Object Oriented Programming for
Reinforced Concrete T-beam Bridge Design

by
Yaling Li

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in
Civil Engineering

Approved:

K. B. Rojiani

R. M. Barker  D. A. Garst

June, 1995
Blacksburg, Virginia
Object Oriented Programming for
Reinforced Concrete T-Beam Bridge Design

by
Yaling Li

Committee Chairman: Dr. Kamal B. Rojiani
Charles E. Via, Jr. Department of Civil Engineering
Virginia Polytechnic Institute and State University

ABSTRACT

This study considers the application of object oriented programming approach to
develop a Windows based program for the design of reinforced concrete T-beam bridges
in accordance with the newly adopted AASHTO LRFD Bridge Specification. The
program can perform both the analysis and design of the deck and beam. The program is
interactive in nature and it is possible to change all input data and design parameters
during execution.

A series of second-degree interpolating polynomial functions were developed for
representing influence line coordinates for various load effects in the deck and beam
subjected to moving loads. The analysis results using the program were compared with
those obtained from the commercial programs. The design results obtained from the
program were compared with example problems for several test structures. Both design
and analysis comparisons gave satisfactory results. The successful performance of the
program demonstrates the significant advantages of the object oriented programming. It
was concluded that the benefits of object oriented programming approach make this a
viable technique for the development of computer applications for structural engineering.
Acknowledgments

I wish to express my sincere thanks to Dr. Kamal Rojiani, my advisor and committee chairman, for his continuous advice, guidance and encouragement. It is largely due to his helpful assistance and suggestions throughout this study, that I have been able to complete this thesis.

Special thanks go to Dr. Richard M. Barker who gave me considerable advice and support on this project, provided the verification examples for the project which helped me and also served as a member of my thesis committee. I will remember his kindness and friendship for an international student forever.

I would like to thank Prof. Don A. Garst for all the kind help he gave me during my graduate studies at Virginia Polytechnic Institute and State University and for serving on my thesis committee.

Finally, I would like to express my deepest gratitude to my parents for their patience and encouragement, to my husband for his love and support and to my son for giving me his love and confidence. Without them, it is would not have been possible for me to finish my Master’s degree in the United States.
# Table of Contents

Acknowledgments ........................................................................................................ iii

Table of Contents .......................................................................................................... iv

List of Figures ................................................................................................................ viii

List of Tables .................................................................................................................. x

Chapter 1  Introduction ................................................................................................ 1

1.1 Introduction ............................................................................................................ 1

1.2 Scope of Study ....................................................................................................... 2

1.3 Organization .......................................................................................................... 3

Chapter 2  An Overview of Object Oriented Programming ...................................... 4

2.1 Introduction .......................................................................................................... 4

2.2 Procedural Programming ..................................................................................... 4

2.3 Object Oriented Programming ............................................................................. 5

2.3.1 Object Oriented Approach ............................................................................ 5

2.3.2 Classes and Objects ....................................................................................... 6

2.3.3 How Does Object Oriented Programming Work ....................................... 7

2.4 Characteristics of Object Oriented Programming ............................................... 7

2.4.1 Encapsulation ............................................................................................... 7

2.4.2 Abstraction ................................................................................................... 8

2.4.3 Inheritance ................................................................................................... 8

2.4.4 Polymorphism .............................................................................................. 9

2.5 Advantages and Disadvantages of Object Oriented Programming ............. 10
Chapter 3  Design Procedure for Reinforced Concrete T-Beam Bridge

Superstructures ............................................. 14

3.1 Introduction ............................................. 14

3.2 Overview of Design Procedure ............................... 14

3.3 Influence Line Equations .................................. 17
    3.3.1 Muller - Breslau's Principle .......................... 17
    3.3.2 Influence Line Equations ............................... 18

3.4 Deck Design ............................................. 24
    3.4.1 Deck Input Data ....................................... 24
    3.4.2 Deck Design Procedure ................................. 25
    3.4.3 Deck Design Results ................................... 27

3.5 Beam Design ............................................. 28
    3.5.1 Beam Input Data ....................................... 30
    3.5.2 Beam Design Procedure ................................. 30
        3.5.2.1 Assumptions for Beam Design ...................... 30
        3.5.2.2 Distribution Factor Method ......................... 31
        3.5.2.3 Calculation of Live Load Effects ..................... 33
        3.5.2.4 Crack Control .................................. 37
        3.5.2.5 Fatigue Checking ................................ 38
        3.5.2.6 Beam Deflection ................................ 39
        3.5.2.7 Flexural Capacity ............................... 40
Chapter 4 Program Architecture ................................................. 45
  4.1 Introduction ....................................................................... 45
  4.2 Structural Classes .......................................................... 45
      4.2.1 Class Structure ....................................................... 46
      4.2.2 Class Inline .......................................................... 48
      4.2.3 Class Deck ........................................................... 49
      4.2.4 Class Beam .......................................................... 51
  4.3 Interface Classes ............................................................ 52
  4.4 Program Operation .......................................................... 55
      4.4.1 Main Menu ............................................................ 55
      4.4.2 File Menu ............................................................ 56
      4.4.3 Title Menu ........................................................... 57
      4.4.4 Material Menu ....................................................... 57
      4.4.5 Load Menu ........................................................... 58
      4.4.6 Deck Menu .......................................................... 59
      4.4.7 Beam Menu .......................................................... 62
      4.4.8 Design Menu ........................................................ 64
      4.4.9 Results Menu ......................................................... 67
      4.4.10 Result Windows ...................................................... 67
      4.4.11 Help Menu .......................................................... 70

Chapter 5 Results and Discussions ............................................. 72
5.1 Introduction ........................................................................................................ 72
5.2 Comparison of Deck and Beam Design Results ........................................... 72
  5.2.1 General Description .................................................................................. 73
  5.2.2 Section Selection ...................................................................................... 74
  5.2.3 Comparison of Design Results ................................................................. 74
5.3 Comparison of Continuous Beam Analysis Results .................................... 84
  5.3.1 General Description ................................................................................ 84
  5.3.2 Loading .................................................................................................... 84
  5.3.3 Distribution Factor .................................................................................. 86
  5.3.4 Comparison of Analysis Results ............................................................... 86
5.4 Discussions of Results .................................................................................... 88

Chapter 6 Summary and Conclusions ................................................................. 90
  6.1 Summary ...................................................................................................... 90
  6.2 Conclusions ................................................................................................ 91

References ........................................................................................................... 92

Appendix A Class Definitions .............................................................................. 94
Appendix B Output Files for Three Test Structures ........................................ 119
Appendix C Comparison of Influence Line Equations with RISA-2D ............. 132

Vita ....................................................................................................................... 138

Table of Contents vii
List of Figures

Figure 3.1 Location of maximum effects for a continuous deck or beam .......... 18
Figure 3.2 General shape of influence lines for different effects .................. 19
Figure 3.3 The maximum positive moment position .................................. 26
Figure 3.4 The maximum negative moment position ................................. 26
Figure 3.5 Live load distribution for moment calculation ......................... 34
Figure 3.6 Live load distribution for shear calculation ............................. 35
Figure 4.1 Main Menu ........................................................................... 56
Figure 4.2 Project Name Dialog Box ......................................................... 57
Figure 4.3 Material Strength Dialog Box ............................................... 58
Figure 4.4 Load Modifiers Dialog Box ..................................................... 59
Figure 4.5 Deck Description Dialog Box .................................................. 60
Figure 4.6 Error Message for Deck Description Dialog Box .................... 60
Figure 4.7 Deck Thickness Dialog Box .................................................. 61
Figure 4.8 Deck Cover and Bar Diameter Dialog Box .............................. 61
Figure 4.9 Multi-span Continuous Beam ................................................. 62
Figure 4.10 Beam Description Dialog Box ............................................... 63
Figure 4.11 Beam Size Dialog Box .......................................................... 63
Figure 4.12 Beam Cover and Stirrup Size Dialog Box ............................... 64
Figure 4.13 Deck Design Dialog Box ...................................................... 65
Figure 4.14 Beam Design Dialog Box ...................................................... 66
Figure 4.15 Bridge Structure Description .................................................. 68
Figure 4.16 Deck Design Results Windows .............................................. 68
Figure 4.17 Beam Design Results Window .............................................. 69
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.18</td>
<td>Help / About Dialog Box</td>
<td>70</td>
</tr>
<tr>
<td>5.1(a)</td>
<td>Deck geometry for the first test structure</td>
<td>73</td>
</tr>
<tr>
<td>5.1(b)</td>
<td>Beam geometry for the first test structure</td>
<td>73</td>
</tr>
<tr>
<td>5.2(a)</td>
<td>Deck geometry for the second test structure</td>
<td>81</td>
</tr>
<tr>
<td>5.2(b)</td>
<td>Beam geometry for the second test structure</td>
<td>81</td>
</tr>
<tr>
<td>5.3</td>
<td>Continuous beam geometry</td>
<td>84</td>
</tr>
<tr>
<td>5.4</td>
<td>Applied loads on continuous beam</td>
<td>85</td>
</tr>
</tbody>
</table>
List of Tables

Table 3.1 Default values for material properties .................................................. 16
Table 3.2 Default values for load modifiers ......................................................... 16
Table 3.3 Default values for cover and bar size for deck steel .......................... 16
Table 3.4 Default values for cover and stirrup size for beam steel .................. 16
Table 3.5 Influence line coordinates in continuous beams with different span number by RISA-2D ................................................................. 22
Table 5.1 Comparison of deck analysis results for the first test structure .......... 75
Table 5.2 Comparison of deck design results for the first test structure ............ 76
Table 5.3 Comparison of beam analysis results for moments for the first test structure ......................................................................................................................... 77
Table 5.4 Comparison of beam analysis results for shears for the first test structure ......................................................................................................................... 78
Table 5.5 Comparison of beam design results for the first test structure .......... 79
Table 5.6 Comparison of deck design results for the second test structure ...... 82
Table 5.7 Comparison of beam design results for the second test structure ...... 83
Table 5.8 Distribution factors for moment and shear ........................................... 86
Table 5.9 Comparison of beam analysis results for moment for the third test structure ......................................................................................................................... 87
Table 5.10 Comparison of beam analysis results for shear for the third test structure ...................................................................................................................... 88

List of Tables
Chapter 1
Introduction

1.1 Introduction

Most of the existing structural engineering programs were developed using the procedural programming methodology. One of the characteristics of procedural programming is that functions and data are treated as separate entities. As a result, when the problem to be solved is large and complex, the corresponding program becomes very complex and is difficult to maintain. In recent years, there has been considerable interest in the use of the object oriented programming methodology for developing computer programs. The basic concept of the object oriented programming approach is to combine both data and functions into a single entity called an object. Such objects simulate both the characteristics and behavior of real world objects. It has been shown that the object oriented programming approach results in increased programmer productivity and a significant reduction in the effort required for maintaining programs.

An object oriented program typically consists of a number of objects. Each object has its own data as well as needed functions. The significant characteristics of object oriented programming, such as encapsulation, abstraction, inheritance and polymorphism, make it possible to efficiently manage and develop complex engineering applications.

Of all the object oriented programming languages in use today, C++ is most popular and successful. As an extension of C programming language, C++ combines both object oriented programming and traditional procedural programming. Also, several integrated C++ compilers, such as the Microsoft Visual C++ compiler and the Borland C++ compiler, provide class libraries for developing applications in the Windows
environment. These class libraries provide a complete framework and the basic object classes for developing applications.

1.2 Scope of Study

The object oriented programming methodology is a new way of thinking about the problem to be solved on a computer. It adapts the computer to the problem instead of molding the problem to the computer [Pohl, 1993]. Thus, the object oriented programming methodology has many advantages over conventional procedural programming especially for large and complex engineering applications. In recent years, the object oriented programming approach has received a great deal of attention in structural engineering. However, most existing structural programs still use conventional procedural programming languages. More experience is needed in using the object oriented programming approach for the development of structural engineering applications.

As a result of many years of study and research, the AASHTO LRFD Bridge Design Specifications (1994) was published last year. The new AASHTO Specifications uses both U.S. Customary and Metric Units. It provides a more accurate approach for analysis of bridges and improves bridge design methods. Many structural engineers are not yet familiar with the use of the load and resistance factor design philosophy or the use of metric units in designing highway bridge structures and foundations.

Considering the many benefits of object oriented programming and the recent publication of the AASHTO LRFD Bridge Design Specifications, it is necessary and practical to develop a computer program for the analysis and design of bridges using object oriented programming and the new AASHTO Specifications.
The objective of this thesis is to develop a Windows based C++ program for the
design of reinforced concrete T-beam bridges using the object oriented programming
methodology. The program is capable of analyzing and designing the deck and the beam
in accordance with AASHTO LRFD Bridge Design Specifications.

1.3 Organization

An overview of the object oriented programming methodology and its application
in structural engineering are presented in Chapter 2. In Chapter 3, the procedure for the
design and analysis of the superstructure for a T-beam bridge, including the deck and the
beam, in accordance with the AASHTO LRFD Bridge Design Specifications is presented.
Chapter 4 describes the program architecture and provides a description of the major
classes used in the program. This chapter also serves as a users manual for the program.
Chapter 5 contains the comparison of results obtained from the program with those
obtained using other programs and sources. A summary of the research and the major
conclusions of the study are provided in Chapter 6. In Appendix A, a complete definition
for each class used in the program is given. Appendix B contains the design results for
several test structures and Appendix C presents the comparison of influence line equations
with the RISA-2D program.
Chapter 2
An Overview of Object Oriented Programming

2.1 Introduction

In this chapter, an introduction to object oriented programming is given followed by a brief description of the key concepts of object oriented programming. The advantages and disadvantages of object oriented programming compared to procedural programming are discussed. A brief description of some of the more popular object oriented programming languages is presented. This chapter also presents some applications of object oriented programming in structural engineering.

2.2 Procedural Programming

The traditional software development approach is called procedural programming. There are many procedural programming languages in use today. These include BASIC, FORTRAN, Pascal and C. In conventional procedural programs, data and functions are treated as separate entities. To enable functions to access data, the data must be declared to be global, which greatly increases the chances of data being accidentally corrupted by functions. In addition, if the data structure is changed, all functions which use that data must be modified. This can be a difficult task for large programs with many functions. For simple programs, the disadvantages of the procedural programming paradigm are not critical, so procedural programming is still used extensively. However, when the problem to be solved grows larger and more complex, the program becomes very complicated and difficult to maintain [Rojiani, 1994]. Another problem with procedural programming is that the major components of the program, data and functions, do not model the real
world. Therefore, it is difficult for the programmer to visualize the relationship between the program and the problem being solved.

2.3 Object Oriented Programming

The object oriented programming methodology provides a means for managing the complexity of software. In object oriented programming, both data and functions operating on that data are combined together into a single unit called an object. Such objects simulate both the characteristics and behavior of real world objects.

2.3.1 Object Oriented Approach

Object oriented programming methodology simulates a real-world situation on computers. The fundamental idea of the object oriented programming approach is to combine both data and functions into a single entity called an object. Thus, data and its functions are said to be encapsulated into an object. Data encapsulation and data hiding are the most significant departure from procedural programming. As the name indicates, the object is the basic concept behind object oriented programming. Objects rather than commands are the main roles in the object oriented approach.

Object oriented programming is not primarily concerned with the details of operation. It deals with the overall organization of the program. The integrity of each object makes it easier for functions to be written separately. As a result, a large program can be divided into smaller parts, with each part having its own data and functions. This makes it possible for different programmers to work on various parts of a large project independently and makes software development very efficient.

Object oriented programming methodology is a new way of thinking about the problem to be solved on a computer. It adapts the computer to the problem rather than

An Overview of Object Oriented Programming
trying to mold the problem to the computer. Thus, the object oriented programming methodology has many advantages over conventional procedural programming especially for large and complex engineering applications.

2.3.2 Classes and Objects

There are two key concepts, classes and objects, in object oriented programming. A class is a data structure that specifies the data properties and defines the functions that operate on the data. An object is an instance of a class. Objects that have similar instance variables and behavior are grouped and abstracted into one class. Thus, classes act as the template and many objects of the same class can be created. As an example of a class, consider a class for representing a beam in a structure, class Beam. The data members of the Beam class might be the following: beam span L, loads on the beam P, maximum moment in the beam M, maximum shear in the beam V, required area of steel As and required area of stirrups Av.

Examples of some functions that may be part of the Beam class include: CalcMaxM() which calculates the maximum moment in the beam, CalcMaxV() which calculates the maximum shear in the beam, ComputeAsRequired() which computes required area of steel for bending and ComputeSpacing() which computes the spacing of stirrups.

An object beam can be created using the Beam class. The beam object has its own data members L, P, M, V, As and Av.

The data members in a class can be declared as private, protected or public. Private data is accessible only by the functions defined within the same class. Protected data is accessible by functions within the same class and in its derived classes. Public data is accessible to any functions of the same class and its derived classes, and also to
functions outside the class. Typically all data is declared as private so that it can only be accessible by functions within the same class. This ensures that data cannot be altered by other functions in the program.

2.3.3 How The Object Oriented Programming Works

An object oriented program typically consists of a number of objects. Since every object has its own data and functions, when an object needs to perform a given task, it calls the corresponding function to operate upon its own data. For example, a program for designing beams may contain a beam object. Whenever the program has to design a beam, the message is sent to the beam object to perform the desired task. Since the beam object has all the data and functions needed to perform the beam design, one doesn't have to be concerned about the details of how the task is implemented.

2.4 Characteristics of Object Oriented Programming

The most important characteristics of object oriented programming are: encapsulation, abstraction, inheritance and polymorphism. These characteristics are briefly described below.

2.4.1 Encapsulation

In an object oriented programming language, both data and functions operating on the data are combined in one object. Functions within an object can access all data belonging to the same object but other functions cannot access these data directly. Therefore, an object can establish a boundary around itself and safeguard its data from being altered by other functions in the program. The ability of an object to conceal its data from other parts of a program is called encapsulation.
Encapsulation is important, since it prevents data from being inadvertently modified by outside functions. In object oriented programming, if an outside function wants to access the data of another object, it has to make a request to that object to access the data. Encapsulation enhances program changeability and maintainability.

2.4.2 Abstraction

Abstraction is the process of ignoring details in order to concentrate on essential characteristics. An object oriented program consists of a number of objects, which communicate with each other by calling each other's functions. In order to interact with other objects, the object makes its behavior known to other objects. When a function from an outside object requests an action, it does not need to know how the request is fulfilled but simply what information needs to be passed and what information is returned. Thus, objects interact with each other through a clearly defined interface.

While procedural programming languages support abstraction, object oriented programming languages provide much more powerful abstraction mechanisms, which makes writing large programs simpler and makes the program easier to understand.

2.4.3 Inheritance

One feature of object oriented programming that is not found at in all procedural programming is the ability to define a hierarchy of classes. A new class can be derived from an existing class by modifying the properties of the existing class. The original class is called the base class and the newly derived class is called the derived class. The derivation of one class from another class is called inheritance. Inheritance clarifies the relationship among classes.
The base class contains characteristics common to a group of derived classes. The derived class inherits the characteristics of the base class but adds new ones of its own. For example, a simple beam is a beam but it also has certain characteristics not found in all beams. Thus, we can derive a class called SimpleBeam from the class Beam. The derived class will have all of the features of the base class as well as new features that are specific to itself.

One of the practical benefits of inheritance is that derived classes can share code and data from the base class. This reduces the size of the program and make a program easily extensible. In fact, it is this feature of inheritance that makes it possible to use class libraries already developed by professional programmers.

2.4.4 Polymorphism

In a derived class, it is possible to redefine functions in the base class without changing their names. Same function call in the program performs different tasks depending upon the specific type of object involved. This is called polymorphism. For example, suppose a program needs to draw several figures, such as rectangles, circles, lines and polygons. A base class Shape can be declared which contains common characteristics of all these figures. Several classes can be derived from the base class Shape such as Rectangle, Circle, Line and Polygon. In each derived class there is a function called Draw() which draws the corresponding figure. When the message to draw a specific figure is sent, the program automatically calls the appropriate version of Draw() for the current object type.
2.5 Advantages and Disadvantages of Object Oriented Programming

The characteristics of object oriented programming, encapsulation, abstraction, inheritance and polymorphism, provide many advantages. Encapsulation hides details of the implementation of a class, which protects the program from unwanted modification and makes complex classes easier to use. With inheritance, new applications can easily be created by using preexisting components which have been previously tested and debugged. For example, the Microsoft Foundation Classes (MFC) library included with the Microsoft Visual C++ compiler [Microsoft Corporation, 1993], provides all the necessary classes for developing applications for the Microsoft Windows graphical operating environment. The effort required to develop Windows applications is significantly reduced when these class libraries are used. The concept of polymorphism allows programmers to develop cleaner programs at a higher level of abstraction. In summary, the object oriented programming methodology facilitates the management of complex engineering software systems and has many advantages over traditional procedural programming.

Object oriented programming does have a few disadvantages. One of the chief disadvantages is higher memory requirement. Also, because of the overhead involved, programs developed using object oriented programming may result in longer execution time than procedural programs.

