, physical testing, or previously verified computer programs. This is difficult, because closed-form solutions are generally limited to the most trivial cases, physical testing is expensive, and few—if any—computer programs for bridges have completed a formal validation procedure.

Solution

The researchers have systematically identified both the range of practical input values and possible outputs for bridge design and analysis. To make the solution manageable, the process was divided into smaller subdomains, such as dead-load distribution and live-load actions. Twenty subdomains were identified for bridge superstructures. A test-bed of bridge data with well-defined parametric inputs and outputs was developed. The data were created to rigorously test the limits of the associated subdomain. These bridges or portions of bridges are usable by developers, end users, and others. The researchers suggest that at the completion of this project, in the second quarter of 2002, the data sets should be made available through a website or compact disc.

A careful review of the design specifications catalogued the possible results of computations. For superstructures, nearly 900 possible outputs were identified, such as exterior girder slab weight and deflection of steel I-sections due to truckloads. Each output was assigned a unique report identifier, permitting efficient review and comparison of results.

Finally, to implement the validation process, modifications were made to the LRFD computer program to produce a simple comma-delimited ASCII text file of outputs that can be imported into a relational database. The database values can be compared efficiently with the results from another program that has been modified to produce a similar comma-delimited file.

Figure 1 illustrates a comparison of results from Pennsylvania Department of Transportation (DOT) Prestressed LRFD and Wyoming BRASS Girder LRFD. In this case, the only difference between the programs appears to be the spacing of calculated values along the length of the girder. The process is particularly valuable in the development of software for the *AASHTO LRFD Bridge Design Specifications* and in the review of proposed amendments to the specification provisions.

Application

The validation of the concept and the research approach used design programs from Pennsylvania, Wyoming, Washington, and Alaska DOTs. The product, *Bridge Software Validation Guidelines and Examples*, NCHRP Project 12-50, provides a standardized process for a host of applications in bridge engineering, specification development, and software development and maintenance. These applications include

- ◆ A systematic method for comparing and evaluating bridge design and analysis software,
- ◆ A standardized report format for presenting and comparing results for a specific bridge design, and
- ◆ A powerful method for formally reviewing specification changes.

Benefits

The research illustrated important benefits:

1. Previously unknown errors were detected.
2. Validation of the programs was achieved in significantly less time and with much more rigor—months of work can be completed in hours.
3. Hundreds of test problems are now economically possible—when previously only tens were used. Moreover, the problems are more robust and address all specifications as well as the usual bridge configurations.
4. The standardized results format is likely to evolve into an industry standard.

As a result of the demonstrated benefits, AASHTO has decided to implement the NCHRP 12-50 process in the AASHTOWare computer software bridge products now being developed for load rating and design (Virtis and Opis).

Design Comparisons

Using the NCHRP 12-50 methods, a designer easily can compare the results of alternative computational processes. The results can be imported into a common viewer for comparison, making the differences apparent. In the past, validation typically has consisted of producing a manual example that was computed and compared with results of the program. A typical manual example could take several days to several weeks to complete, producing a set of results representing a single bridge structure. The NCHRP 12-50 process has generated and compared dozens of examples in the same amount of time.

In addition to the time saved producing the examples, the NCHRP 12-50 process also realized savings in maintenance. Manual example problems are almost as costly to maintain as they are to produce. Changes in specifications or procedures often can lead to major revisions in the manual computations. The NCHRP 12-50 process simply reexecutes the automated processes.

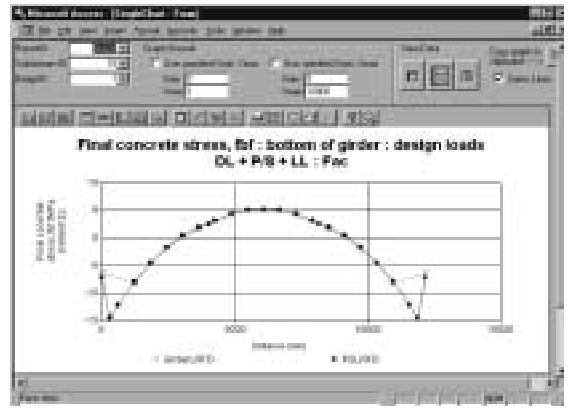

FIGURE 1 Computer program compares final concrete stresses from Pennsylvania DOT Prestressed LRFD and Wyoming BRASS Girder LRFD.

Specification Review

Specification writing committees can use the software to compare different versions of the specifications on a large set of bridges to determine if the changes accomplish the desired objectives, and to prevent problems from developing. Similarly, bridge engineers can see the consequences of the changes on current engineering practices.

Software Validation

The cost of rigorous validation of software is 25 percent or more of the development cost. It is higher for engineering applications that involve manual computations. The cost associated with improperly functioning software that is unserviceable or that crashes can be large. Software developers may use the NCHRP 12-50 process for verification and regression testing (version testing) of their software and to reduce verification times and costs.

Michael Baker Jr., Inc., in association with BridgeTech, Inc. and Modjeski and Masters, Inc. performed NCHRP Project 12-50. For further information contact Mark Mlynarski, Michael Baker Jr., Inc., 420 Rouser Road, Coraopolis, PA, 15108-2722 (telephone 412-269-7933, e-mail mmlynarski@mbakercorp.com).

EDITOR'S NOTE: Appreciation is expressed to David B. Beal, Transportation Research Board, for his efforts in developing this article.

Suggestions for "Research Pays Off" topics are welcome. Contact G. P. Jayaprakash, Transportation Research Board, 2101 Constitution Avenue, NW, Washington, DC 20418 (telephone 202-334-2952, e-mail gjayapra@nas.edu).