2.6 Object Oriented Programming Languages

The first object oriented programming language was SIMULA which was developed in the mid-1960's. SIMULA is a language for programming computer simulations. Initially object oriented programming was not widely accepted since few object oriented programming languages were available [Yu and Adeli, 1991]. Object oriented programming became popular mainly through the introduction of SmallTalk.
SmallTalk, the oldest fully object oriented language, is an extension of SIMULA and developed in the 1970's [Kim and Lochovsky, 1989]. The explosion of interest in object oriented programming occurred after C++, an object oriented version of the popular C programming language, emerged in the early 1980s. Today C++ has become the most popular and successful object oriented programming language.

The reason for the success of C++ lies in the fact that C++ is a superset of the C programming language and many programmers are proficient in C [Hekmatpour, 1990]. C++ became a widely accepted object oriented programming languages especially among the programmers who wanted to switch from procedural programming to object oriented programming. As an extension of the procedural programming language, C++ combines both object oriented programming and conventional procedural programming. Thus C++ makes it possible to create reusable objects, but uses procedural methods to build these objects. Also, C++ allows reuse of previously written C code and gives the programmer the freedom to use as little or as much of the object oriented programming techniques as needed. Thus, C++ has proved to be an enormously popular and attractive object oriented programming language.

2.7 Applications of Object Oriented Programming in Structural Engineering

In recent years, the object oriented programming approach has received attention in engineering. Several studies on the use of object oriented programming techniques for structural engineering applications have been published. Miller gave an overview of how object oriented programming can provide a useful tool for structural engineering applications. He also presented an object oriented approach to structural analysis and design [Miller, 1991]. An-Nashif and Powell described an object oriented algorithm for automated modeling of frame structures in which the structure is defined in terms of its
components and connections [An-Nashif and Powell, 1991]. Kulkarni used the object-oriented programming approach in developing applications for the analysis and design of reinforced concrete structures [Kulkarni, 1993]. He also compared two object-oriented programming languages, Actor and C++, and developed several applications for the design of reinforced concrete beams. Yu and Adeli discussed how object-oriented programming concepts can be utilized in a computer-aided structural design. They developed a C++ program for the design of welded steel plate girders according to the AISC LRFD Specifications system [Yu and Adeli, 1991]. Patel developed a Windows-based post processor for the design of reinforced concrete space frames using the object-oriented technique [Patel, 1994]. The program is capable of designing beams and columns of reinforced concrete space frames in accordance with the ACI specification. Akhras and Foo described the design and development of two object-oriented systems, one for the inspection of timber trusses and another one for the condition survey of concrete pavements [Akhras and Foo, 1993]. Jaiswal and Riggs developed an object model which can analyze and design two-dimensional steel frame structures subjected to static loads [Jaiswal and Riggs, 1991].

Although a few structural engineering applications have been developed using the object-oriented programming technique, most existing structural design programs still use conventional procedural languages. Therefore, more studies need to be done in using the object-oriented programming approach for the development of structural engineering applications.

In this chapter, an overview of object-oriented programming was presented. The key concepts and major characteristics of object-oriented programming, and the advantages and disadvantages of object-oriented programming, as well as the object
oriented programming languages were briefly discussed. Also, some applications of object oriented programming in structural engineering were provided.
Chapter 3
Design Procedure for Reinforced Concrete T-Beam Bridge Superstructure

3.1 Introduction

In this chapter, an overview of the design of reinforced concrete T-beam bridge superstructure in accordance with the newly-adopted AASHTO LRFD Bridge Design Specifications is presented. Then, the method developed for determining influence line equations for the analysis of the deck and beam is described. Finally, the procedure for the design of the deck and beam as well as the assumptions made during the design process are described. This chapter also gives a brief description of the changes in the new AASHTO Specifications for continuous beam design.

3.2 Overview of Design Procedure

In August 1994, the new AASHTO LRFD Bridge Design Specification was published. The Specification is based on the load and resistance factor design philosophy and represents a new approach towards the design of highway bridge structures and foundations. The new AASHTO Specifications uses both U.S. Customary units and metric units. The Federal Highway Administration has set October 1996 as the target date after which all direct federal and federally aided construction projects will be required to use the metric system [Barker, 1994]. The program presented in the thesis was developed using the new AASHTO LRFD Bridge Design Specifications and uses metric units for the design of reinforced concrete T-beam bridge superstructure, decks and beams.
The reinforced concrete T-beam bridge is a common type of bridge used for small and medium spans. T-beam construction consists of a vertical rectangular stem with a wide top flange. The top flange is usually the transversely reinforced deck slab and the riding surface for traffic. The design of the T-beam bridge superstructure can be divided into two parts: deck design and beam design [Heins and Lawrie, 1984]. The analyses of deck and beam are quite difficult since the deck and the beam are continuous structures. This program performs the analysis and design of a T-beam bridge superstructure consisting of a maximum of 10 beam spans, subjected to a HL-93 live load. There is no limit on the number of decks.

The input to the program consists of data related to the general description of the bridge structure including the name of the project, number of decks and beams, deck and beam spacing, length of the overhanging cantilever, skew angle of the bridge; data related to materials and loads including compressive strength of concrete, yield strength of steel, ductility, redundancy and operational importance factors; data related to the deck including deck structural thickness and slab thickness, thickness of overhanging cantilever, thickness of future wearing surface, top and bottom cover to the reinforcing steel, designation number of bars; and data related to the beam including width and depth of the beam, cover to reinforcement, and stirrup size. Other information entered during the deck or beam design includes bar spacing for deck reinforcement, and the size as well as number of bars.

In order to minimize data entry, many default values are provided. These include compressive strength of concrete and yield strength of steel (Table 3.1), factors relating to ductility, redundancy and operational importance (Table 3.2), top and bottom cover as well as size of bars for deck reinforcement (Table 3.3), clear cover for beam flexural reinforcement and size of stirrups in the beam (Table 3.4).
Table 3.1 Default values for material properties

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concrete compressive strength, ( f_c )</td>
<td>30 MPa</td>
</tr>
<tr>
<td>Yield strength of steel, ( f_y )</td>
<td>400 MPa</td>
</tr>
</tbody>
</table>

Table 3.2 Default values for load modifiers

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factor relating to ductility, ( \eta_D )</td>
<td>0.95</td>
</tr>
<tr>
<td>Factor relating to redundancy, ( \eta_R )</td>
<td>0.95</td>
</tr>
<tr>
<td>Factor relating to operational importance, ( \eta_I )</td>
<td>1.05</td>
</tr>
</tbody>
</table>

Table 3.3 Default values for cover and bar size for deck steel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deck top cover to reinforcing steel</td>
<td>60 mm</td>
</tr>
<tr>
<td>Deck bottom cover to reinforcing steel</td>
<td>25 mm</td>
</tr>
<tr>
<td>Size of reinforcing bars</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 3.4 Default values for cover and stirrup size for beam steel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cover for flexural reinforcement</td>
<td>50 mm</td>
</tr>
<tr>
<td>Stirrup size</td>
<td>15</td>
</tr>
</tbody>
</table>
3.3 Influence Line Equations

When a moving load changes position on a structure, it causes varying effects in the structure. For a given resistance, such as the reaction at a support, the shear or moment at a section of a beam, there is usually one position that will result in the most severe effect. In general, the variation of the response of a structure caused by a moving unit load can be expressed as a function of the position of the load [Au and Christiano, 1987]. The graphical representation of the response is called the influence line.

3.3.1 Muller-Breslau's Principle

For statically determinate structures, the influence lines consist of straight line segments only, so it is easy to get the influence line function for the reaction, shear or bending moment at any point along the structure. However, for statically indeterminate structures such as continuous beams, it is difficult to obtain the influence line function. One of the most effective methods of obtaining influence lines is by the use of Muller-Breslau's principle, which states that the ordinates of the influence line for any action in a structure are equal to those of the deflection curve obtained by releasing the restraint corresponding to this action and introducing a corresponding unit displacement at that point [Ghali and Neville, 1989].

In bridge design, the loads of interest are transient loads such as the design truck load, the design tandem load and the design lane load. Thus, the maximum reaction, shear and positive or negative moment, can be obtained from influence lines for those effects. Based on the analysis of continuous beams and previous experience with bridge design, it is usually assumed that the maximum positive moment occurs at either a distance of 0.4 times the first span from the first left support, or a distance of 0.5 times the second span from the second left support. The maximum negative moment occurs at either the first
support when there is an overhanging cantilever or at the second support. The maximum shear occurs at one of three possible locations: to the right of the first support, to the left of the second support or to the right of the second support (Fig 3.1).

![Diagram](image)

Fig. 3.1 Location of maximum effects for a continuous deck or beam

From Muller-Breslau's principle, the general shape of the influence lines for different effects can be obtained. These are shown in Fig. 3.2.

### 3.3.2 Influence Line Equations

From the above discussion, it is possible to determine the shape of influence lines using Muller-Breslau's principle. However, we also need to know the magnitudes of the ordinates of the influence lines. From Fig. 3.2 it is observed that the shapes of the influence lines are similar to a quadratic or a cubic curve. Thus, it is possible to use second-degree or third-degree interpolating functions to represent the influence line equations.
Fig. 3.2 General shape of influence lines for different effects
The second-degree interpolating function can be written as

\[ f(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} f(x_0) + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} f(x_1) + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} f(x_2) \]

and the third-degree interpolating function is

\[ f(x) = \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} f(x_0) + \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} f(x_1) + \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} f(x_2) + \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)} f(x_3) \]

in which \(x_0, x_1, x_2, x_3\) represent points along the \(x\)-axis and \(f(x_0), f(x_1), f(x_2), f(x_3)\) are the corresponding functional values (which in our case are the ordinates of the influence line).

The functional values \(f(x_i)\) can be obtained from structural analysis. In this study, the RISA-2D structural analysis program was used to obtain the values of \(f(x_0), f(x_1), f(x_2), f(x_3)\). For the second-degree interpolating function, \(x_0, x_1, x_2\) were taken as 0.3, 0.5, 0.7 times the span length respectively. For the third-degree interpolating function, \(x_0, x_1, x_2, x_3\) were taken as 0.1, 0.2, 0.3, 0.4 times the span length respectively. For example, for three-span continuous beams with a span ratio of 1.2, the influence line for the moment at a location of 0.4 times the first span from the first left support can be expressed as:

For \(x <= 0.4\)

\[ M(x) = 0.05028 (x-0.2) (x-0.3) (x-0.4) / (-0.0006) + 0.10114 (x-0.1) (x-0.3) (x-0.4) / (0.0004) + 0.15319 (x-0.1) (x-0.2) (x-0.4) / (-0.0006) + 0.20700 (x-0.1) (x-0.2) (x-0.3) / (0.0024) \]
For $x \leq 1.0$

$$M(x) = 0.26700 \ (x-0.6) \ (x-0.8) \ (x-1.0) / (-0.048)$$
$$+ 0.12229 \ (x-0.4) \ (x-0.8) \ (x-1.0) / (0.016)$$
$$+ 0.05171 \ (x-0.4) \ (x-0.6) \ (x-1.0) / (-0.016)$$

(3-1b)

For $x \leq 2.0$

$$M(x) = -0.03996 \ (x-0.5) \ (x-0.7) \ (x-1.0) / (-0.0168)$$
$$- 0.03857 \ (x-0.3) \ (x-0.7) \ (x-1.0) / (0.01)$$
$$- 0.02484 \ (x-0.3) \ (x-0.5) \ (x-1.0) / (-0.0168)$$

(3-1c)

For $x \leq 3.0$

$$M(x) = 0.00956 \ (x-0.5) \ (x-0.7) \ (x-1.0) / (-0.0168)$$
$$+ 0.01004 \ (x-0.3) \ (x-0.7) \ (x-1.0) / (0.01)$$
$$+ 0.00731 \ (x-0.3) \ (x-0.5) \ (x-1.0) / (-0.0168)$$

(3-1d)

For $x > 3$

$$M(x) = 0$$

(3-1e)

Table C.1 shows the comparison of influence line coordinates for moments in a three span beam obtained using RISA-2D with the corresponding values obtained using interpolating functions. Table C.2 shows the comparison of influence line coordinates for shears in a three span beam obtained using RISA-2D with the corresponding values obtained using interpolating functions. For all the moments and shears near the supports where the maximum shear occurs, there is no difference between the two methods. For shears at a distance of 0.4 times the first span from the first left support and 0.5 times the second span from the second left support, where the maximum positive moments occur, the maximum difference is less than 7%. Thus we can use these interpolating functions for
representing influence line equations to calculate the maximum live load effects. This greatly simplifies the analysis.

<table>
<thead>
<tr>
<th>Table 3.5</th>
<th>Influence line coordinates for continuous beam with different number of spans by RISA-2D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>V 100</td>
</tr>
<tr>
<td></td>
<td>position</td>
</tr>
<tr>
<td>3 span</td>
<td></td>
</tr>
<tr>
<td>4 span</td>
<td></td>
</tr>
<tr>
<td>5 span</td>
<td></td>
</tr>
<tr>
<td>6 span</td>
<td></td>
</tr>
<tr>
<td>7 span</td>
<td></td>
</tr>
<tr>
<td></td>
<td>M 104</td>
</tr>
<tr>
<td></td>
<td>position</td>
</tr>
<tr>
<td>3 span</td>
<td></td>
</tr>
<tr>
<td>4 span</td>
<td></td>
</tr>
<tr>
<td>5 span</td>
<td></td>
</tr>
<tr>
<td>6 span</td>
<td></td>
</tr>
<tr>
<td>7 span</td>
<td></td>
</tr>
<tr>
<td></td>
<td>M 200</td>
</tr>
<tr>
<td></td>
<td>position</td>
</tr>
<tr>
<td>3 span</td>
<td></td>
</tr>
<tr>
<td>4 span</td>
<td></td>
</tr>
<tr>
<td>5 span</td>
<td></td>
</tr>
<tr>
<td>6 span</td>
<td></td>
</tr>
<tr>
<td>7 span</td>
<td></td>
</tr>
<tr>
<td></td>
<td>M 205</td>
</tr>
<tr>
<td></td>
<td>position</td>
</tr>
<tr>
<td>3 span</td>
<td></td>
</tr>
<tr>
<td>4 span</td>
<td></td>
</tr>
<tr>
<td>5 span</td>
<td></td>
</tr>
<tr>
<td>6 span</td>
<td></td>
</tr>
<tr>
<td>7 span</td>
<td></td>
</tr>
</tbody>
</table>
Table 3.5 (continued)

<table>
<thead>
<tr>
<th></th>
<th>V 104</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>position</td>
<td>0.2</td>
<td>0.4</td>
<td>0.7</td>
<td>1.3</td>
<td>1.5</td>
<td>1.7</td>
<td>2.3</td>
<td>2.5</td>
</tr>
<tr>
<td>3 span</td>
<td>-0.2512</td>
<td>-0.4896</td>
<td>0.2048</td>
<td>-0.0770</td>
<td>-0.0750</td>
<td>-0.0490</td>
<td>0.0238</td>
<td>0.0250</td>
<td>0.0182</td>
</tr>
<tr>
<td>4 span</td>
<td>-0.2514</td>
<td>-0.4900</td>
<td>0.2044</td>
<td>-0.0761</td>
<td>-0.0737</td>
<td>-0.0476</td>
<td>0.0206</td>
<td>0.0201</td>
<td>0.0131</td>
</tr>
<tr>
<td>5 span</td>
<td>-0.2515</td>
<td>-0.4900</td>
<td>0.2044</td>
<td>-0.0761</td>
<td>-0.0736</td>
<td>-0.0475</td>
<td>0.0204</td>
<td>0.0197</td>
<td>0.0128</td>
</tr>
<tr>
<td>6 span</td>
<td>-0.2515</td>
<td>-0.4900</td>
<td>0.2043</td>
<td>-0.0761</td>
<td>-0.0736</td>
<td>-0.0475</td>
<td>0.0204</td>
<td>0.0197</td>
<td>0.0127</td>
</tr>
<tr>
<td>7 span</td>
<td>-0.2515</td>
<td>-0.4900</td>
<td>0.2043</td>
<td>-0.0761</td>
<td>-0.0736</td>
<td>-0.0475</td>
<td>0.0204</td>
<td>0.0197</td>
<td>0.0127</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>V 110</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>position</td>
<td>0.3</td>
<td>0.5</td>
<td>0.7</td>
<td>1.3</td>
<td>1.5</td>
<td>1.7</td>
<td>2.3</td>
<td>2.5</td>
</tr>
<tr>
<td>3 span</td>
<td>-0.3728</td>
<td>-0.6000</td>
<td>-0.7952</td>
<td>-0.0770</td>
<td>-0.0750</td>
<td>-0.0490</td>
<td>0.0238</td>
<td>0.0250</td>
<td>0.0182</td>
</tr>
<tr>
<td>4 span</td>
<td>-0.3731</td>
<td>-0.6005</td>
<td>-0.7956</td>
<td>-0.0761</td>
<td>-0.0737</td>
<td>-0.0476</td>
<td>0.0206</td>
<td>0.0201</td>
<td>0.0131</td>
</tr>
<tr>
<td>5 span</td>
<td>-0.3732</td>
<td>-0.6005</td>
<td>-0.7957</td>
<td>-0.0761</td>
<td>-0.0736</td>
<td>-0.0475</td>
<td>0.0204</td>
<td>0.0197</td>
<td>0.0128</td>
</tr>
<tr>
<td>6 span</td>
<td>-0.3732</td>
<td>-0.6005</td>
<td>-0.7957</td>
<td>-0.0761</td>
<td>-0.0736</td>
<td>-0.0475</td>
<td>0.0204</td>
<td>0.0197</td>
<td>0.0127</td>
</tr>
<tr>
<td>7 span</td>
<td>-0.3732</td>
<td>-0.6005</td>
<td>-0.7957</td>
<td>-0.0761</td>
<td>-0.0736</td>
<td>-0.0475</td>
<td>0.0204</td>
<td>0.0197</td>
<td>0.0127</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>V 200</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>position</td>
<td>0.3</td>
<td>0.5</td>
<td>0.7</td>
<td>1.3</td>
<td>1.5</td>
<td>1.7</td>
<td>2.3</td>
<td>2.5</td>
</tr>
<tr>
<td>3 span</td>
<td>0.0910</td>
<td>0.1250</td>
<td>0.1190</td>
<td>0.7280</td>
<td>0.5000</td>
<td>0.2720</td>
<td>-0.1190</td>
<td>-0.1250</td>
<td>-0.0910</td>
</tr>
<tr>
<td>4 span</td>
<td>0.0926</td>
<td>0.1272</td>
<td>0.1211</td>
<td>0.7236</td>
<td>0.4933</td>
<td>0.2651</td>
<td>-0.1031</td>
<td>-0.1005</td>
<td>-0.0656</td>
</tr>
<tr>
<td>5 span</td>
<td>0.0927</td>
<td>0.1274</td>
<td>0.1213</td>
<td>0.7233</td>
<td>0.4928</td>
<td>0.2648</td>
<td>-0.1020</td>
<td>-0.0987</td>
<td>-0.0638</td>
</tr>
<tr>
<td>6 span</td>
<td>0.0928</td>
<td>0.1274</td>
<td>0.1213</td>
<td>0.7233</td>
<td>0.4928</td>
<td>0.2648</td>
<td>-0.1019</td>
<td>-0.0986</td>
<td>-0.0637</td>
</tr>
<tr>
<td>7 span</td>
<td>0.0928</td>
<td>0.1274</td>
<td>0.1213</td>
<td>0.7233</td>
<td>0.4928</td>
<td>0.2648</td>
<td>-0.1019</td>
<td>-0.0986</td>
<td>-0.0637</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>V 205</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>position</td>
<td>0.3</td>
<td>0.5</td>
<td>0.7</td>
<td>1.3</td>
<td>1.5</td>
<td>1.7</td>
<td>2.3</td>
<td>2.5</td>
</tr>
<tr>
<td>3 span</td>
<td>0.0910</td>
<td>0.1250</td>
<td>0.1190</td>
<td>-0.2720</td>
<td>-0.5000</td>
<td>0.2720</td>
<td>-0.1190</td>
<td>-0.1250</td>
<td>-0.0910</td>
</tr>
<tr>
<td>4 span</td>
<td>0.0926</td>
<td>0.1272</td>
<td>0.1211</td>
<td>-0.7638</td>
<td>0.4933</td>
<td>0.2651</td>
<td>-0.1031</td>
<td>-0.1005</td>
<td>-0.0656</td>
</tr>
<tr>
<td>5 span</td>
<td>0.0927</td>
<td>0.1274</td>
<td>0.1213</td>
<td>-0.2767</td>
<td>0.4928</td>
<td>0.2648</td>
<td>-0.1020</td>
<td>-0.0987</td>
<td>-0.0638</td>
</tr>
<tr>
<td>6 span</td>
<td>0.0928</td>
<td>0.1274</td>
<td>0.1213</td>
<td>-0.2767</td>
<td>0.4928</td>
<td>0.2646</td>
<td>-0.1019</td>
<td>-0.0986</td>
<td>-0.0637</td>
</tr>
<tr>
<td>7 span</td>
<td>0.0928</td>
<td>0.1274</td>
<td>0.1213</td>
<td>-0.2767</td>
<td>0.4928</td>
<td>0.2646</td>
<td>-0.1019</td>
<td>-0.0986</td>
<td>-0.0637</td>
</tr>
</tbody>
</table>

Table 3.5 shows influence line coordinates for continuous beams with different number of spans obtained using the RISA-2D program. It is observed that as the number of spans increases, the difference between the influence line coordinates decreases. Thus,
it is assumed that for continuous beams having four or more spans the influence line coordinates are the same as those for a four span beam.

3.4 Deck Design

In a T-beam bridge, the deck directly supports the wheel loads. According to the new AASHTO Specifications, an approximate method of analysis can be used for the design of the deck. In this method the deck is subdivided into strips perpendicular to the supporting components. This approximate strip model for the deck is similar to the method contained in the previous AASHTO Specifications.

3.4.1 Deck Input Data

For the design of the deck, the input data is divided into six groups: (1) material properties such as compressive strength of concrete and yield strength of steel; (2) load modifiers including factors relating to ductility, redundancy and operational importance; (3) parameters describing the deck such as the number of decks, deck spacing, and length of the overhanging cantilever; (4) deck thickness including the deck structural thickness, the actual thickness of the slab, the thickness of overhanging cantilever and the thickness of the future wearing surface; (5) deck cover including cover to the top and bottom reinforcing bars, and bar size; (6) deck design data including deck reinforcing bar spacing for positive moment and negative moment. The first five data groups are entered before the deck is designed and the last data group is entered during the design and can be modified based on information provided by the program. Default values for the data in group 5 are also provided as shown in Table 3.3.
3.4.2 Deck Design Procedure

The deck design is based on the AASHTO approximate strip method [AASHTO, 1994]. The widths of the equivalent strip of the deck are taken as:

For overhang: \[ 1140 + 0.833 \times (\text{overhang length} - 680); \]
For positive moment: \[ 660 + 0.55 \times (\text{deck spacing}); \]
For negative moment: \[ 1220 + 0.25 \times (\text{deck spacing}). \]

The load per unit width of the equivalent strips is obtained by dividing the total load on one design traffic lane by the calculated strip width.

Since the deck spacing is usually equal and the number of deck spans is more than four, the deck force effects were calculated by using the influence line equations for a continuous beam with five equal spans. Fewer equal spans will give slightly different force effects. The maximum positive moment was taken as the larger of \( M_{104} \) and \( M_{205} \) as shown in Fig. 3.3. The maximum negative moment was taken as the larger of \( M_{100} \) and \( M_{200} \) as shown in Fig. 3.4. According to the AASHTO Specifications, when the strip method is used, the maximum positive moment in any deck panel between beams should be taken to apply to all positive moment regions. Similarly, the maximum negative moment in any deck should be taken to apply to all negative moment regions. Thus, all positive moment regions in the deck were designed using the larger of \( M_{104} \) and \( M_{205} \), and all negative moment regions in the deck were designed using the larger of \( M_{100} \) and \( M_{200} \).
The required reinforcement for the deck is computed for the Strength I Limit State which is the basic load combination relating to normal vehicular use of the bridge without wind. The factored load effect is given by

\[ \eta \sum \gamma_i q_i = \eta [1.25DC + 1.50DW + 1.75(LL + IM)] \]  \hspace{1cm} (3-2)

where:

- \( \eta = \eta_D \eta_R \eta_L \)
- \( DC = \) dead load effect of structural components and non-structural attachments
- \( DW = \) dead load effect of wearing surface and utilities
- \( LL = \) vehicular live load effect
- \( IM = \) vehicular dynamic load allowance
- \( q_i = \) load effects
- \( \gamma_i = \) load factors
Based on the required area of steel computed by the program, the bar sizes and spacing for maximum positive and negative moments can be entered. The adequacy of the reinforcement for crack control is checked for Service I Limit State which is the load combination relating to normal operational use of the bridge with a 90 km/hr wind, and with all loads taken at their normal values.

Cracking is controlled by limiting the tensile stress under service loads according to the following equation

\[ f_s \leq f_{sa} = \frac{Z}{(d_c A)^{1/3}} \leq 0.6 f_y \]  \hspace{1cm} (3-3)

in which  
- \( Z \) = crack width parameter, 23000 N/mm for severe exposure conditions,
- \( d_c \) = depth of concrete from extreme tension fiber to center of closest bar, but not greater than 50 mm,
- \( A \) = effective concrete tensile area per bar having the same centroid as the reinforcement.

From the AASHTO Specifications, the deck shear resistance is considered to be adequate when the approximate strip method for moment design is used. Also, the deck fatigue limit state need not be investigated for T-beam superstructure applications.

3.4.3 Deck Design Results

The deck design results can be displayed on the screen or printed on the printer. The deck results consist of (1) the number of decks, deck spacing and length of the overhanging cantilever; (2) deck structural thickness, actual slab thickness, thickness of overhanging cantilever and thickness of future wearing surface; (3) compressive strength
of concrete and yield strength of steel; (4) the maximum reaction, maximum positive and negative moment in the deck, as well as load modifiers used in deck design; (5) bar sizes, spacing and area of reinforcing bars for both maximum positive and negative moments, and the positive and negative moment capacity; (6) results of the ductility, cracking and flexural capacity checks. The areas of reinforcing bars for distribution, shrinkage and temperature are also shown.

3.5 Beam Design

The new AASHTO Specifications is based on the load and resistance factor design (LRFD) philosophy, which can be expressed as:

\[ \eta \sum \gamma_i q_i \leq \varphi R_n \]  

(3-4)

where \( \eta \) = load modifier for ductility, redundancy and operational importance
\( \gamma_i \) = statistically based load factor for load type i
\( q_i \) = load effect
\( \varphi \) = statistically based resistance factor
\( R_n \) = nominal resistance

There are two basic changes in beam design when using the new AASHTO Specifications:

(1) Live load model

Three design loads have to be considered: the design truck, design tandem and design lane. The live load combinations to be considered are the design truck plus the
design lane load, and the design tandem plus the design lane load. In addition, truck
fatigue, truck deflection and impact of 33% have to be considered.

(2) Shear design

The nominal shear resistance is determined using the modified compression field
theory, which can be expressed as

\[ V_n = V_c + V_s \]  \hspace{1cm} (3-5) \]

where \( V_n \) is the nominal shear resistance, \( V_c \) is the shear resistance provided by the
concrete, \( V_s \) is the shear resistance provided by the reinforcing steel. \( V_c \) and \( V_s \) are
obtained from

\[ V_c = 0.083 \beta \sqrt{f'c} \ b_v \ d_v \]  \hspace{1cm} (3-5a) \]

\[ V_s = \frac{A_v \ f_y \ d_v (\cot \theta + \cot \alpha) \sin \alpha}{s} \]  \hspace{1cm} (3-5b) \]

where \( \beta = \) factor indicating ability of diagonally cracked concrete to transmit tension

\( \theta = \) angle of inclination of diagonal compressive stresses

\( \alpha = \) angle of inclination of transverse reinforcement to longitudinal axis

\( b_v = \) effective web width taken as the minimum web width within the depth \( d_v \)

\( d_v = \) effective shear depth

\( s = \) spacing of stirrups

\( A_v = \) area of shear reinforcement within a distance \( s \)
3.5.1 Beam Input Data

For beam design, the input data are: (1) material properties including compressive strength of concrete and yield strength of steel; (2) load modification factors including factors relating to ductility, redundancy and operational importance; (3) number of beams, beam spacing and skew angle of the bridge; (4) beam sizes including width and depth of the beam, and clear cover; (5) stirrup size, number of reinforcing bars for both maximum positive and negative moments. Default values are also provided as shown in Table 3.4. The number of bars and bar sizes for both positive and negative moments are also entered during the design of the beam. These can be adjusted during program execution based on the results of the various checks made by the program.

3.5.2 Beam Design Procedure

3.5.2.1 Assumptions for Beam Design

It is assumed that all of the continuous beams are symmetrical about their center line and interior spans are equal to or greater than the corresponding exterior spans. The ratio of interior to exterior span is designated as $N$ and is obtained from:

$$N = \frac{\text{beamSpan}[1]}{\text{beamSpan}[0]}$$  \hspace{1cm} (3-6)

In which:

- $N = 1.0$ \hspace{0.5cm} when $N$ is between 1.0 and 1.1;
- $N = 1.2$ \hspace{0.5cm} when $N$ is between 1.1 and 1.3;
- $N = 1.4$ \hspace{0.5cm} when $N$ is between 1.3 and 1.5;

For symmetric continuous beams subjected to moving vehicular loads, if the ratio $N$ is small, the maximum moment and shear usually occur at the first or second span.
Thus, only the force effects for the first two spans were considered. It is assumed that the maximum positive moment is at either 0.4 times the first span from the first left support or 0.5 times the second span from the second left support. It is assumed that the maximum negative moment is at the second support, the maximum shear is at one of three possible locations: to the right of the first support, to the left of the second support, or to the right of the second support (Fig 3.1). These assumptions do not give significant errors but greatly simplify the analysis method for continuous beams subjected to moving loads. Also, as mentioned earlier, continuous beams having four or more spans are assumed to have the same influence line equations as those for a four span continuous beam.

3.5.2.2 Distribution Factor Method

According to the new AASHTO Specifications, if the number of deck spans is greater than or equal to four, the beams are parallel, and have approximately the same stiffness, the distribution factor method for moment and shear can be used in the analysis and design of the beams. For reinforced concrete T-beams, the distribution factor method for moment and shear can be used if the following conditions are also true:

\[
1100 \text{ mm} \leq \text{deck spacing} \leq 4900 \text{ mm}; \\
110 \text{ mm} \leq \text{deck thickness} \leq 300 \text{ mm}; \\
6000 \text{ mm} \leq \text{beam spacing} \leq 73000 \text{ mm}.
\]

The distribution factor method was used in the program to analyze and design the beams.
The distribution factors for moment are:

For one design lane loaded:

\[ m_{g_M} = 0.06 + \left( \frac{\text{deck spacing}}{4300} \right)^{0.4} \left( \frac{\text{deck spacing}}{\text{beam spacing}} \right)^{0.3} \left( \frac{K_g}{L^3 t} \right)^{0.1} \]  \hspace{1cm} (3-7a)

For two or more design lanes loaded:

\[ m_{g_M} = 0.075 + \left( \frac{\text{deck spacing}}{2900} \right)^{0.6} \left( \frac{\text{deck spacing}}{\text{beam spacing}} \right)^{0.2} \left( \frac{K_g}{L^3 t} \right)^{0.1} \]  \hspace{1cm} (3-7b)

in which \( m_{g_M} \) = the distribution factor for moment

\( K_g \) = the longitudinal stiffness parameter

\( L \) = the beam spacing

\( t \) = the deck thickness

For preliminary design, \( K_g / (L^3 t) \) can be taken as 1.0.

The distribution factors for shear are:

For one design lane loaded:

\[ m_{g_V} = 0.36 + \frac{\text{deck spacing}}{7600} \]  \hspace{1cm} (3-8a)

For two or more design lanes loaded:

\[ m_{g_V} = 0.2 + \frac{\text{deck spacing}}{3600} - \left( \frac{\text{deck spacing}}{10700} \right)^{2.0} \]  \hspace{1cm} (3-8b)

in which \( m_{g_V} \) = the distribution factor for shear.
3.5.2.3 Calculation of Live Load Effects

The distributed live load moments are obtained from:

\[ M_{LL+IM} = mg \ r \ ((M_{Tr} \ or \ M_{Ta})(1+IM) + M_{Ln}) \]  \hspace{1cm} (3-9a)

in which

- \( M_{Tr} \) = moment due to design truck load
- \( M_{Ta} \) = moment due to design tandem load
- \( M_{Ln} \) = moment due to design lane load
- \( IM \) = dynamic load allowance (\( IM = 0.33 \))
- \( M_{LL+IM} \) = moment due to live loads
- \( mg \) = the distribution factor for moment
- \( r \) = reduction of load distribution factors for moment in longitudinal on skewed supports

The distributed live load shears are obtained from:

\[ V_{LL+IM} = mg \ r \ ((V_{Tr} \ or \ V_{Ta})(1+0.33) + V_{Ln}) \]  \hspace{1cm} (3-9b)

in which

- \( V_{Tr} \) = shear due to design truck load
- \( V_{Ta} \) = shear due to design tandem load
- \( V_{Ln} \) = shear due to design lane load
- \( IM \) = dynamic load allowance (\( IM = 0.33 \))
- \( V_{LL+IM} \) = shear due to live loads
- \( mg \) = the distribution factor for shear
- \( r \) = correction factor for load distribution factors for shear of the obtuse corner
Usually T-beam bridges are used for small or medium spans. For relatively short spans, the design tandem load controls positive moments and the truck load governs negative moments. For example, for a three-span continuous beam with a span ratio of 1.2, the live loads for moment calculation will be placed as shown in Fig. 3.5. The live loads for shear calculation will be placed as shown in Fig. 3.6.

![Diagram of live load placement](image)

Fig 3.5 Live load placement for moment calculation
for M205

\[ 110\text{kN} \ 110\text{kN} \]

\[ 0.5L \ 1200\text{mm} \]

\[ 9.3\text{N/mm} \]

Fig 3.5 Continued

for V100

\[ 145\text{kN} \ 145\text{kN} \ 35\text{kN} \]

\[ 4300\text{mm} \ 4300\text{mm} \]

\[ 110\text{kN} \ 110\text{kN} \]

\[ 1200\text{mm} \]

\[ 9.3\text{N/mm} \] \[ 9.3\text{N/mm} \]

Fig 3.6 Live load placement for shear calculation
for V110

\[\begin{align*}
35\text{kN} & \quad 145\text{kN} & \quad 145\text{kN} \\
4300\text{mm} & \quad 4300\text{mm} \\
4300\text{mm} & \quad 4300\text{mm} \\
1200\text{mm} & \quad 1200\text{mm} \\
\end{align*}\]

Fig 3.6 Continued

for V200

\[\begin{align*}
145\text{kN} & \quad 145\text{kN} & \quad 35\text{kN} \\
4300\text{mm} & \quad 4300\text{mm} \\
1200\text{mm} & \quad 1200\text{mm} \\
\end{align*}\]
3.5.2.4 Crack Control

For crack control, the flexural reinforcement provided in the beam is checked using Service I Limit State.

\[
M_s = M_{DC} + M_{DW} + M_{LL+IM} \quad (3-10a)
\]
\[
As = \frac{Ms}{fs \cdot j \cdot d} \quad (3-10b)
\]
\[
f_s = \frac{n \cdot Ms \cdot (d-x)}{I_{cr}} \leq f_{sa} \quad (3-10c)
\]
\[
f_{sa} = \frac{Z}{(d_e \cdot A)^{1/3}} \leq 0.6 f_y \quad (3-10d)
\]

in which
- \( M_s \) = moment due to combined loads at service state
- \( M_{DC} \) = moment due to DC dead load
- \( M_{DW} \) = moment due to DW dead load
- \( M_{LL+IM} \) = moment due to live load and dynamic load
- \( f_{sa} \) = tensile stress under service limit state
- \( I_{cr} \) = moment of inertia of elastic-cracked transformed section about the centroidal axis of the beam
- \( x \) = depth from extreme compression fiber to centroidal axis of the beam
- \( d \) = effective depth from extreme compression fiber to centroid of tensile force in tensile reinforcement
- \( n \) = modular ratio
- \( j \equiv 0.875 \)

After reinforcement has been selected, the program performs a check to ensure that the provided reinforcing bars do satisfy the crack control criterion at the maximum positive and negative moment sections.
3.5.2.5 Fatigue Investigation

For a continuous T-beam without prestress, there will be regions, at the bottom of the beam or at the top of the beam, where permanent loads do not produce compressive stress. Fatigue must be considered in these regions.

When consideration of fatigue is necessary, the stress range \((f_{\text{max}} - f_{\text{min}})\) is determined using the Fatigue Limit State which considers one design truck with constant spacing of 9000 mm between 145 kN axles, a dynamic load allowance \(IM\) of 15%, and distribution factor for one traffic lane:

\[
M_f = 0.75 \, g \, r \, M_{LL} (1 + IM)
\]  

(3-11)

in which \(g = \) distribution factor
\(r = \) reduction of load distribution factors for moment in longitudinal on skewed supports
\(M_{LL} = \) moment due to live loads
\(IM = \) dynamic load allowance (IM = 0.15)

The stress range \((f_{\text{max}} - f_{\text{min}})\) should not exceed:

\[
f_f = 145 - 0.33 \, f_{\text{min}} + 55 \, (r/h)
\]  

(3-12)

where \(f_{\text{min}} = \) algebraic minimum stress level resulting from the fatigue load combination,
\(f_{\text{max}} = \) algebraic maximum stress level resulting from the fatigue load combination,
\(r/h = \) ratio of base radius to height of rolled-on transverse deformations,
\(r/h\) may be taken as 0.3
The section properties for fatigue investigations are based on the cracked section.

The program checks fatigue at three positions: middle of the first and the second span where the maximum positive moment usually occurs, and the second support where the maximum negative moment usually occurs.

3.5.2.6 Beam Deflection

Bridges should be designed to avoid undesirable structural or psychological effects due to deformations. The deflection of the beam consists of the live load deflection and the dead load deflection.

(1) Live load deflection

It is usually assumed that the deflection produced by live load is maximum where the moment is maximum. The live load deflection is given by

\[ \Delta_{\text{live}} = \alpha \frac{M_a L^2}{E_c I_e} \]  \hspace{1cm} (3-13)

where \( M_a = M_{DC} + M_{DW} + mg M_{LL} (1 + IM) \) \hspace{1cm} (3-13a)
and \( I_e = \left(\frac{M_{\sigma}}{M_a}\right)^3 I_g + \left(1 - \left(\frac{M_{\sigma}}{M_a}\right)^3\right) I_{\sigma e} \leq I_g \) \hspace{1cm} (3-13b)

in which

\( M_a = \) maximum moment at the stage for which deflection is computed
\( M_{DC} = \) moment due to DC dead load
\( M_{DW} = \) moment due to DW dead load
\( M_{LL+IM} = \) moment due to live load
\( mg = \) distribution factor for deflection which is equal to the number of lanes divided by the number of beams
\( IM = \) dynamic load allowance (IM = 0.33)
Mcr = cracking moment
Ie = effective moment of inertia
Ig = moment of inertia of the gross concrete section about the centroidal axis neglecting the reinforcement
Ec = modulus of elasticity of concrete

(2) Dead load deflection

The instantaneous dead load deflection is multiplied by a creep factor \( \lambda \) to get a longtime dead load deflection.

\[
\Delta_{\text{dead}} = (1 + \lambda) \Delta_i \tag{3-14}
\]

\[
\lambda = 3.0 - 1.2 \left( \frac{A_{s'}}{A_s} \right) \geq 1.6 \tag{3-14a}
\]

\[
\Delta_i = \alpha \frac{M_{\text{dead}}}{E_c I_e} \tag{3-14b}
\]

in which \( \Delta i \) = instantaneous dead load deflection
\( \lambda \) = creep factor
\( A_s \) = area of tension reinforcement
\( A_{s'} \) = area of compression reinforcement
\( \alpha \) = coefficient reflecting the structure configuration

3.5.2.7 Flexural Capacity

The bar sizes and number of bars selected for crack control must be checked for bending using the Strength I Limit State. The flexural capacity of the beam is:
\[ Mu \leq \varphi \, Mn \]  
\[ (3-15) \]

Where

\[ Mu = 0.95 (1.25M_{DC} + 1.50M_{DW} + 1.75M_{LL+IM}) \]  
\[ (3-15a) \]

\[ \varphi \, Mn = \varphi \, A_s \, f_y \, (d-a/2) \]  
\[ (3-15b) \]

\[ a = A_s \, f_y / 0.85 \, f_c \, b \]  
\[ (3-15c) \]

In which

- \( Mu \) = factored moment
- \( Mn \) = nominal flexural resistance
- \( \varphi \) = resistance factor
- \( M_{DC} \) = moment due to DC dead load
- \( M_{DW} \) = moment due to DW dead load
- \( M_{LL+IM} \) = moment due to live load
- \( A_s \) = area of tension reinforcement
- \( d \) = depth of beam
- \( a \) = depth of equivalent rectangular stress block
- \( b \) = width of the compression face of the beam

Also the limits for reinforcement should be checked. These include the maximum reinforcement check which is

\[ \frac{c}{de} = \frac{a}{\beta_c \, de} \leq 0.42 \]  
\[ (3-16a) \]

And the minimum reinforcement check which is

\[ \rho = \frac{A_s}{Ag} > \rho_{min} = 0.03 \, \frac{f'c}{f_y} \]  
\[ (3-16b) \]
3.5.2.8 Shear Capacity

For non-prestressed concrete beams the shear resistance can be separated into two components $V_c$, the shear capacity of the concrete, and $V_s$, the shear capacity of the transverse reinforcement. Thus, the nominal shear resistance $V_n$ can be determined as the lesser of:

$$V_n = V_c + V_s$$

$$V_n = 0.25 f'_c b v d v$$

in which

$$V_c = 0.083 \beta \sqrt{f'_c} b v d v$$

$$V_s = \frac{A_v f_v d_v (\cot \theta + \cot \alpha) \sin \alpha}{s}$$

where $b_v =$ effective web width taken as the minimum web width within the depth $d v$

d$v =$ effective shear depth taken as the distance between the resultant of the tensile force and the compressive force due to flexure

$s =$ spacing of stirrups

$A_v =$ area of shear reinforcement within a distance $s$

$\beta =$ factor indicating ability of diagonally cracked concrete to transmit tension

$\theta =$ angle of inclination of diagonal compressive stresses

$\alpha =$ angle of inclination of transverse reinforcement

The values of $\beta$ and $\theta$ can be determined by using the tables and figures given in the new AASHTO Specifications. For non-prestressed concrete sections not subjected to axial tension and containing at least the minimum amount of transverse reinforcement, the
new AASHTO Specifications gives a simplified procedure for obtaining the values of $\beta$ and $\theta$. For this case

$$\beta = 2.0 \quad \text{and} \quad \theta = 45^\circ$$

In this study, the simplified method was used to determine the values of $\beta$ and $\theta$.

Shear also causes tension in the longitudinal reinforcement. For a given shear, this tension increases as $V_c$ increases. This extra tension must be added to that caused by flexure:

$$A_s f_y \geq \frac{M_u}{\varphi_f d_v} + (\frac{V_u}{\varphi_v} - 0.5V_e) \cot \theta$$  \hspace{1cm} (3-18)

If this equation is not satisfied, either the tensile reinforcement $A_s$ must be increased or stirrups must be placed closer together to increase $V_s$. Thus

$$V_s = \max\left\{\frac{V_u}{\varphi_v} - V_c, \frac{2(\frac{V_u}{\varphi_v} - (A_s f_y - \frac{M_u}{\varphi_f d_v}) \tan \theta)}{A_s f_y d_v \cot \theta}\right\}$$  \hspace{1cm} (3-19a)

$$s = \frac{A_s f_y d_v \cot \theta}{V_s}$$  \hspace{1cm} (3-19b)

3.5.3 Beam Design Results

Beam design results include (1) number of beams, beam spacing and skew angle of the bridge; (2) effective flange width, depth of flange, width of web, depth of beam and clear cover; (3) compressive strength of concrete, yield strength of steel and load

Design Procedure for Reinforced Concrete T-Beam Bridge Superstructure
modifiers used in beam design; (4) maximum positive and negative moment, shear at left and right end of the first and second span; (5) bar sizes and number of bars for both maximum positive and negative moments, nominal flexural resistance for positive and negative moment; (6) size and spacing of stirrup, and nominal shear resistance; (7) maximum live load and dead load deflection at first and second span; (8) results of ductility, minimum reinforcement, cracking, fatigue and flexural capacity checks under maximum positive and negative moments for first and second span.
Chapter 4
Program Architecture

4.1 Introduction

The object oriented programming technique was used in developing the program. This chapter provides a description of two types of classes developed in the program: structural classes and interface classes. Structural classes are used for representing structural elements such as the bridge superstructure, the deck and the beam. Interface classes are used for representing user interface elements such as windows for displaying the structure and program results, dialog boxes for obtaining input, and menus. All classes are derived from classes in the Microsoft Foundation Class (MFC) library. This chapter also serves as a user manual for the program.

4.2 Structural Classes

The structural classes developed are: Structure for representing the bridge superstructure, Inline for representing influence line equations used to compute different load effects, Deck for representing the bridge deck, and Beam for representing the bridge beam. In order to use the application framework supplied by the Microsoft Foundation Class Library, all structural classes are derived from CObject, which is the base class for most of the classes in the Microsoft Foundation Class library. Also, in order to make the structural classes serializable, the DECLARE_SERIAL macro was used in the class declaration and the IMPLEMENT_SERIAL macro was used in the class implementation for each of the classes. A more complete definition of each class can be found in Appendix A, which lists the header file for each class in the program.
4.2.1 Class Structure

The class Structure represents the bridge superstructure. All input data is stored in the Structure class. In order that structural data can be accessed by derived classes, all data members in Structure class are declared as protected. The important data members in Structure include: project name, description of the bridge, material properties, load modifiers, information regarding reinforcing bars in the deck and beam for flexural and shear design.

There are two constructors in Structure. One is the default constructor necessary to make Structure serializable. The second constructor is used to transfer data entered through dialog boxes. The member functions in Structure are declared as public so that the structural classes can use these functions to gain access to data. Some of these functions are:

calLane()              Calculates number of design lanes
                      number of design lanes = (int) \( \frac{width\ of\ the\ bridge}{3600} \)

calLoadModi()              Calculates the load modifier \( \eta \)
                      \( \eta = \eta_D \times \eta_R \times \eta_I \)

calMultFactor()              Calculates the multiple presence factor \( m \)
                      \[
                      m = \begin{cases} 
                      1.2 & \text{when design lane} = 1 \\
                      1.0 & \text{when design lane} = 2 \\
                      0.8 & \text{when design lane} = 3 \\
                      0.65 & \text{when design lane} > 3 
                      \end{cases}
                      \]

calEc()              Calculates Young's modulus of concrete using
                      \[
                      E_c = 0.043 \times (2400)^{1.5} \times \sqrt{f_c'}\]

Program Architecture
calBeta1() Calculates ratio of depth of equivalent uniformly stressed compression zone to depth of actual compression zone

$$\beta_i = 0.85 - 0.05 \times \frac{f'_{ce} - 28}{f'}$$

caln() Calculates \( n = \frac{Es}{Ec} \)

calDiameter() Calculates nominal diameter of reinforcing bar for a given bar size

calRatio() Calculates beam span ratio

The class Structure also contains several functions that receive the data. These are:

getMaterialData() Obtains material, concrete and steel properties

getLoadData() Obtains load factors relating to ductility, redundancy and operational importance, \( \eta_d, \eta_r, \) and \( \eta_i \) respectively

getDeckData() Obtains data relating to deck description

getBeamData() Obtains data relating to beam description

getDeckInput() Obtains bar sizes and bar numbers during deck design

getBeamInput() Obtains bar sizes and bar numbers during beam design

Class Structure also has a member function called Show which displays the general description of the bridge superstructure so that the accuracy of the input can be verified. An instance of the Structure class called myStructure is created in the constructor of the document class CBridgeDoc.
4.2.2 Class Infl line

The class Infl line is used to represent the influence line equations used for the analysis of load effects. The class Infl line has one private data member \( p_f \) which is the ratio of the position of the concentrated load to the span. All member functions in class Infl line are interpolating polynomials which represent the different force effects. These functions are declared as public so that objects of the Deck and Beam class can use these functions. An instance of the Infl line class called myInfl line is created in the class Deck and Beam.

There are six groups of influence line functions: (1) three-span continuous beam with span ratio between 0.9 - 1.1; (2) three-span continuous beam with span ratio between 1.1 - 1.3; (3) three-span continuous beam with span ratio between 1.3 - 1.5; (4) continuous beam having four or more spans with span ratios between 0.9 - 1.1; (5) continuous beam having four or more spans with span ratio between 1.1 - 1.3; (6) continuous beam having four or more spans with span ratio between 1.3 - 1.5.

In each group, there are eight influence line functions: (i) influence line function for maximum positive moment in first span \( M_{104} \); (2) influence line function for maximum positive moment in second span \( M_{205} \); (3) influence line function for maximum negative moment \( M_{200} \); (4) influence line function for shear at left end of first span \( V_{100} \); (5) influence line function for shear at center of first span \( V_{104} \); (6) influence line function for shear at right end of first span \( V_{110} \); (7) influence line equation for shear at left end of the second span \( V_{200} \); (8) influence line equation for shear at middle of the second span \( V_{205} \).

For example, for a three-span continuous beam with a span ratio of 1.2, the eight influence line functions are:

\[
\text{span32M104()} \quad \text{maximum positive moment in first span} \\
\text{span32M200()} \quad \text{maximum negative moment}
\]
span32M205()  maximum positive moment in second span
span32V100()  shear at left end of the first span
span32V104()  shear at middle of the first span
span32V110()  shear at right end of the first span
span32V200()  shear at left end of the second span
span32V205()  shear at middle of the second span

4.2.3 Class Deck

The class Deck is derived from the base class Structure. It contains all data members and member functions necessary for representing the deck slab. The object myDeck created from the class Deck represents the actual deck to be designed. The class Deck has functions for performing analysis and design, and for displaying the output of the design. Most data used in this class are declared as private. Other classes cannot accessed these data which protects the deck design process from being affected by other functions in the program.

The data obtained from Structure for deck design are the deck material properties such as $f'_c$, $f'_t$, $E_c$, $E_s$, $n$ and $\beta$, load modifier $\eta$, description of the deck, and deck reinforcement. The deck output values include maximum reaction, maximum positive and negative moments, required reinforcement and provided reinforcement for both positive and negative moment regions in the deck, the provided flexural capacity, results of ductility checks, distribution, shrinkage and temperature reinforcement.

All functions defined in class Deck are declared as public. The major functions in the Deck class include:
calDC() Calculates force effects due to structural and nonstructural load components

calDW() Calculates force effects due to wearing surface and utilities load components

calLive() Calculates force effects due to live load components

calEquWidth() Calculates width of equivalent interior strips using the approximate strip method

calEffect() Calculates maximum reaction, maximum positive and negative moments in the deck

calDeckAs() Calculates required area of steel for positive and negative regions

checkPosMom() Checks flexural capacity of the deck at the max +M position

checkNegMom() Checks flexural capacity of the deck at the max -M position

crackPos() Checks crack control at the max +M position

crackNeg() Checks cracking control for the deck at the max -M position

calDistAs() Calculates distribution steel area

calTempAs() Calculates shrinkage and temperature steel area

Show() Displays deck design results

In addition to performing deck analysis and design, the class Deck also allows input deck parameters to be modified during design. For example, if from the deck design results, it is found that the deck flexural capacity checking is inadequate during deck design, the deck depth can be increased or more reinforcement can be provided.
4.2.4 Class Beam

The class Beam contains all necessary data members and member functions for beam design. This class can handle analysis and design of the beam as well as display of design results. The class Beam is derived from the Structure class and thus inherits all of the member variables of the Structure class. Like the Deck class, most data stored in the Beam class is declared as private which cannot not be accessed from outside the Beam class. The object myBeam created from the class Beam represents the actual beam to be designed.

All functions defined in the Beam class are declared as public. The important member functions are:

- calDistIM() Calculates distribution factor for moment in interior beams
- calDistEM() Calculates distribution factor for moment in exterior beam
- calSkewMom() Calculates skew reduction factor for moment
- calDistShear() Calculates distribution factor and skew reduction factor for shear in the beam
- calLiveM() Calculates distributed live load moment
- calLiveV() Calculates distributed live load shear
- calDCIn() Calculates DC dead load Mom and shear in interior beams
- calDWin() Calculates DW dead load Mom and shear in interior beam
- calDCEx() Calculates DC dead load Mom and shear in exterior beams
- calDWEex() Calculates DW dead load Mom and shear in exterior beam
- calcEffFlange() Calculates effective flange width
- serviceAs() Calculates required area of reinforcing bar for positive and negative moments at service limit state
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>crackCheck()</td>
<td>Checks crack control for both positive and negative moments at service limit state</td>
</tr>
<tr>
<td>getServiceAs()</td>
<td>Determines required area of reinforcing bar for positive and negative moments at service limit state</td>
</tr>
<tr>
<td>fatigueM104()</td>
<td>Checks fatigue at maximum positive moment and maximum negative moment positions</td>
</tr>
<tr>
<td>deflectM104()</td>
<td>Calculates maximum live load deflection and maximum dead load deflection as well as beam flexural rigidity in first and second span</td>
</tr>
<tr>
<td>flexureM104()</td>
<td>Checks ductility, minimum reinforcement limit and flexural capacity as well as calculates flexural capacity for maximum positive moment in the beam</td>
</tr>
<tr>
<td>flexureM200()</td>
<td>Checks ductility, minimum reinforcement limit and flexural capacity as well as calculates provided flexural capacity for maximum negative moment in the beam</td>
</tr>
<tr>
<td>shearV104()</td>
<td>Calculates shear strength at the maximum M position</td>
</tr>
<tr>
<td>shearV200()</td>
<td>Calculates shear strength at the maximum V position</td>
</tr>
<tr>
<td>Show()</td>
<td>Displays beam design results</td>
</tr>
</tbody>
</table>

The class Beam also has the capability to allow changes to be made to design parameters during the design and for redesign the beam.

### 4.3 Interface Classes

Interface classes are used to define user interface elements such as windows and dialog boxes used in the Windows environment. In this program, three powerful visual
tools provided with the Microsoft Visual C++ compiler were used to generate the basic
source code for all the interface classes. AppWizard was used to create a skeleton starter
application; App Studio was used to construct the user interface and edit program
resources; Class Wizard was used to process messages and handle dialog boxes.

The main interface classes used in the program are shown as follows. A complete
definition of all these classes are given in Appendix A.

CBridgeDoc

This is the document class derived from the Microsoft Foundation Class (MFC)
library CDocument class. Its primary purpose is to store program data, and save and
load program data from disk. Three objects, myStructure from class Structure,
myDeck from class Deck and myBeam from class Beam, are created in the class
CBridgeDoc. The CBridgeDoc also contains the functions for processing input data.

CBridgeView

This is the view class derived from the MFC class CScrollView. It is
responsible for displaying program data (on the screen, printer or other device). In the
class CBridgeView there is a function called OnDraw() used to display the bridge
design results.

CMainFrame

This is the main frame window class derived from the MFC CFrameWnd class.
The main frame window class manages the main program window, which serves to display
the title bar, the menu bar, borders and other user interface elements.
**CBridge**

This is the application class derived from the MFC application class `CWinApp`. It performs general tasks that do not fall within the province of any of the other classes, such as program initialization and the final cleanup.

Other major interface classes are the classes for dialog boxes, all of which are derived from the MFC dialog class `CDialog`. These include:

- `CTitleDlg`  Project Name dialog box class for entering the project name
- `CMaterialDlg` Material Properties dialog box class for entering material properties
- `CLoadDlg`   Load Modifier dialog box class for entering the three load modifiers, $\eta_D$, $\eta_R$ and $\eta_I$
- `CDeck1Dlg`  Deck Description dialog box class for entering deck data such as number of decks, deck spacing, and length of overhanging cantilever
- `CDeck2Dlg`  Deck Thickness dialog box class for entering deck structural thickness, actual slab thickness, thickness of overhanging cantilever and thickness of future wearing surface
- `CDeck3Dlg`  Deck Cover and Bar Diameter dialog box class for entering the top and bottom cover, and bar size.
- `CBeam1Dlg`  Beam Description dialog box class for entering beam data such as number of beams, beam spacing, and skew angle of the bridge.
- `CBeam2Dlg`  Beam Size dialog box class for entering depth and web width of the beam

*Program Architecture*
CBeam3Dlg  Beam Cover and Stirrup Size dialog box class for entering the 
        cover and designation number of stirrups in the beam
CDeckdDlg  Deck Design dialog box class for entering required reinforcement 
        during deck design
CBeamdDlg  Beam Design dialog box class for entering required reinforcement 
        during beam design
CResuLDlg  Bridge Results dialog box class for selecting the output of design.
CAboutDlg  About dialog box class for displaying information about the 
        program

4.4 Program Operation

This reinforced concrete T-beam bridge design program should be run under the 
Windows environment. It is recommended that the computer have 4MB or more of 
RAM. The program utilizes the standard Windows user interface elements such as menus, 
dialog boxes and child windows for navigation between different parts of the program 
including data input, result checking, modification of design parameters and viewing 
design results. To begin execution of the program, simply double click on the program 
icon from the Windows Program Manager.

4.4.1 Main Menu

There is only one main menu in the program which lists all of the choices for 
program operation. When the application starts, the main menu is displayed in the main 
window. The main menu consists of nine pull down menus: File, Title, Material, Loads, 
Deck, Beam, Design, Results and Help. Fig. 4.1 shows the main menu screen.
4.4.2 File Menu

The File menu has nine items: New, Open, Close, Save, Save As, Print Preview, Print, Print Setup and Exit. The Save and Save As menu items are used to save input data. If a file extension is not specified when saving a file, the program will add the extension .bri to the file name. The Open menu item activates a dialog box which allows the specific file to be opened. If a file extension is not specified when opening a file, the Open dialog box displays a list of all files with the .bri extension. The program also automatically maintains a list of the last four recently opened files on the File menu. These can be opened by simply selecting the name from the File menu. The Print Preview menu item activates the print preview feature that displays the printed appearance of a document. This makes it possible to view the output before sending it to the printer. The Print menu item is used to print program results. The Exit menu item terminates program execution.
4.4.3 Title Menu

The Title menu has one item, Project Name. When selected, this menu item displays the dialog box shown in Fig. 4.2. The user can enter the name of the bridge design project in the edit box.

![Project Name Dialog Box](image)

Fig. 4.2 Project Name Dialog Box

4.4.4 Material Menu

The Material menu has one pull down item, Material Strength. When this menu item is selected, the dialog box shown in Fig. 4.3 is activated. The compressive strength of concrete $f_c$ and yield strength of steel $f_y$ can be entered in the edit fields. It is also possible to simply choose the default material strength values provided in the dialog box. The material properties can be modified during the design if it is found that capacity is not adequate.
4.4.5 Load Menu

The Load menu has one menu item, Load Modifiers. When selected, the Load Modifier dialog box (Fig. 4.4) is displayed to allow the user to input the three load modifiers:

\[
\eta_D = \begin{cases} 
1.05 & \text{for non-ductile components for strength limit state} \\
0.95 & \text{for ductile components for strength limit state} \\
1.0 & \text{for other limit states} 
\end{cases}
\]

\[
\eta_R = \begin{cases} 
1.05 & \text{for non-redundant members for strength limit state} \\
0.95 & \text{for redundant members for strength limit state} \\
1.0 & \text{for other limit states} 
\end{cases}
\]

\[
\eta_I = \text{declared by the client but should not be less than 0.95}
\]

Again, default values for these factors are provided as shown in Fig. 4.4.
4.4.6 Deck Menu

The Deck menu has three items: Deck Description, Deck Thickness and Deck Cover & Bar Diameter. All these menu items activate dialog boxes for entering the necessary data for deck design. Initially all three menu items are grayed (disabled). After the material properties and load modifiers have been entered, the first menu item in the Deck menu becomes active. Then after the deck description has been entered, the second menu item becomes active, and so on. Fig. 4.5 shows the dialog box for entering the general description of the deck.

The Deck Description dialog box contains three input fields: the number of deck spans, deck spacing and the length of overhanging cantilever. Since the program uses the distribution factor method provided by the new AASHTO Specifications for beam analysis, the number of deck spans should not be less than four, and the deck spacing should be between 1100 and 4900 mm, otherwise, the input data are invalid. When the OK button is pressed, the program checks for valid input data. If invalid data has been entered, an error message is displayed. Fig. 4.6 shows the error message when an invalid deck spacing was entered.
Fig. 4.5 Deck Description Dialog Box

Fig. 4.6 Error Message for Deck Description Dialog Box

Fig. 4.7 shows the Deck Thickness dialog box. There are four input fields in this dialog box: deck structural thickness, actual slab thickness, thickness of overhanging cantilever and thickness of future wearing surface. This dialog box also shows the minimum deck slab thickness calculated by the program according to the AASHTO Specifications as an aid to selecting a reasonable deck thickness. The deck thickness should not be less than the required minimum deck thickness, otherwise the selected deck thickness will not satisfy the requirements of the new AASHTO Specifications.
Fig. 4.7 Deck Thickness Dialog Box

Fig. 4.8 Deck Cover and Bar Diameter Dialog Box

Fig. 4.8 shows the Deck Cover and Bar Diameter dialog box. The three input fields are the top clear cover, the bottom clear cover and the bar size. The default values are also shown in Fig. 4.8.
4.4.7 Beam Menu

Initially the items in the Beam menu are grayed (disabled). The Beam menu becomes active after all the deck data has been entered. There are three menu items in the Beam menu: Beam Description, Beam Size, and Beam Cover and Stirrup Size. These three menu items activate dialog boxes for entering the necessary data for beam analysis and design. Fig. 4.10 shows the Beam Description dialog box. This dialog box contains five input fields: the number of beams, skew angle of bridge and the span lengths for the first three beam spans. As mentioned in Chapter 3, the program assumes that all the continuous beams to be designed are symmetric about their center line and there are only three different spans in the continuous beams (for three or four span continuous beams, there are only two different spans) as shown in Fig. 4.9. Thus, only the first three spans of a multi-span continuous beam have to be entered.

Fig. 4.9 Multi-span Continuous Beam
Fig. 4.10 Beam Description dialog box

Fig. 4.11 Beam Size dialog box
Fig. 4.11 shows the dialog box for entering the beam size. The two items to be entered are the depth of beam, and the width of the web. This dialog box also shows the minimum depth and width calculated by the program according to the new AASHTO Specifications. A beam size greater than the required minimum beam size should be entered in order to satisfy AASHTO requirements.

Fig. 4.12 shows the Beam Cover and Stirrup Size dialog box. The default values are also shown in the figure.

![Beam Cover and Stirrup Size dialog box](image)

Fig. 4.12 Beam Cover and Stirrup Size dialog box

### 4.4.8 Design Menu

The Design menu has two items: Deck Design and Beam Design. These menu items become active only after all the necessary deck and beam data have been entered. Fig. 4.13 shows the Deck Design dialog box.

When Deck Design menu item is selected, the program designs the deck using the method described in Section 3.3.2. The Deck Design dialog box gives the steel areas for both positive moment and negative moment, as well as the bar size entered in the Deck Program Architecture
Cover and Bar Diameter dialog box. The bar spacing can now be selected from Table C.4, Cross-sectional Area per Meter Width of Metric Bars of the Same Size, in Appendix C of the ACI Building Code/Commentary. The bar spacing entered should not be greater than 500 mm which is the maximum spacing according to the AASHTO Specifications.

![Deck Design Dialog Box](image)

**Fig. 4.13** Deck Design Dialog Box

In the dialog box shown in Fig. 4.13, the required area of steel for positive moment is 0.794421 mm^2/mm and for negative moment is 1.10186 mm^2/mm. The bar size entered by the user before is 15. From Table C.4 in ACI Building Code/Commentary, the required bar spacing for positive moment is 250 mm which provides an area of 0.8 mm^2/mm (> 0.794421 mm^2/mm). The required bar spacing for negative moment is 175 which provides an area of 1.143 mm^2/mm (> 1.10186 mm^2/mm).
When the Beam Design menu item is selected, the beam is designed and the reinforcement is calculated using the method described in Section 3.4.2. Fig. 4.14 shows the Beam Design dialog box. This dialog box gives the required areas of steel for maximum positive and negative moments in the beam. The required input is the number of bars and the bar sizes for maximum positive and negative moments respectively. These can be selected from the Table, ASTM Standard Reinforcing Bars, given in Appendix C of the ACI Building Code/Commentary.

For a given area of reinforcement, there are several choices of bar sizes and number of bars. It is recommended that for negative moment at the support, a larger number of smaller reinforcing bars be selected to satisfy crack control requirements. For positive moment in the middle of span, it is recommended that a smaller number of larger reinforcing bars be selected.

![Beam Design Dialog Box](image)

Fig. 4.14 Beam Design Dialog Box
4.4.9 Results Menu

The Results pull down menu has three menu items: Structure Description, Deck Design and Beam Design. Each menu item activates a different child window which displays the corresponding design results. The Structure Description menu item shows the bridge structure description, and deck and beam structure geometry. The Deck Design and Beam Design menu items display the deck and beam design results. As mentioned in Section 3.4.2, for continuous beams subjected to moving vehicular loads, if the span ratio is small, the maximum moment and shear usually occur at the first or second span. Thus, for beam design, the program only calculates the load effects for the first two spans, and only displays the results for the first two spans.

4.4.10 Result Windows

Fig. 4.15 shows the Structure Description child window which gives the general description of the bridge superstructure. This window displays the input data and shows the deck and beam structure geometry. The information contained in this window can be used to check the accuracy of the input data. If incorrect data has been entered, it can be corrected by selecting the corresponding menu item and changing the incorrect values. The revised results are displayed in the window immediately. It is not necessary to rerun the program to make corrections to the input data. The output for several test structures is shown in Appendix B.
Reinforced Concrete T-Bar Bridge Design - [Bridge1]

General Description of the Bridge

Number of Deck = 5
Deck Spacing = 2440.00 mm
Width of the Bridge = 13428.00 mm

Deck Structure Graphics

Fig. 4.15 Bridge Structure Description

Reinforced Concrete T-Bar Bridge Design - [Bridge1]

Deck Analysis and Design Results

Deck Structural Thickness = 190.00 mm
Deck Slab Thickness = 203.00 mm
Thickness of Overhanging Cantilever = 230.00 mm
Thickness of Future Wearing Surface = 75.00 mm
Deck Top Cover to the Reinforcement = 60.00 mm
Deck Bottom Cover to the Reinforcement = 25.00 mm

Load Modifier = 0.9476
Maximum Reaction in Deck = 187.36 N / mm
Maximum Positive Moment in Deck = 42655.62 N*mm / mm
Maximum Negative Moment in Deck = -50873.07 N*mm / mm

Fig. 4.16 Deck Design Results Windows

Program Architecture
Fig. 4.16 shows the Deck Design Results Window. The Deck Design Results displays the results of the deck design as well as the results of all the checks. There are two kinds of checks in the design of deck: ductility and capacity checks, and crack control check.

If the ductility and capacity check is not satisfied, the deck depth can be increased by selecting the Deck Thickness menu item, or the area of steel can be increased by selecting the Deck Design menu item.

For crack control, if the check is not satisfied, the Deck Thickness menu item can be selected to increase the deck depth, or the Deck Design menu item can be selected to choose a larger number of smaller size reinforcing bars.

![Beam Analysis and Design Results](image)

Beam Analysis and Design Results

<table>
<thead>
<tr>
<th>Number of beams =</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beam spacing =</td>
<td>10670.00 mm</td>
</tr>
<tr>
<td></td>
<td>12800.00 mm</td>
</tr>
<tr>
<td></td>
<td>10670.00 mm</td>
</tr>
<tr>
<td>Skew angle of bridge =</td>
<td>30.00 deg</td>
</tr>
<tr>
<td>Maximum positive service moment in the beam =</td>
<td>711116160.00</td>
</tr>
</tbody>
</table>

Fig. 4.17 Beam Design Results Window
Fig. 4.17 shows the Beam Design Results window. In addition to displaying beam design results, the beam design result window also provides information on the various checks: ductility, minimum reinforcement and flexural capacity, crack control, and fatigue.

For ductility, minimum reinforcement and flexural capacity check, if the minimum reinforcement or the flexural capacity is not satisfied, the Beam Design menu item can be selected to increase the reinforcement. If ductility is not satisfied, the Beam Size menu item can be selected to increase the beam depth or web width.

For crack control check, if crack control is not satisfied, the Beam Design menu item can be selected to choose a larger number of smaller reinforcing bars.

If fatigue check is not satisfied, it means that the stress range \( (f_{\text{max}} - f_{\text{min}}) \) for the fatigue limit state is greater than the allowable fatigue stress range. In this case the Beam Design menu item can be selected to increase the reinforcement or the Beam Size menu item can be selected to increase the beam depth.

### 4.4.11 Help Menu

![Help / About Dialog Box](image)

Fig. 4.18 Help / About Dialog Box
The only pull down item in the Help menu is the About menu item which displays information about the program (see Fig. 4.18).

In this chapter, the two types of classes used in the program, structural classes and interface classes, were described. All classes were derived from the Microsoft Foundation Class library. Also, the use of the program for designing reinforced concrete T-beam bridges is described.
Chapter 5
Results and Discussions

5.1 Introduction

In this chapter, a comparison of results obtained using the program developed in this thesis with those obtained from other sources is presented. Three test structures were used to verify the results obtained from the program. The first two structures consisted of T-beam bridges, which were designed in accordance with the new AASHTO Specifications. These structures were used to verify the accuracy of the design algorithms. The third structure was a continuous beam in the T-beam bridge. This structure was used to compare the continuous beam analysis results obtained using the influence line equations developed in this study and those obtained using commercial programs. Detailed results are given in appendix B.

5.2 Comparison of Deck and Beam Design Results

Since the new AASHTO LRFD Bridge Design Specifications using metric units were just published in August 1994, it is difficult to find commercial programs which use the new AASHTO Specifications. Fortunately, there are two available examples that use the new AASHTO Specifications and metric units to design reinforced concrete T-beam bridge superstructures [Barker, 1994]. Thus, these example structures were used as the test structures, and the design results of the deck and the beam obtained using the program were compared with the results obtained from these two examples.
5.2.1 General Description

The first test structure is a reinforced concrete T-beam bridge with three spans of 10670 mm, 12800 mm and 10670 mm beam, five equal spans of 2440 mm deck, a 990 mm long overhanging cantilever and a skew of 30° angle. The compression strength of concrete, $f_c$, is 30 MPa and the yield strength of steel, $f_y$, is 400 MPa. For the deck, there is 13 mm for sacrificial wear of concrete surface and 75 mm for future wearing surface consisting of bituminous overlay. The bridge is designed for a HL-93 live load. Figs. 5.1(a) and 5.1(b) show the deck and beam geometry respectively.

---

990mm 2440mm 2440mm 2440mm 2440mm 2440mm 990mm

---

Fig. 5.1(a) Deck geometry for the first test structure

---

10670mm 12800mm 10670mm

---

Fig. 5.1(b) Beam geometry for the first test structure
5.2.2 Section Selection

From the new AASHTO Specifications, the minimum deck structural thickness, \( h_{\text{min}} \), is 181 mm. For deck design, the deck slab section can be chosen as:

- deck structural thickness \( h_i = 190 \text{ mm} > 181 \text{ mm} \)
- deck actual thickness \( h_d = 190 + 13 = 203 \text{ mm} \)
- thickness of overhang \( h_c = 230 \text{ mm} > 203 \text{ mm} \)
- thickness of wearing surface \( h_c = 75 \text{ mm} \)

Also, from the new AASHTO Specifications, the minimum beam depth, \( h_{\text{min}} \), is 832 mm and the width of the beam web is recommended as more than 314 mm. For the beam design, the beam section can be chosen as:

- depth of the beam \( h = 990 \text{ mm} > 832 \text{ mm} \)
- depth of the beam web \( b = 350 \text{ mm} > 314 \text{ mm} \)

5.2.3 Comparison of Design Results

Table 5.1 shows the comparison of analysis results for the deck obtained from the program and the example [Barker, 1994]. In the example, the deck was analyzed by using the RISA-2D computer program and was designed using the new AASHTO approximate strip method. Table 5.2 shows the comparison of design results for the deck obtained from the program and the example. From Tables 5.1 and 5.2, it is seen that the maximum difference in the deck results is 1.02%. Thus for all practical purposes, the results provided by the computer program and those obtained from the example problem are the same. Thus, using the five equal span continuous structure influence line equations to calculate deck force effects gives accurate results.
Table 5.1 Comparison of deck analysis results for the first test structure

<table>
<thead>
<tr>
<th>Quantity Item</th>
<th>Program</th>
<th>Example</th>
<th>Difference(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC load $V_{100}$ (N/mm)</td>
<td>18.06</td>
<td>18.07</td>
<td>0.09</td>
</tr>
<tr>
<td>DC load $M_{100}$ (Nmm/mm)</td>
<td>-6662.8</td>
<td>-6669</td>
<td>0.09</td>
</tr>
<tr>
<td>DC load $M_{104}$ (Nmm/mm)</td>
<td>-1081.4</td>
<td>-1084</td>
<td>0.24</td>
</tr>
<tr>
<td>DC load $M_{200}$ (Nmm/mm)</td>
<td>-1248.6</td>
<td>-1248</td>
<td>0.05</td>
</tr>
<tr>
<td>DW load $V_{100}$ (N/mm)</td>
<td>2.76</td>
<td>2.76</td>
<td>0.00</td>
</tr>
<tr>
<td>DW load $M_{100}$ (Nmm/mm)</td>
<td>-307.99</td>
<td>-309</td>
<td>0.32</td>
</tr>
<tr>
<td>DW load $M_{104}$ (Nmm/mm)</td>
<td>609.34</td>
<td>611</td>
<td>0.03</td>
</tr>
<tr>
<td>DW load $M_{200}$ (Nmm/mm)</td>
<td>-972.40</td>
<td>-975</td>
<td>0.03</td>
</tr>
<tr>
<td>LL load $V_{100}$ (N/mm)</td>
<td>73.47</td>
<td>73.76</td>
<td>0.39</td>
</tr>
<tr>
<td>LL load $M_{100}$ (Nmm/mm)</td>
<td>-19289</td>
<td>-19290</td>
<td>0.00</td>
</tr>
<tr>
<td>LL load $M_{104}$ (Nmm/mm)</td>
<td>19528</td>
<td>19560</td>
<td>0.16</td>
</tr>
<tr>
<td>LL load $M_{200}$ (Nmm/mm)</td>
<td>-20953</td>
<td>-20740</td>
<td>1.02</td>
</tr>
<tr>
<td>Maximum Reaction (N/mm)</td>
<td>187.36</td>
<td>188.03</td>
<td>0.36</td>
</tr>
<tr>
<td>Maximum positive Moment (Nmm/mm)</td>
<td>42656</td>
<td>42723</td>
<td>0.16</td>
</tr>
<tr>
<td>Maximum negative Moment (Nmm/mm)</td>
<td>-50873</td>
<td>-50882</td>
<td>0.02</td>
</tr>
</tbody>
</table>

in which

DC = dead load of structural components and nonstructural attachments

DW = dead load of wearing surface and utilities

LL = vehicular live load

$V_{100}$ = Reaction of the first support

$M_{100}$ = negative moment on the first support

$M_{104}$ = positive moment in the middle of the first span

$M_{200}$ = negative moment on the second support
Table 5.2 Comparison of deck design results for the first test structure

<table>
<thead>
<tr>
<th>Quantity Item</th>
<th>Program</th>
<th>Example</th>
<th>Difference(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required As for maximum positive Moment (mm²/mm)</td>
<td>0.7944</td>
<td>0.7977</td>
<td>0.41</td>
</tr>
<tr>
<td>Required As for maximum negative moment (mm²/mm)</td>
<td>1.102</td>
<td>1.105</td>
<td>0.27</td>
</tr>
<tr>
<td>Provided positive moment (N mm/mm)</td>
<td>43630</td>
<td>43400</td>
<td>0.53</td>
</tr>
<tr>
<td>Provided negative moment (N mm/mm)</td>
<td>-52111</td>
<td>-51870</td>
<td>0.46</td>
</tr>
<tr>
<td>Ductility at positive moment regions</td>
<td>0.0962</td>
<td>0.0953</td>
<td>0.94</td>
</tr>
<tr>
<td>Ductility at negative moment regions</td>
<td>0.1597</td>
<td>0.1587</td>
<td>0.63</td>
</tr>
<tr>
<td>Distribution As (mm²/mm)</td>
<td>0.5388</td>
<td>0.5360</td>
<td>0.52</td>
</tr>
<tr>
<td>Temperature As (mm²/mm)</td>
<td>0.1903</td>
<td>0.1903</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Tables 5.3 and 5.4 show the comparison of analysis results for the beam obtained from the program and the example. In the example, RISA-2D program was used to analyze the beam. In the program, the beam was analyzed using the influence line equations generated in the Section 3.3. From the Table 5.3, it is seen that the maximum difference in the beam results is 0.78% which shows that the analysis results computed by the program agreed very closely to those computed by RISA-2D.

Table 5.5 shows the comparison of design results for the beam obtained from the program and the example. In the example, the beam was designed in accordance with the new AASHTO Specifications using hand calculation. Also, the modified compression field theory was employed to calculate the nominal shear resistance.
Table 5.3 Comparison of beam analysis results for moments for the first test structure

<table>
<thead>
<tr>
<th>Quantity Item</th>
<th>Program</th>
<th>Example</th>
<th>Difference(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC load-in-M104 (10^8 Nmm)</td>
<td>1.473</td>
<td>1.473</td>
<td>0.00</td>
</tr>
<tr>
<td>DC load-in-M200 (10^8 Nmm)</td>
<td>-2.516</td>
<td>-2.517</td>
<td>0.00</td>
</tr>
<tr>
<td>DC load-in-M205 (10^8 Nmm)</td>
<td>1.203</td>
<td>1.203</td>
<td>0.00</td>
</tr>
<tr>
<td>DC load-ex-M104 (10^8 Nmm)</td>
<td>1.992</td>
<td>1.995</td>
<td>0.15</td>
</tr>
<tr>
<td>DC load-ex-M200 (10^8 Nmm)</td>
<td>-3.403</td>
<td>-3.408</td>
<td>0.15</td>
</tr>
<tr>
<td>DC load-ex-M205 (10^8 Nmm)</td>
<td>1.626</td>
<td>1.629</td>
<td>0.18</td>
</tr>
<tr>
<td>LL load-in-M104 (10^8 Nmm)</td>
<td>4.792</td>
<td>4.795</td>
<td>0.06</td>
</tr>
<tr>
<td>LL load-in-M200 (10^8 Nmm)</td>
<td>-4.081</td>
<td>-4.054</td>
<td>0.67</td>
</tr>
<tr>
<td>LL load-in-M205 (10^8 Nmm)</td>
<td>4.552</td>
<td>4.552</td>
<td>0.00</td>
</tr>
<tr>
<td>LL load-ex-M104 (10^8 Nmm)</td>
<td>4.896</td>
<td>4.898</td>
<td>0.00</td>
</tr>
<tr>
<td>LL load-ex-M200 (10^8 Nmm)</td>
<td>-4.241</td>
<td>-4.208</td>
<td>0.78</td>
</tr>
<tr>
<td>LL load-ex-M205 (10^8 Nmm)</td>
<td>4.804</td>
<td>4.805</td>
<td>0.00</td>
</tr>
</tbody>
</table>

in which

DC = dead load of structural components and nonstructural attachments

LL = vehicular live load

in = force effect in interior beam

ex = force effect in exterior beam

M104 = positive moment in the middle of the first span

M200 = negative moment on the second support

M205 = positive moment in the middle of the second span

Results and Discussions
<table>
<thead>
<tr>
<th>Quantity Item</th>
<th>Program</th>
<th>Example</th>
<th>Difference(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC load-in-(V_{100}) (10^3 , N)</td>
<td>73.23</td>
<td>73.7</td>
<td>0.10</td>
</tr>
<tr>
<td>DC load-in-(V_{110}) (10^3 , N)</td>
<td>-120.4</td>
<td>120.4</td>
<td>0.00</td>
</tr>
<tr>
<td>DC load-in-(V_{200}) (10^3 , N)</td>
<td>116.18</td>
<td>116.2</td>
<td>0.00</td>
</tr>
<tr>
<td>DC load-ex-(V_{100}) (10^3 , N)</td>
<td>99.03</td>
<td>99.2</td>
<td>0.17</td>
</tr>
<tr>
<td>DC load-ex-(V_{110}) (10^3 , N)</td>
<td>162.8</td>
<td>163.1</td>
<td>0.18</td>
</tr>
<tr>
<td>DC load-ex-(V_{200}) (10^3 , N)</td>
<td>157.11</td>
<td>157.4</td>
<td>0.18</td>
</tr>
<tr>
<td>LL load-in-(V_{100}) (10^3 , N)</td>
<td>315.94</td>
<td>315.9</td>
<td>0.00</td>
</tr>
<tr>
<td>LL load-in-(V_{110}) (10^3 , N)</td>
<td>369.1</td>
<td>369.04</td>
<td>0.00</td>
</tr>
<tr>
<td>LL load-in-(V_{200}) (10^3 , N)</td>
<td>374.12</td>
<td>374.01</td>
<td>0.03</td>
</tr>
<tr>
<td>LL load-ex-(V_{100}) (10^3 , N)</td>
<td>291.65</td>
<td>291.42</td>
<td>0.08</td>
</tr>
<tr>
<td>LL load-ex-(V_{110}) (10^3 , N)</td>
<td>340.72</td>
<td>340.45</td>
<td>0.08</td>
</tr>
<tr>
<td>LL load-ex-(V_{200}) (10^3 , N)</td>
<td>345.36</td>
<td>345.04</td>
<td>0.09</td>
</tr>
</tbody>
</table>

in which

- **DC** = dead load of structural components and nonstructural attachments
- **LL** = vehicular live load
- **in** = force effect in interior beam
- **ex** = force effect in exterior beam
- \(V_{100}\) = shear at the left end of the first span
- \(V_{110}\) = shear at the right end of the first span
- \(V_{200}\) = shear at the left end of the second span
Table 5.5 Comparison of beam design results for the first test structure

<table>
<thead>
<tr>
<th>Quantity Item</th>
<th>Program</th>
<th>Example</th>
<th>Difference(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required As for maximum positive moment (mm²)</td>
<td>3703</td>
<td>3730</td>
<td>0.72</td>
</tr>
<tr>
<td>Required As for maximum negative moment (mm²)</td>
<td>4165.8</td>
<td>4180</td>
<td>0.36</td>
</tr>
<tr>
<td>Maximum service positive moment (N mm)</td>
<td>711.1*10⁶</td>
<td>711.8*10⁶</td>
<td>0.10</td>
</tr>
<tr>
<td>Maximum service negative moment (N mm)</td>
<td>-802.6*10⁶</td>
<td>-800.0*10⁶</td>
<td>0.32</td>
</tr>
<tr>
<td>Maximum factored positive moment (N mm)</td>
<td>1080.0*10⁶</td>
<td>1083.0*10⁶</td>
<td>0.28</td>
</tr>
<tr>
<td>Maximum factored negative moment (N mm)</td>
<td>-1161.0*10⁶</td>
<td>-1159.0*10⁶</td>
<td>0.17</td>
</tr>
<tr>
<td>Maximum provided positive moment (N mm)</td>
<td>1287.8*10⁶</td>
<td>1286.0*10⁶</td>
<td>0.14</td>
</tr>
<tr>
<td>Maximum provided negative moment (N mm)</td>
<td>-1294.1*10⁶</td>
<td>-1296.0*10⁶</td>
<td>0.15</td>
</tr>
<tr>
<td>Maximum dead load deflection (mm)</td>
<td>9.20</td>
<td>9.405</td>
<td>2.18</td>
</tr>
<tr>
<td>Maximum live load deflection (mm)</td>
<td>2.89</td>
<td>2.896</td>
<td>0.21</td>
</tr>
<tr>
<td>Shear in the middle of the first span (N)</td>
<td>206272</td>
<td>206372</td>
<td>0.05</td>
</tr>
<tr>
<td>Shear in the second support (N)</td>
<td>722137</td>
<td>723200</td>
<td>0.15</td>
</tr>
<tr>
<td>Stirrup spacing at the middle of the first span (mm)</td>
<td>504</td>
<td>504</td>
<td>0.00</td>
</tr>
<tr>
<td>Stirrup spacing at the second support (mm)</td>
<td>120</td>
<td>158</td>
<td>24.0</td>
</tr>
</tbody>
</table>

From Table 5.5, it is seen that the differences in the beam design results are typically less than 2% except for stirrup spacing near the supports where the difference is
24%. The reason for the large difference for this case is that different methods were used to calculate the nominal shear resistance in the program and in the example.

As mentioned in Chapter 3, when using the modified compression field theory to calculate the shear strength, two parameters need to be determined: $\beta$, a factor indicating the ability of diagonally cracked concrete to transmit tension, and $\theta$, the angle of inclination of diagonal compressive stress. The new AASHTO Specifications provides two methods for determining the parameters $\beta$ and $\theta$: the general procedure and a simplified procedure. In the program, the simplified procedure was used where $\beta$ is taken as 2.0 and $\theta$ is taken as 45°. In the example, however, the general procedure was used in which $\beta$ and $\theta$ were found from the figures and tables provided in the new AASHTO Specifications using the reinforcement strain $\varepsilon_x$ as the indicator of the longitudinal stiffness of the section and of the magnitude of the moment. According to the commentary to the AASHTO Specifications, the simplified procedure is conservative for small members containing transverse reinforcement since it does not consider the stresses that can be transmitted across diagonally cracked concrete. Thus, the program gave smaller beam shear stirrup spacing near the supports compared to that of the example.

The second test structure is also a reinforced concrete T-beam bridge. It has three spans of 10000 mm, 12000 mm and 10000 mm for the beams, four equal spans of 2950 mm for the deck, a 1190 mm long overhanging cantilever and a skew of 40° angle. Other data are the same as those for the first test structure. Figs. 5.2(a) and 5.2(b) show the deck and beam geometry respectively.
The comparison of design results for the deck and the beam obtained from the program and the example is given in Tables. 5.6 and 5.7. The maximum difference in the deck design was 0.65%. The difference in the beam design results was less than 2.5% expect for the stirrup spacing near the supports, which was 23.8%. Again, the program gave smaller beam shear stirrup spacing near the supports compared to that of the example.
Table 5.6 Comparison of deck design results for the second test structure

<table>
<thead>
<tr>
<th>Quantity Item</th>
<th>Program</th>
<th>Example</th>
<th>Difference(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum positive Moment (N/mm/mm)</td>
<td>50112.4</td>
<td>50347</td>
<td>0.47</td>
</tr>
<tr>
<td>Maximum negative Moment (N/mm/mm)</td>
<td>-73899.2</td>
<td>-74084</td>
<td>0.25</td>
</tr>
<tr>
<td>Required As for maximum positive Moment (mm²/mm)</td>
<td>0.8774</td>
<td>0.8810</td>
<td>0.41</td>
</tr>
<tr>
<td>Required As for maximum negative moment (mm²/mm)</td>
<td>1.4902</td>
<td>1.50</td>
<td>0.65</td>
</tr>
<tr>
<td>Provided positive moment (N mm/mm)</td>
<td>57585.6</td>
<td>57294</td>
<td>0.51</td>
</tr>
<tr>
<td>Provided negative moment (N mm/mm)</td>
<td>-76658.3</td>
<td>-76291</td>
<td>0.48</td>
</tr>
<tr>
<td>Ductility at positive moment regions</td>
<td>0.1130</td>
<td>0.1125</td>
<td>0.45</td>
</tr>
<tr>
<td>Ductility at negative moment regions</td>
<td>0.2082</td>
<td>0.2070</td>
<td>0.58</td>
</tr>
<tr>
<td>Distribution As (mm²/mm)</td>
<td>0.6736</td>
<td>0.6700</td>
<td>0.54</td>
</tr>
<tr>
<td>Temperature As (mm²/mm)</td>
<td>0.20</td>
<td>0.20</td>
<td>0.00</td>
</tr>
</tbody>
</table>

The comparison of results for the two test structures indicates that the analysis results for both the deck and beam obtained from the program and the examples are essentially the same. Design results for the deck and beam also agreed very closely with the exception of the beam shear stirrup spacing near the supports. For this case the program gives conservative stirrup spacing compared to the examples.
<table>
<thead>
<tr>
<th>Quantity Item</th>
<th>Program</th>
<th>Example</th>
<th>Difference(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required As for maximum positive moment (mm²)</td>
<td>3824.7</td>
<td>3811</td>
<td>0.36</td>
</tr>
<tr>
<td>Required As for maximum negative moment (mm²)</td>
<td>4257.4</td>
<td>4215</td>
<td>1.00</td>
</tr>
<tr>
<td>Maximum service positive moment (N mm)</td>
<td>734.4*10⁶</td>
<td>735.4*10⁶</td>
<td>0.14</td>
</tr>
<tr>
<td>Maximum service negative moment (N mm)</td>
<td>-820.2*10⁶</td>
<td>-816.1*10⁶</td>
<td>0.50</td>
</tr>
<tr>
<td>Maximum factored positive moment (N mm)</td>
<td>1119.4*10⁶</td>
<td>1123.4*10⁶</td>
<td>0.36</td>
</tr>
<tr>
<td>Maximum factored negative moment (N mm)</td>
<td>-1191.8*10⁶</td>
<td>-1187.3*10⁶</td>
<td>0.38</td>
</tr>
<tr>
<td>Maximum provided positive moment (N mm)</td>
<td>1287.0*10⁶</td>
<td>1300.0*10⁶</td>
<td>1.00</td>
</tr>
<tr>
<td>Maximum provided negative moment (N mm)</td>
<td>-1294.0*10⁶</td>
<td>-1312.0*10⁶</td>
<td>1.37</td>
</tr>
<tr>
<td>Maximum dead load deflection (mm)</td>
<td>8.476</td>
<td>8.28</td>
<td>2.37</td>
</tr>
<tr>
<td>Maximum live load deflection (mm)</td>
<td>2.87</td>
<td>2.80</td>
<td>2.50</td>
</tr>
<tr>
<td>Shear in the middle of the first span (N)</td>
<td>246494</td>
<td>246580</td>
<td>0.04</td>
</tr>
<tr>
<td>Shear at the second support (N)</td>
<td>823929</td>
<td>824649</td>
<td>0.09</td>
</tr>
<tr>
<td>Stirrup spacing at the middle of the first span (mm)</td>
<td>504</td>
<td>504</td>
<td>0.00</td>
</tr>
<tr>
<td>Stirrup spacing at the second support (mm)</td>
<td>99.02</td>
<td>130</td>
<td>23.8</td>
</tr>
</tbody>
</table>
5.3 Comparison of Continuous Beam Analysis Results

The third test structure is a five span continuous beam. This was used to compare analysis results obtained using the influence line equations developed in this study and those obtained using two commercial programs: RISA-2D [RISA Technologies, 1993] and SAP90 [Computers and Structures Inc., 1991].

5.3.1 General Description

The test structure is a five span symmetric continuous beam in a T-beam bridge. The beam spans were 8300 mm, 10000 mm, 12000 mm, 10000 mm and 8300 mm and the span ratio was 1.2. The exterior supports are pinned while all interior supports are roller supports. Fig. 5.3 shows the beam geometry.

![Continuous beam geometry](image)

Fig. 5.3 Continuous beam geometry

5.3.2 Loading

The loading on the beam consisted of the following: uniform dead load, uniform design lane load and design truck or design tandem load. The deck slab section chosen was: deck actual thickness $h_d = 203$ mm, thickness of overhang $h_c = 230$ mm and there is no wearing surface. The beam section was chosen as: depth of the beam $h = 990$ mm and width of the beam web $b = 350$ mm. Thus, the uniform dead load was 24.54 kN/m for exterior beams (see Fig. 5.4 (a)), the uniform design lane load was 9.3 kN/m.
(a) Uniform Dead Load

(b) Uniform Design Lane Load

(c) Design Truck Load

(d) Design Tandem Load

Fig. 5.4 Applied loads on continuous beam
(see Fig. 5.4 (b)), the design truck loads consist of twin 145 kN concentrated loads plus 35 kN concentrated load as shown in Fig. 5.4 (c). The design tandem loads consist of two concentrated loads of 110 kN as shown in Fig. 5.4 (d).

### 5.3.3 Distribution Factors

In the program, the distribution factor method was used for moment and shear analysis. The test structure distribution factors for moment and shear, assuming the deck spacing to be 2440 mm and the length of overhanging cantilever to be 990 mm, are shown in Table 5.8.

<table>
<thead>
<tr>
<th>Distribution Factors</th>
<th>L₁ = 8333 (mm)</th>
<th>L₁₂ = 9167 (mm)</th>
<th>L₂ = 10000 (mm)</th>
<th>L₂₃ = 11000 (mm)</th>
<th>L₃ = 12000 (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interior beam factor for M</td>
<td>0.7802</td>
<td>0.7669</td>
<td>0.75494</td>
<td>0.7421</td>
<td>0.7306</td>
</tr>
<tr>
<td>Exterior beam factor for M</td>
<td>0.7802</td>
<td>0.7669</td>
<td>0.7623</td>
<td>0.7623</td>
<td>0.7623</td>
</tr>
<tr>
<td>Interior beam factor for V</td>
<td>0.8258</td>
<td>0.8258</td>
<td>0.8258</td>
<td>0.8258</td>
<td>0.8258</td>
</tr>
<tr>
<td>Exterior beam factor for V</td>
<td>0.7623</td>
<td>0.7623</td>
<td>0.7623</td>
<td>0.7623</td>
<td>0.7623</td>
</tr>
</tbody>
</table>

### 5.3.4 Comparison of Analysis Results

Tables 5.9 and 5.10 give the comparison of continuous beam analysis results obtained using the program and two other commercial programs, RISA-2D and SAP90.

It is seen that the maximum difference in the beam analysis results between the program and RISA-2D was 0.81% which indicates that the results computed by the program and by RISA-2D are essentially the same. The maximum difference in the results...
obtained from the program and SAP90 was 1.41% which is slightly larger than that from RISA-2D. This is due to the fact that the coordinates of the influence line equations were determined fitting the equations to the results obtained using the RISA-2D program. However, the difference in the results are quite small. Thus, it can be concluded that for continuous structures subjected to dead load or moving load, the analysis results obtained from the program developed in the thesis and those obtained from RISA-2D and SAP90 are the same.

Table 5.9 Comparison of beam analysis results for moment for the third test structure

<table>
<thead>
<tr>
<th>Response Item</th>
<th>Program</th>
<th>RISA-2D</th>
<th>SAP90</th>
<th>Difference(%) RISA-2D SAP90</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC load-ex-M104 (kN m)</td>
<td>126.69</td>
<td>126.26</td>
<td>126.92</td>
<td>0.34 0.18</td>
</tr>
<tr>
<td>DC load-ex-M200 (kN m)</td>
<td>-190.46</td>
<td>-191.14</td>
<td>-190.21</td>
<td>0.35 0.13</td>
</tr>
<tr>
<td>DC load-ex-M205 (kN m)</td>
<td>79.46</td>
<td>78.82</td>
<td>80.48</td>
<td>0.81 1.27</td>
</tr>
<tr>
<td>LL load-in-M104 (kN m)</td>
<td>384.89</td>
<td>387.81</td>
<td>390.4</td>
<td>0.75 1.41</td>
</tr>
<tr>
<td>LL load-in-M200 (kN m)</td>
<td>-310.68</td>
<td>-312.05</td>
<td>-308.54</td>
<td>0.44 0.69</td>
</tr>
<tr>
<td>LL load-in-M205 (kN m)</td>
<td>368.92</td>
<td>371.76</td>
<td>373.05</td>
<td>0.76 1.11</td>
</tr>
<tr>
<td>LL load-ex-M104 (kN m)</td>
<td>384.89</td>
<td>387.81</td>
<td>390.4</td>
<td>0.75 1.41</td>
</tr>
<tr>
<td>LL load-ex-M200 (kN m)</td>
<td>-310.68</td>
<td>-312.05</td>
<td>-308.54</td>
<td>0.44 0.69</td>
</tr>
<tr>
<td>LL load-ex-M205 (kN m)</td>
<td>372.51</td>
<td>375.40</td>
<td>376.71</td>
<td>0.77 1.11</td>
</tr>
<tr>
<td>Maximum positive moment</td>
<td>831.92</td>
<td>836.49</td>
<td>841.85</td>
<td>0.55 1.18</td>
</tr>
<tr>
<td>in the first span (kN m)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum negative moment</td>
<td>-781.77</td>
<td>-785.01</td>
<td>-777.71</td>
<td>0.41 0.52</td>
</tr>
<tr>
<td>in the first span (kN m)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum positive moment</td>
<td>751.22</td>
<td>755.48</td>
<td>759.84</td>
<td>0.56 1.13</td>
</tr>
<tr>
<td>in the second span (kN m)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum negative moment</td>
<td>-781.77</td>
<td>-785.01</td>
<td>-777.71</td>
<td>0.41 0.52</td>
</tr>
<tr>
<td>in the second span (kN m)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 5.10 Comparison of beam analysis results for shear for the third test structure

<table>
<thead>
<tr>
<th>Response Item</th>
<th>Program</th>
<th>RISA-2D</th>
<th>SAP90</th>
<th>Difference(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RISA-2D</td>
</tr>
<tr>
<td>DC load-ex-V100 (kN)</td>
<td>78.90</td>
<td>78.81</td>
<td>78.92</td>
<td>0.11</td>
</tr>
<tr>
<td>DC load-ex-V110 (kN)</td>
<td>124.79</td>
<td>124.87</td>
<td>124.76</td>
<td>0.06</td>
</tr>
<tr>
<td>DC load-ex-V200 (kN)</td>
<td>114.98</td>
<td>115.34</td>
<td>115.26</td>
<td>0.31</td>
</tr>
<tr>
<td>LL load-in-V100 (kN)</td>
<td>249.51</td>
<td>251.37</td>
<td>250.97</td>
<td>0.74</td>
</tr>
<tr>
<td>LL load-in-V110 (kN)</td>
<td>-289.33</td>
<td>-290.60</td>
<td>-290.81</td>
<td>0.43</td>
</tr>
<tr>
<td>LL load-in-V200 (kN)</td>
<td>300.37</td>
<td>300.96</td>
<td>300.81</td>
<td>0.19</td>
</tr>
<tr>
<td>LL load-ex-V100 (kN)</td>
<td>230.33</td>
<td>232.04</td>
<td>231.67</td>
<td>0.74</td>
</tr>
<tr>
<td>LL load-ex-V110 (kN)</td>
<td>267.09</td>
<td>268.25</td>
<td>268.44</td>
<td>0.43</td>
</tr>
<tr>
<td>LL load-ex-V200 (kN)</td>
<td>277.28</td>
<td>277.82</td>
<td>277.68</td>
<td>0.19</td>
</tr>
<tr>
<td>Maximum shear in the first span (kN)</td>
<td>623.40</td>
<td>625.52</td>
<td>625.72</td>
<td>0.34</td>
</tr>
<tr>
<td>Maximum shear in the second span (kN)</td>
<td>631.93</td>
<td>630.36</td>
<td>630.02</td>
<td>0.25</td>
</tr>
</tbody>
</table>

5.4 Discussions of Results

In the design of a T-beam bridge superstructure, the analysis of the continuous deck or beam subjected to moving loads is a difficult task. The general procedure used for computing the various load effects in continuous structures is the finite element analysis method. Most commercial programs use this approach. In order to simplify the analysis of the deck and beam in T-beam bridge design, the influence line equations were developed to obtain the corresponding force effects. This significantly reduces the computational efforts required and simplifies the analysis procedure. The maximum
difference in the analysis results was found to be less than 1.5% when compared with RISA-2D and SAP90. This shows that analysis results obtained using the influence line equations developed in this study are essentially the same as those obtained from other commercial programs.

A comparison of deck design results for two test structures indicates that the maximum difference in design results obtained from the program and the example problems of Barker [Barker, 1994] was less than 1%. Thus, for the design of bridge decks, the program gave the same results as the example problems.

Except for beam shear stirrup spacing near the supports, the beam design results obtained from the program agreed very closely with those for the solved examples. The maximum difference was 2.5%. The difference in beam shear stirrup spacing near the supports was due to the different methods used to calculate the nominal shear resistance. The program gives conservative values for stirrup spacing for this case. Thus, for the design of the beams, the program gave the same results as the solved examples.

In this chapter, the analysis and design results for three reinforced concrete T-beam bridges were compared. The deck and beam design results obtained using the program developed were compared with the solved examples in Barker [Barker, 1994]. The beam analysis results were compared with those obtained from two commercial programs, RISA-2D and SAP90. All of the comparisons gave satisfactory results.
Chapter 6
Summary and Conclusions

The major objectives of this study were to develop a Windows based object oriented program for the design of reinforced concrete T-beam bridges using the new AASHTO LRFD Bridge Design Specifications. This chapter presents a summary of the work and gives the main conclusions of the study.

6.1 Summary

The program developed in this study is capable of analyzing and designing the superstructure including the deck and the beam of reinforced concrete T-beam bridges in accordance with the newly-adopted AASHTO LRFD Bridge Design Specifications.

The analysis of continuous structures subjected to the moving loads is a difficult task. To obtain load effects on the deck and beam, the program can read in analysis results computed by the commercial structure analysis programs. It can also obtain the load effects using interpolating influence line functions. The second option provides an integrated environment for both the analysis and design of the continuous structures.

The program can perform both deck and beam design. For deck design, the AASHTO approximate strip method is used. The program computes maximum positive and negative moments, and the required area of reinforcing steel. For beam design, the program computes maximum positive and negative moments, and the shear at the left and right end of the first two beam spans. It also determines the required area of reinforcing steel for maximum positive and negative moments, and the stirrup spacing at the mid-span.
and near the support. In addition, the program also checks code requirements for ductility, minimum reinforcement, cracking, fatigue, flexural capacity and shear capacity.

The program runs in the Windows environment and it was developed in C++ using the object oriented programming approach. The deck and beam design process is interactive. Menus, dialog boxes and child windows are used throughout the program. All the analysis and design results can be checked during program execution. Also, the input data and design parameters can easily be modified at any stage of the design process.

6.2 Conclusions

The efficient development and successful performance of the program demonstrates the many advantages of object oriented programming and the attractive characteristics of C++ language. All of the structural classes and the interface were derived from the Microsoft Foundation Class library, which significantly reduced the effort required for developing the program.

Based on a comparison with several example problems it was shown that the program can correctly analyze and design the bridge deck and beam in accordance with the new AASHTO LRFD Bridge Design Specifications. The program gave the same analysis results as compared with results from two commercial programs, RISA-2D and SAP90. The program gave similar design results as those from several solved examples.

Although, the program developed in this study can only design reinforced concrete T-beam bridge superstructures, the approach used here can easily be adapted to other types of bridges. Since object oriented programming has so many advantages over conventional procedural programming, it is anticipated that in the future, many engineering applications will be developed using object oriented programming.

Summary and Conclusions
References


Appendix A

Class Definitions for the Program

A.1 Class for Bridge Structure (STRUCT.H)

////////////////////////////////////////////
// STRUCT.H -- Class definition for the bridge structure
////////////////////////////////////////////

#ifndef STRUCTURE_H
#define STRUCTURE_H

class Structure:public CObject
{
  // ------------------------ Data members ------------------------

public:
  CString title;     // name of project (input)
  float hmin;       // minimum deck thickness
  float bmin;       // minimum width of beam web
  float hmin;       // minimum depth of beam

  BOOL bHaveData;

protected:
  // Material properities
  float fc;         // compressive strength of concrete (input)
  float fy;         // yield strength of steel (input)

  // Data related to the calculation of loads
  float dLoadModi;  // factor relating to ductility (input)
  float rLoadModi;  // factor relating to redundancy (input)
  float iLoadModi;  // factor relating to operational importance (input)

  // Data related to the description of the deck
  int  numDeck;     // number of decks (input)
  float deckSpan;   // deck spacing (input)
  float overhang;   // length of overhang cantilever (input)
  float hi;         // deck structural thickness for design (input)
  float hd;         // actual deck thickness for load calculation (input)
  float hc;         // thickness of overhanging cantilever (input)
  float hw;         // thickness of future wearing surface (input)
  float topCover;   // deck top cover to reinforcement (input)
float botCover; // deck bottom cover to reinforcement (input)
int dDeckNum; // designation number of reinforcing bar in deck (input)

// Data related to the description of the beam
int numBeam; // number of beams (input)
float skewAngle; // skew angle of the bridge (input)
float beamSpan[10]; // beam spacing (input)
float b; // width of beam stem (input)
float h; // depth of beam (input)
float cover; // cover to reinforcement in beam (input)
int stirSizeNum; // designation number of stirrup in beam (input)

// User enters spacing of bars in deck for positive and negative moments
float posSpace; // positive reinforcing bar spacing (input)
float negSpace; // negative reinforcing bar spacing (input)

// User enters areas, number and diameter of bars for max +M in beam
int numBar104; // number of bars for max +M (input)
int dBarNum104; // designation number of bars for max +M (input)

// User enters areas, number and diameter of bars for max -M in beam
int numBar200; // number of bars for max -M (input)
int dBarNum200; // designation number of bars for max -M (input)

float spanRatio; // beam span ratio (spanRatio = beamSpan[1]/beamSpan[0])
float width; // total roadway width of the bridge
float length; // total length of the bridge
int numLane; // number of design lane
float multiFactor; // multiple presence factor "m"
float loadModifier; // factor relating to ductility, redundancy
// and operational importance "eta"
float Es; // Young's modulus for steel (Es = 200000 MPa)
float Ec; // Young's modulus for concrete
int n; // n = Es / Ec
float beta1; // ratio of the depth of equivalent rectangular
// stress block to depth of actual compression zone
float dDeckBar; // diameter of reinforcing bar in deck

DECLARE_SERIAL(Structure)

// ---------------------------------------- Member functions ----------------------------------------

public:
    // constructors and destructor
    Structure(); // default constructor
    Structure(CString title1, int numDeck1, int numBeam1, float deckSpan1,
        float beamSpan1, float beamSpan2, float beamSpan3,
        float overhang1, float skewAngle1,
        float dLoadModi1, float rLoadModi1, float iLoadModi1,

float fc1, float fy1, float hi1, float hd1, float hc1, float hw1,
float topCover1, float botCover1, int dDeckNum1,
float b1, float h1, float cover1, int stirSizeNum1,
float posSpace1, float negSpace1,
int numBar1041, int dBarNum1041, int numBar2001, int dBarNum2001);  
// constructor

// computational routines
float calHimin(); // calculate minimum deck thickness
float calBmin();  // calculate minimum width of beam web
float calHmin();  // calculate minimum depth of beam
float calLength(); // calculate total length of the bridge
float calRatio();  // calculate beam span ratio
float calWidth();  // calculate roadway width of the bridge
int calLane();    // calculate number of design lane
float calILoadModi();  // calculate load modifier "eta"
float calMultFactor();  // calculate multiple presence factor "m"
float calEc();   // calculate Young's modulus of concrete
float calBeta1();    // calculate factor beta1
int caln();      // calculate n = Es / Ec
float calDiameter(int number);  // calculate nominal diameter of reinforcing bar

// functions to set design input data
void setMaterialData(float fc1, float fy1);
void setLoadData(float dLoadModi1, float rLoadModi1, float iLoadModi1);
void setDeckData(int numDeck1, float deckSpan1, float overhang1);
void setDeckData1(float hi1, float hd1, float hc1, float hw1);
void setDeckData2(float topCover1, float botCover1, int dDeckNum1);
void setDeckData3(float posSpace1, float negSpace1);
void setBeamData(int numBeam1, float skewAngle1, float beamSpan1,
float beamSpan2, float beamSpan3);
void setBeamData1(float b1, float h1);
void setBeamData2(float cover1, int stirSizeNum1);
void setBeamData3(int numBar1041, int dBarNum1041, int numBar2001, int dBarNum2001);

// functions to obtain design data from user
void getMaterialData(float &fc1, float &fy1);
void getLoadData(float &dLoadModi1, float &rLoadModi1, float &iLoadModi1);
void getDeckData(int &numDeck1, float &deckSpan1, float &overhang1);
void getDeckInput1(float &hi1, float &hd1, float &hc1, float &hw1);
void getDeckInput2(float &topCover1, float &botCover1, int &dDeckNum1);
void getDeckInput3(float &posSpace1, float &negSpace1);
void getBeamData(int &numBeam1, float &beamSpan1, float &beamSpan2,
float &beamSpan3, float &skewAngle1);
void getBeamInput1(float &b1, float &h1);
void getBeamInput2(float &cover1, int &stirSizeNum1);
void getBeamInput3(int &numBar1041, int &dBarNum1041,
int &numBar2001, int &dBarNum2001);

// functions to display bridge description and structure
void Show(CDC* pDC);
virtual void Serialize(CArchive& ar);

};
#endif

A.2 Class for Deck Analysis and Design (DECK.H)

//****************************************************************************
// DECK.H -- Class definition for deck analysis and design
//****************************************************************************

#ifndef DECK_H
#define DECK_H

class Deck: public Structure
{
  // -------------------------- Data members --------------------------

private:
  // Moment and reaction effects of dead loads
  float DCLoadR200;   // R200 due to structural and nonstructural component dead load
  float DCLoadM200;   // M200 due to structural and nonstructural component dead load
  float DCLoadM204;   // M204 due to structural and nonstructural component dead load
  float DCLoadM300;   // M300 due to structural and nonstructural component dead load
  // (DC = deck + overhang + barrier)
  float DWLoadR200;   // R200 due to wearing surface and utilities dead load
  float DWLoadM200;   // M200 due to wearing surface and utilities dead load
  float DWLoadM204;   // M204 due to wearing surface and utilities dead load
  float DWLoadM300;   // M300 due to wearing surface and utilities dead load

  // Width of equivalent interior strips
  float equiWidth1;   // overhang equivalent width
  float equiWidth2;   // + Moment equivalent width
  float equiWidth3;   // - Moment equivalent width

  // Moment and reaction effects of live loads
  float liveLoadR200; // max reaction at exterior support due to live load
  float liveLoadM200; // max negative moment at exterior support due to live load
  float liveLoadM204; // max positive moment in 1st span due to live load
  float liveLoadM300; // max negative moment at 1st interior support due to live load

  // Maximum force effects in the deck
  float maxReaction;   // max reaction at exterior support (due to combined loads)

Appendix A  Class Definitions for the Program 97
float maxPosMom;  // max positive moment (due to combined loads)
float maxNegMom;  // max negative moment (due to combined loads)

// Areas of reinforcing bar in deck slab computed by program
float AsDeckPos;  // area of reinforcing steel for positive moment
float AsDeckNeg;  // area of reinforcing steel for negative moment

float AsDist;  // distribution reinforcement in deck
float AsTemp;  // shrinkage and temperature reinforcement in deck

// Check if input areas of reinforcing bar in deck slab is OK
float posAs;  // provided area of reinforcement for pos moment
float posDuct;  // positive bar ductility
float posMu;  // provided maximum positive moment
int posCheck;  // check ductility for positive moment
  // 1 -- both ductility and strength are OK
  // 2 -- ductility is OK but strength is NG
  // 0 -- ductility is NG and no strength calculation

float negAs;  // provided area of reinforcement for neg moment
float negDuct;  // negative bar ductility
float negMu;  // provided maximum negative moment
int negCheck;  // check ductility for negative moment
  // 1 -- both ductility and strength are OK
  // 2 -- ductility is OK but strength is NG
  // 0 -- ductility is NG and no strength calculation

public:

  BOOL  d_bHaveData;

DECLARE_SERIAL(Deck)

// ----------------------------- Member functions -----------------------------

public:

  Deck();  // default constructor
  Deck(CString title1,int numDeck1,int numBeam1,float deckSpan1,
    float beamSpan1,float beamSpan2,float beamSpan3,
    float overhang1,float skewAngle1,
    float dLoadModi1,float rLoadModi1,float iLoadModi1,
    float fc1,float fy1,float hi1,float hd1,float hc1,float hw1,
    float topCover1,float botCover1,int dDeckNum1,
    float b1,float h1,float cover1,int stirSizeNum1,
    float posSpace1,float negSpace1,
    int numBar1041,int dBarNum1041,int numBar2001,int dBarNum2001);
    // constructor

  // Calculate moments and reactions due to structural and non-structural component load
  void calDC(float *pt_DCLoadR200,float *pt_DCLoadM200,float *pt_DCLoadM204,
    float *pt_DCLoadM300);
  // Calculate moments and reactions due to wearing surface and utilities load
  void calDW(float *pt_DWLoadR200,float *pt_DWLoadM200,float *pt_DWLoadM204,
    float *pt_DWLoadM204,
```c
float *pt_DWLoadM300);

// Calculate width of equivalent interior strips
void calEquWidth(float *pt_equiWidth1, float *pt_equiWidth2, float *pt_equiWidth3);

// Calculate moments and reactions due to live load components
void calLive(float *pt_liveLoadR200, float *pt_liveLoadM200,
             float *pt_liveLoadM204, float *pt_liveLoadM300);

// Calculate max reaction, negative and positive moment due to combined load
void calEffect(float *pt_maxReact, float *pt_maxPosMom, float *pt_maxNegMom);
void getEffect();

// Calculate areas of reinforcing bar for positive and negative moment in deck
void calDeckAs(float *pt_AsDeckPos, float *pt_AsDeckNeg);
void getDeckAs(float &AsDeckPos, float &AsDeckNeg1);

// Check if ductility & strength is OK
void checkPosMom(int *pt_posCheck, float *pt_posAs, float *pt_posDuct, float *pt_posMu);
    // check for positive moment in deck
void checkNegMom(int *pt_negCheck, float *pt_negAs, float *pt_negDuct, float *pt_negMu);
    // check for negative moment in deck

// Calculate area of distribution reinforcement in deck slab
float calDistAs(void);
// Calculate area of shrinkage and temperature reinforcement in deck slab
float calTempAs(void);

// Control of cracking
int crackPos(void);  // control of cracking in positive moment reinforcement
int crackNeg(void);  // control of cracking in negative moment reinforcement

// Display the deck design results
void Show(CDC* pDC);
```

### A.3 Class for Beam Analysis and Design (BEAM.H)

```c
    //*******************************************************************************
    // BEAM.H -- Class definition for beam analysis and design
    //*******************************************************************************

#ifndef Beam_H
#define Beam_H

class Beam: public Structure

 Appendix A  Class Definitions for the Program 99
```
private:

// Distribution factor for moment
float mgLM1; // moment distribution factor in interior beam L1
float mgLM12; // moment distribution factor in interior beam L1 & L2
float mgLM2; // moment distribution factor in interior beam L2
float mgLM23; // moment distribution factor in interior beam L2 & L3
float mgLM3; // moment distribution factor in interior beam L3

float mgEM1; // moment distribution factor in exterior beam L1
float mgEM12; // moment distribution factor in exterior beam L1 & L2
float mgEM2; // moment distribution factor in exterior beam L2
float mgEM23; // moment distribution factor in exterior beam L2 & L3
float mgEM3; // moment distribution factor in exterior beam L3

float skewM1; // skew angle reduction factor for moment in beam L1
float skewM12; // skew angle reduction factor for moment in beam L1 & L2
float skewM2; // skew angle reduction factor for moment in beam L2
float skewM23; // skew angle reduction factor for moment in beam L2 & L3
float skewM3; // skew angle reduction factor for moment in beam L3

// Distribution factor for shear
float mgIV; // distribution factor for shear in interior beam
float mgEV; // distribution factor for shear in exterior beam
float skewV; // reduction factor for shear due to skew angle

// Distributed live load moment
float liveLM104; // positive distributed live load moment
    // in interior beam at first span
float liveLM200; // negative distributed live load moment
    // in interior beam at first interior support
float liveLM205; // positive distributed live load moment
    // in interior beam at second span
float liveLM104; // positive distributed live load moment
    // in exterior beam at first span
float liveLM200; // negative distributed live load moment
    // in exterior beam at first interior support
float liveLM205; // positive distributed live load moment
    // in exterior beam at second span

// Distributed live load shear
float liveV100; // distributed live load shear in interior beam
    // at left end of first span
float liveV110; // distributed live load shear in interior beam
    // at right end of first span
float liveV200; // distributed live load shear in interior beam
    // at left end of second span
float liveV100; // distributed live load shear in exterior beam
float liveExV110; // at left end of first span
float liveExV200; // at left end of second span

// Dead load moment and shear due to structural and nonstructural components
float DCinM104; // pos DC load moment in interior beam at first span
float DCinM200; // neg DC load moment in interior beam at 1st interior support
float DCinM205; // pos DC load moment in interior beam at second span
float DCinV100; // DC load shear in interior beam at left of 1st span
float DCinV110; // DC load shear in interior beam at right of 1st span
float DCinV200; // DC load shear in interior beam at left of 2nd span

float DCExM104; // pos DC load moment in exterior beam at first span
float DCExM200; // neg DC load moment in exterior beam at 1st interior support
float DCExM205; // pos DC load moment in exterior beam at second span
float DCExV100; // DC load shear in exterior beam at left of 1st span
float DCExV110; // DC load shear in exterior beam at right of 1st span
float DCExV200; // DC load shear in exterior beam at left of 2nd span

// Dead load moment and shear due to wearing surface and utilities components
float DWinM104; // pos DW load moment in interior beam at first span
float DWinM200; // neg DW load moment in interior beam at 1st interior support
float DWinM205; // pos DW load moment in interior beam at second span
float DWinV100; // DW load shear in interior beam at left of 1st span
float DWinV110; // DW load shear in interior beam at right of 1st span
float DWinV200; // DW load shear in interior beam at left of 2nd span

float DWExM104; // pos DW load moment in exterior beam at first span
float DWExM200; // neg DW load moment in exterior beam at 1st interior support
float DWExM205; // pos DW load moment in exterior beam at second span
float DWExV100; // DW load shear in exterior beam at left of 1st span
float DWExV110; // DW load shear in exterior beam at right of 1st span
float DWExV200; // DW load shear in exterior beam at left of 2nd span

// Factored force effects in the beam
float M104; // factored maximum positive moment at first span
float M200; // factored maximum negative moment at second support
float M205; // factored maximum positive moment at second span
float V100; // factored shear at left of first span
float V110; // factored shear at right of first span
float V200; // factored shear at left of second span

// Effective flange width
float bi104; // effective flange width in interior beam for M104
float be104; // effective flange width in exterior beam for M104
float bi200; // effective flange width in interior beam for M200
float be200; // effective flange width in exterior beam for M200
float bi205; // effective flange width in interior beam for M205
float be205; // effective flange width in exterior beam for M205
// Data for beam design output
int crackingM104; // crack check for Max +M, 1 -- fs < fsa OK
        // 0 -- fs > fsa NG
int crackingM200; // crack check for Max -M, 1 -- fs < fsa OK
        // 0 -- fs > fsa NG
float As104;    // designed As for service positive moment Ms104
float As200;    // designed As for service negative moment Ms200
float Ms104;    // service positive moment
float Ms200;    // service negative moment

// Investigate fatigue
int fatigCheck104; // fatigue check for Max +M, 1 -- OK
        // 0 -- NG
int fatigCheck205; // fatigue check for Max +M, 1 -- OK
        // 0 -- NG
int fatigCheck200; // fatigue check for Max +M, 1 -- OK
        // 0 -- NG

// Deflection and camber
float E104;
float defLIVE104;
float defDEAD104;
float defLIVE205;
float defDEAD205;

// Investigate flexural strength limit state for M104 and M200
int flexCheck104, flexCheck200;
float AsM104, Mu104, duct104, rou104;
float AsM200, Mu200, duct200, rou200;
float dBar104, dBar200;

// Investigate shear strength limit state at location 200 and i04
int shearCheck104, shearCheck200;
float Vu200, MM200;
float Vu104, MM104;
float s104, s200;
float stirSize; // diameter of stirrups in beam
float stirArea; // area of stirrups in beam

public:

BOOL b_bHaveData;

DECLARE_SERIAL(Beam)

// ----------------------------- Member functions -----------------------------

public:

Beam(); // default constructor
Beam(CString title1, int numDeck1, int numBeam1, float deckSpan1,
    float beamSpan1, float beamSpan2, float beamSpan3,

Appendix A  Class Definitions for the Program 102
float overhang1, float skewAngle1,
float dLoadMod1, float rLoadMod1, float iLoadMod1,
float fc1, float fy1, float h1, float h1, float hc1, float hw1,
float topCover1, float botCover1, int dDeckNum1,
float b1, float h1, float cover1, int stirSizeNum1,
float posSpace1, float negSpace1,
int numBar1041, int dBarNum1041, int numBar2001, int dBarNum2001);

// constructor

void calDistIM(float *pt_mgIM1, float *pt_mgIM12, float *pt_mgIM2,
float *pt_mgIM23, float *pt_mgIM3);

// calculate moment distribution factor in interior beam
void calDistEM(float *pt_mgEM1, float *pt_mgEM12, float *pt_mgEM2,
float *pt_mgEM23, float *pt_mgEM3);

// calculate moment distribution factor in interior beam
void calSkewMom(float *pt_skewM1, float *pt_skewM12, float *pt_skewM2,
float *pt_skewM23, float *pt_skewM3);

// calculate skew reduction factor for moment
void calDistShear(float *pt_mgIV, float *pt_mgIV, float *pt_skewV);

// calculate skew reduction factor for shear

void calLiveM(float *pt_liveInM104, float *pt_liveExM104,
float *pt_liveInM200, float *pt_liveExM200,
float *pt_liveInM205, float *pt_liveExM205);

void calLiveV(float *pt_liveInV100, float *pt_liveExV100,
float *pt_liveInV110, float *pt_liveExV110,
float *pt_liveInV200, float *pt_liveExV200);

// Calculate dead load moment and shear
void calDCIn(float *pt_DCInM104, float *pt_DCInM200, float *pt_DCInM205,
float *pt_DCInV100, float *pt_DCInV110, float *pt_DCInV200);

// calculate DC load moment and shear in interior beam
void calDWin(float *pt_DWinM104, float *pt_DWinM200, float *pt_DWinM205,
float *pt_DWinV100, float *pt_DWinV110, float *pt_DWinV200);

// calculate DW load moment and shear in interior beam
void calDCEx(float *pt_DCExM104, float *pt_DCExM200, float *pt_DCExM205,
float *pt_DCExV100, float *pt_DCExV110, float *pt_DCExV200);

// calculate DC load moment and shear in exterior beam
void calDWeEx(float *pt_DWeExM104, float *pt_DWeExM200, float *pt_DWeExM205,
float *pt_DWeExV100, float *pt_DWeExV110, float *pt_DWeExV200);

// calculate DW load moment and shear in exterior beam

// Calculate effective flange width
void calEffFlange(float *pt_bi104, float *pt_be104,
float *pt_bi200, float *pt_be200,
float *pt_bi205, float *pt_be205);

void calEffect(float *pt_M104, float *pt_M200, float *pt_M205,
float *pt_V100, float *pt_V110, float *pt_V200);
// Crack control for M104 and M200
void serviceAs(float *pt_As104, float *pt_Ms104,
               float *pt_As200, float *pt_Ms200);
void crackCheck(int *pt_crackingM104, int *pt_crackingM200,
                float *pt_AsM104, float *pt_AsM200);
void getServiceAs(float &As1041, float &As2001);

// Investigate fatigue
void fatigueM104(int *pt_fatigCheck104, int *pt_fatigCheck205, int *pt_fatigCheck200);

// Calculate deflection and camber
void deflectM104(float *pt_EI104, float *pt_defLive104, float *pt_defDead104,
                 float *pt_defLive205, float *pt_defDead205);

// Investigate flexural strength limit state for M104 and M200
void flexureM104(int *pt_flexCheck104, float *pt_Mu104,
                 float *pt_duct104, float *pt_rout104);
void flexureM200(int *pt_flexCheck200, float *pt_Mu200,
                 float *pt_duct200, float *pt_rout200);

// Investigate shear strength limit state at location 104 and 200
void shearV104(int *pt_shearCheck104, float *pt_Vu104,
               float *pt_MM104, float *pt_s104);
void shearV200(int *pt_shearCheck200, float *pt_Vu200,
               float *pt_MM200, float *pt_s200);

// Display beam design results
void Show(CDC* pDC);

#endif

A.4 Class for Influence Line Equations (INFLINE.H)

/***************************************************************************
 // INFLINE.H -- Class definition for determining the influence line equations
 //***************************************************************************/

#ifndef INFLINE_H
#define INFLINE_H

class Inline:public CObject
{
    private:
        float pl;                // position of load on the deck or beam

        DECLARE_SERIAL(Inline)

Appendix A Class Definitions for the Program 104
public:
    Inline(); // default constructor

    // interpolating polynomials to represent the influence lines
    // Ratio = 1.0 (equal spans)
    // three - equal span continuous beams
    float span3V100(float pl);
    float span3M104(float pl);
    float span3M200(float pl);
    float span3M205(float pl);
    float span3V104(float pl);
    float span3V110(float pl);
    float span3V200(float pl);
    float span3V205(float pl);

    // four and four plus - equal span continuous beams
    float span5V100(float pl);
    float span5M104(float pl);
    float span5M200(float pl);
    float span5M205(float pl);
    float span5V104(float pl);
    float span5V110(float pl);
    float span5V200(float pl);
    float span5V205(float pl);

    // Ratio = 1.2 (beamSpan[1]/beamSpan[0] = beamSpan[2]/beamSpan[1] = 1.2)
    // three - span continuous beams
    float span32V100(float pl);
    float span32M104(float pl);
    float span32M200(float pl);
    float span32M205(float pl);
    float span32V104(float pl);
    float span32V110(float pl);
    float span32V200(float pl);
    float span32V205(float pl);

    // four and four plus - span continuous beams
    float span52V100(float pl);
    float span52M104(float pl);
    float span52M200(float pl);
    float span52M205(float pl);
    float span52V104(float pl);
    float span52V110(float pl);
    float span52V200(float pl);
    float span52V205(float pl);

    // Ratio = 1.4 (beamSpan[1]/beamSpan[0] = beamSpan[2]/beamSpan[1] = 1.4)
    // three - span continuous beams
    float span34V100(float pl);
A.5 Bridge Application Interface Classes

/************************************************************************
// bridge.h : main header file for the BRIDGE application
/************************************************************************

#ifndef __AFXWIN_H__
#error include 'stdafx.h' before including this file for PCH
#endif

#include "resource.h" // main symbols

// CBridgeApp:
// See bridge.cpp for the implementation of this class
///<

class CBridgeApp : public CWinApp
{
    public:
        CBridgeApp();

    // Overrides
    virtual BOOL InitInstance();

    // Implementation

    /**(AFX_MSG(CBridgeApp)
        afx_msg void OnAppAbout();
*/
// NOTE - the ClassWizard will add and remove member functions here.
// DO NOT EDIT what you see in these blocks of generated code !

//}}AFX_MSG
DECLARE_MESSAGE_MAP()

class CMainFrame : public CMDIFrameWnd
{
    DECLARE_DYNAMIC(CMainFrame)

    public:
    CMainFrame();

    // Attributes
    public:
    
    // Operations
    public:
    
    // Implementation
    public:
    virtual ~CMainFrame();

    #ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
    #endif

    protected: // control bar embedded members
    CStatusBar       m_wndStatusBar;
    CToolBar         m_wndToolBar;

    // Generated message map functions
    protected:
    //{{AFX_MSG(CMainFrame)
    afx_msg int OnCreate(LPCREATESTRUCT lpCreateStruct);
    // NOTE - the ClassWizard will add and remove member functions here.
    // DO NOT EDIT what you see in these blocks of generated code !
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()

};

Appendix A  Class Definitions for the Program
//bridgdoc.h: interface of the CBridgeDoc class

#include "struct.h"
#include "deck.h"
#include "beam.h"

class CBridgeDoc: public CDocument
{
    protected: // create from serialization only
        CBridgeDoc();
        DECLARE_DYNCREATE(CBridgeDoc)

    // Attributes
    public:
        int resultChoice;

        int loadInputCheck;
        int materialInputCheck;
        int deck1InputCheck;
        int deck2InputCheck;
        int deck3InputCheck;
        int deck4InputCheck;
        int beam1InputCheck;
        int beam2InputCheck;
        int beam3InputCheck;
        int beam4InputCheck;

        Structure myStructure;
        Deck myDeck;
        Beam myBeam;

    //-------------
        float fc;
        float fy;

        float dLoadModi;
        float rLoadModi;
        float iLoadModi;

        int numDeck;
        float deckSpan;
        float overhang;
        float h;
        float hd;
        float hc;
        float hw;
        float topCover;
float botCover;
int dDeckNum;

int numBeam;
float skewAngle;
float beamSpan[10];
float b;
float h;
float cover;
int stirSizeNum;

float posSpace;
float negSpace;

int numBar104;
int dBarNum104;
int numBar200;
int dBarNum200;

// Operations
public:

// Implementation
public:
    virtual ~CBridgeDoc();
    virtual void Serialize(CArchive& ar); // overridden for document i/o
#endif _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif
private:
    virtual BOOL OnNewDocument();

// Generated message map functions
protected:
    DECLARE_MESSAGE_MAP()
   afx_msg void OnMaterialStrength();
   afx_msg void OnLoadModifier();
   afx_msg void OnDeckDescription();
   afx_msg void OnDeckThickness();
   afx_msg void OnDeckCover();
   afx_msg void OnProjectName();
   afx_msg void OnBeamSize();
   afx_msg void OnBeamDescription();
   afx_msg void OnBeamCover();
   afx_msg void OnDeckDes();
   afx_msg void OnBeamDesignMom();
   afx_msg void OnUpdateLoadModifier(CCmdUI* pCmdUI);
   afx_msg void OnUpdateDeckDescription(CCmdUI* pCmdUI);
   afx_msg void OnUpdateDeckThickness(CCmdUI* pCmdUI);
   afx_msg void OnUpdateDeckCover(CCmdUI* pCmdUI);
afx_msg void OnUpdateBeamDescription(CCmdUI* pCmdUI);
afx_msg void OnUpdateBeamSize(CCmdUI* pCmdUI);
afx_msg void OnUpdateBeamCover(CCmdUI* pCmdUI);
afx_msg void OnUpdateDeckDesign(CCmdUI* pCmdUI);
afx_msg void OnUpdateBeamDesign(CCmdUI* pCmdUI);
afx_msg void OnBridgeResult();
afx_msg void OnUpdateBridgeResult(CCmdUI* pCmdUI);
afx_msg void OnResultDeck();
afx_msg void OnResultBeam();
afx_msg void OnUpdateResultDeck(CCmdUI* pCmdUI);
afx_msg void OnUpdateResultBeam(CCmdUI* pCmdUI);

DECLARE_MESSAGE_MAP()

class CBridgeView : public CScrollView
{
protected: // create from serialization only
  CBridgeView();
  DECLARE_DYNCREATE(CBridgeView)

  // Attributes
  public:
  CBridgeDoc* GetDocument();

  private:
  int charWidth, lineHeight;

  // Operations
  public:

  // Implementation
  public:
  virtual void OnInitialUpdate();
  virtual ~CBridgeView();
  virtual void OnDraw(CDC* pDC); // overridden to draw this view
  virtual void OnUpdate(CView* pSender, LPARAM lHint=0L, CObject* pHint=NULL);

#ifdef _DEBUG
  virtual void AssertValid() const;
  virtual void Dump(CDumpContext& dc) const;
#endif

  // Printing support

 Appendix A  Class Definitions for the Program  110
protected:
    virtual BOOL OnPreparePrinting(CPrintInfo* pInfo);
    virtual void OnBeginPrinting(CDC* pDC, CPrintInfo* pInfo);
    virtual void OnEndPrinting(CDC* pDC, CPrintInfo* pInfo);

    // Generated message map functions
    protected:
        //{{AFX_MSG(CBridgeView)
        afx_msg int OnCreate(LPCREATESTRUCT lpCreateStruct);
        //}}AFX_MSG
        DECLARE_MESSAGE_MAP()
    
#endif_DEBUG // debug version in bridgvw.cpp
inline CBridgeDoc* CBridgeView::GetDocument()
    { return (CBridgeDoc*) m_pDocument; }
#endif

/** 
    // titledlg.h : header file for project name dialog class
    //*********************************************************************/

    // CTitleDlg dialog

class CTitleDlg : public CDialog
{
    // Construction
    public:
        CTitleDlg(CWnd* pParent = NULL);  // standard constructor

    // Dialog Data
        //{{AFX_DATA(CTitleDlg)
        enum { IDD = IDD_PROJECT_NAME };  
        CString m_title;
        //}}AFX_DATA

    // Implementation
    protected:
        virtual void DoDataExchange(CDataExchange* pDX);  // DDX/DDV support

        // Generated message map functions
        //{{AFX_MSG(CTitleDlg)
        // NOTE: the ClassWizard will add member functions here
        //}}AFX_MSG
        DECLARE_MESSAGE_MAP()
};
//materialdlg.h : header file for material dialog class

/*****************************************************************

// CMaterialDlg dialog

class CMaterialDlg : public CDialog
{
    // Construction
    public:
    CMaterialDlg(CWnd* pParent = NULL); // standard constructor

    // Dialog Data
    {AFX_DATA(CMaterialDlg)
        enum { IDD = IDD_MATERIAL_STRENGTH };
        float m_fc;
        float m_fy;
    }AFX_DATA

    // Implementation
    protected:
    virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support

    // Generated message map functions
    {AFX_MSG(CMaterialDlg)
        afx_msg void OnDefaultStrength();
    }AFX_MSG
    DECLARE_MESSAGE_MAP()
};

/*****************************************************************

// loaddlg.h : header file for load modifiers dialog class

/*****************************************************************

// CLoadDlg dialog

class CLoadDlg : public CDialog
{
    // Construction
    public:
    CLoadDlg(CWnd* pParent = NULL); // standard constructor

    // Dialog Data
    {AFX_DATA(CLoadDlg)
        enum { IDD = IDD_LOAD_MODIFIER };
        float m_dLoadMod;
        float m_rLoadMod;
    }AFX_DATA

Appendix A   Class Definitions for the Program
float  m_iLoadModi;
//}}AFX_DATA

// Implementation
protected:
    virtual void DoDataExchange(CDataExchange* pDX);  // DDX/DDV support

    // Generated message map functions
   //{{AFX_MSG(CLoadDlg)
    afx_msg void OnDefaultModifier();
   //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
    
};

//******************************************************************************
// deck1dlg.h : header file for general deck description dialog class
//******************************************************************************

// CDeck1Dlg dialog

class CDeck1Dlg : public CDialog
{
    // Construction
    public:
        CDeck1Dlg(CWnd* pParent = NULL);  // standard constructor

    // Dialog Data
   //}}AFX_DATA(CDeck1Dlg)
        enum { IDD = IDD_DECK_DESCRIPTION };  
        int   m_numDeck;
        float m_deckSpan;
        float m_overhang;
   //}}AFX_DATA

    // Implementation
    protected:
        virtual void DoDataExchange(CDataExchange* pDX);  // DDX/DDV support

        // Generated message map functions
       //{{AFX_MSG(CDeck1Dlg)
        // NOTE: the ClassWizard will add member functions here
       //}}AFX_MSG
        DECLARE_MESSAGE_MAP()
    
};

Appendix A  Class Definitions for the Program 113
deck2dlg.h : header file for deck thickness dialog class

CDeck2Dlg dialog

class CDeck2Dlg : public CDialog
{
    public:
        CDeck2Dlg(CWnd* pParent = NULL); // standard constructor

    // Dialog Data
    enum { IDD = IDD_DECK_THICKNESS }; //
    float m_hi;
    float m_hd;
    float m_hc;
    float m_hw;
    float m_himin;
//}AFX_DATA

    // Implementation
    virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support

    DECLARE_MESSAGE_MAP()
};

deck3dlg.h : header file for dialog class for concrete cover and
 designation number of reinforcing bar in the deck

CDeck3Dlg dialog

class CDeck3Dlg : public CDialog
{
    public:
        CDeck3Dlg(CWnd* pParent = NULL); // standard constructor

    // Dialog Data
// {AFX_DATA(CDeck3Dlg)
enum { IDD = IDD_DECK_COVER };
float m_topCover;
float m_botCover;
int m_dDeckNum;
//}}AFX_DATA

// Implementation
protected:
virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support

// Generated message map functions
//}}AFX_MSG(CDeck3Dlg)
afx_msg void OnDefaultCover();
//}}AFX_MSG
DECLARE_MESSAGE_MAP()

/**
// beam1dlg.h : header file for beam depth and web width dialog class
/**

// CBeam1Dlg dialog

class CBeam1Dlg : public CDialog
{
// Construction
public:
    CBeam1Dlg(CWnd* pParent = NULL); // standard constructor

// Dialog Data
//}}AFX_DATA(CBeam1Dlg)
enum { IDD = IDD_BEAM_SIZE };
float m_b;
float m_h;
float m_lmin;
float m_bmin;
//}}AFX_DATA

// Implementation
protected:
virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support

// Generated message map functions
//}}AFX_MSG(CBeam1Dlg)
    // NOTE: the ClassWizard will add member functions here
//}}AFX_MSG
DECLARE_MESSAGE_MAP()

Appendix A Class Definitions for the Program 115
class CBeam2Dlg : public CDialog
{
    // Construction
    public:
    CBeam2Dlg(CWnd* pParent = NULL);  // standard constructor

    // Dialog Data
    //{{AFX_DATA(CBeam2Dlg)
    enum { IDD = IDD_BEAM_DESCRIPTION };  
    int    m_numBeam;
    float  m_skewAngle;
    float  m_beamSpan1;
    float  m_beamSpan2;
    float  m_beamSpan3;
   //}}AFX_DATA

    // Implementation
    protected:
    virtual void DoDataExchange(CDataExchange* pDX);  // DDX/DDV support

    // Generated message map functions
    //{{AFX_MSG(CBeam2Dlg)
    DECLARE_MESSAGE_MAP()
   //}}AFX_MSG

};
// Dialog Data
// {{AFX_DATA(CDeckDesDlg)
enum { IDD = IDD_DECKDESIGN };  
float m_AsDeckPos;  
float m_AsDeckNeg;  
int m_dDeckNum;  
float m_posSpace;  
float m_negSpace;  
//}}AFX_DATA

// Implementation
protected:
virtual void DoDataExchange(CDataExchange* pDX);  // DDX/DDV support

// Generated message map functions
//{{AFX_MSG(CDeckDesDlg)
// NOTE: the ClassWizard will add member functions here
//}}AFX_MSG

DECLARE_MESSAGE_MAP()

/*-------------------------------*/
// deckddlg.h : header file for dialog class for deck design
/*-------------------------------*/

// CDeckDesDlg dialog
class CDeckDesDlg : public CDialog
{
// Construction
public:
CDeckDesDlg(CWnd* pParent = NULL);  // standard constructor

// Dialog Data
//}}AFX_DATA(CDeckDesDlg)
enum { IDD = IDD_DECKDESIGN };  
float m_AsDeckPos;  
float m_AsDeckNeg;  
int m_dDeckNum;  
float m_posSpace;  
float m_negSpace;  
//}}AFX_DATA

// Implementation
protected:
virtual void DoDataExchange(CDataExchange* pDX);  // DDX/DDV support

// Generated message map functions
//{{AFX_MSG(CDeckDesDlg)
// NOTE: the ClassWizard will add member functions here
//}}AFX_MSG

Appendix A  Class Definitions for the Program  117
DECLARE_MESSAGE_MAP()

/**
   beammdlgl.h : header file for dialog class for beam design
*/

// CBeamMDlg dialog
class CBeamMDlg: public CDialog
{
   // Construction
   public:
      CBeamMDlg(CWnd* pParent = NULL); // standard constructor

   // Dialog Data
   //{{AFX_DATA(CBeamMDlg)
   enum { IDD = IDD_BEAM_DESIGN };
      float m_As104;
      float m_As200;
      int m_dBarNum104;
      int m_dBarNum200;
      int m_numBar104;
      int m_numBar200;
   }}AFX_DATA

   // Implementation
   protected:
      virtual void DoDataExchange(CDataExchange* pDX);   // DDX/DDV support

   // Generated message map functions
   //{{AFX_MSG(CBeamMDlg)
   // NOTE: the ClassWizard will add member functions here
  //}}AFX_MSG

DECLARE_MESSAGE_MAP()
Appendix B
Results for The Three Test Structures

In this Appendix the results for the three test structures discussed in Chapter 5 are presented.

B.1 Test Structure One — Three Span T-beam Bridge

B.1.1 Structure Description

The first test structure is a reinforced concrete T-beam bridge consisting of three beam spans of 10670 mm, 12800 mm and 10670 mm, five equal deck spans of 2440 mm, an overhanging cantilever having a length of 990 mm and a skew angle of 30 degrees. The compression strength of concrete, f'c, is 30 MPa and the yield strength of steel, fy, is 400 MPa. The deck is provided with a 13 mm thick concrete surface for sacrificial wear and a 75 mm thick surface of bituminous overlay for future wearing. This structure was used to compare the deck and beam design results obtained using the new AASHTO Specifications.

B.1.2 Results

General Description of the Bridge

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of decks</td>
<td>5</td>
</tr>
<tr>
<td>Deck spacing</td>
<td>2440.00 mm</td>
</tr>
<tr>
<td>Length of overhanging cantilever</td>
<td>990.00 mm</td>
</tr>
<tr>
<td>Width of bridge</td>
<td>13420.00 mm</td>
</tr>
</tbody>
</table>
### Deck Structure

- Number of beams = 3
- Beam spacing:
  - 10670.00 mm
  - 12800.00 mm
  - 10670.00 mm
- Skew angle of bridge = 30.00 deg
- Length of bridge = 34140.00 mm

### Beam Structure

- Number of design lanes = 3
- Multiple presence factor = 0.85
- Load modifier = 0.9476
- Compressive strength of concrete, $f'c =$ 30.00 MPa
- Yield strength of steel, $f_y =$ 400.00 MPa
- Modulus of concrete, $E_c =$ 27691.47 MPa
- Modulus of steel, $E_c =$ 200000.00 MPa
- Deck slab thickness, $h_i =$ 190.00 mm
- Depth of beam, $h =$ 990.00 mm
- Width of beam web, $b =$ 350.00 mm

### Deck Analysis and Design Results

- Number of decks = 5
- Deck spacing = 2440.00 mm
- Length of overhanging cantilever = 990.00 mm
- Width of bridge = 13420.00 mm
- Deck structural thickness = 190.00 mm
- Deck slab thickness = 203.00 mm
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value 1</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thickness of overhanging cantilever</td>
<td>230.00</td>
<td>mm</td>
</tr>
<tr>
<td>Thickness of future wearing surface</td>
<td>75.00</td>
<td>mm</td>
</tr>
<tr>
<td>Deck top cover to reinforcement</td>
<td>60.00</td>
<td>mm</td>
</tr>
<tr>
<td>Deck bottom cover to reinforcement</td>
<td>25.00</td>
<td>mm</td>
</tr>
<tr>
<td>Reaction at first support under DC load (N/mm)</td>
<td>18.06</td>
<td></td>
</tr>
<tr>
<td>-M at first support under DC load (N/mm)</td>
<td>-6662.76</td>
<td></td>
</tr>
<tr>
<td>+M at middle of first span under DC load (N/mm)</td>
<td>-1081.36</td>
<td></td>
</tr>
<tr>
<td>-M at second support under DC load (N/mm)</td>
<td>-1248.57</td>
<td></td>
</tr>
<tr>
<td>Reaction at first support under DW load (N/mm)</td>
<td>2.76</td>
<td></td>
</tr>
<tr>
<td>-M at first support under DW load (N/mm)</td>
<td>-307.99</td>
<td></td>
</tr>
<tr>
<td>+M at middle of first span under DW load (N/mm)</td>
<td>609.34</td>
<td></td>
</tr>
<tr>
<td>-M at second support under DW load (N/mm)</td>
<td>-972.40</td>
<td></td>
</tr>
<tr>
<td>Reaction at first support under live load (N/mm)</td>
<td>73.47</td>
<td></td>
</tr>
<tr>
<td>-M at first support under live load (N/mm)</td>
<td>-19288.67</td>
<td></td>
</tr>
<tr>
<td>+M at middle of first span under live load (N/mm)</td>
<td>19527.77</td>
<td></td>
</tr>
<tr>
<td>-M at second support under live load (N/mm)</td>
<td>-20952.61</td>
<td></td>
</tr>
<tr>
<td>Load modifier</td>
<td>0.9476</td>
<td></td>
</tr>
<tr>
<td>Maximum reaction in deck (N/mm)</td>
<td>187.36</td>
<td></td>
</tr>
<tr>
<td>Maximum positive moment in deck (N/mm)</td>
<td>42655.62</td>
<td></td>
</tr>
<tr>
<td>Maximum negative moment in deck (N/mm)</td>
<td>-50873.07</td>
<td></td>
</tr>
<tr>
<td>Designation number of bar in deck for +M</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Bar spacing for positive moment in deck (mm)</td>
<td>250.00</td>
<td></td>
</tr>
<tr>
<td>Required steel area for positive moment (mm²)</td>
<td>0.7944</td>
<td></td>
</tr>
<tr>
<td>Provided steel area for positive moment (mm²)</td>
<td>0.8042</td>
<td></td>
</tr>
<tr>
<td>Provided positive moment in deck (N/mm)</td>
<td>43629.78</td>
<td></td>
</tr>
<tr>
<td>Ductility of positive moment reinforcement</td>
<td>0.0962</td>
<td></td>
</tr>
<tr>
<td>Designation number of bar in deck for +M</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Bar spacing for negative moment in deck (mm)</td>
<td>175.00</td>
<td></td>
</tr>
<tr>
<td>Required steel area for negative moment (mm²)</td>
<td>1.1019</td>
<td></td>
</tr>
<tr>
<td>Provided steel area for negative moment (mm²)</td>
<td>1.1489</td>
<td></td>
</tr>
<tr>
<td>Provided negative moment in deck (N/mm)</td>
<td>52110.63</td>
<td></td>
</tr>
<tr>
<td>Ductility of negative moment reinforcement</td>
<td>0.1597</td>
<td></td>
</tr>
<tr>
<td>Distribution reinforcement</td>
<td>0.5388</td>
<td>mm²</td>
</tr>
<tr>
<td>Shrinkage and temperature reinforcement</td>
<td>0.1903</td>
<td>mm²</td>
</tr>
</tbody>
</table>
Deck Flexural Capacity Checks:
Ductility and capacity check for maximum positive moment = OK
Ductility and capacity check for maximum negative moment = OK
Deck Crack Control Checks:
Crack control check for maximum positive moment = OK
Crack control check for maximum negative moment = OK

Beam Analysis and Design Results

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of beams</td>
<td>3</td>
</tr>
<tr>
<td>Beam spacing</td>
<td>10670.00 mm</td>
</tr>
<tr>
<td></td>
<td>12800.00 mm</td>
</tr>
<tr>
<td></td>
<td>10670.00 mm</td>
</tr>
<tr>
<td>Skew angle of bridge</td>
<td>30.00 deg</td>
</tr>
<tr>
<td>Maximum positive service moment in beam</td>
<td>711116160.00 N*mm</td>
</tr>
<tr>
<td>Maximum negative service moment in beam</td>
<td>-802550848.00 N*mm</td>
</tr>
<tr>
<td>Maximum factored positive moment in first span</td>
<td>1079602048.00 N*mm</td>
</tr>
<tr>
<td>Maximum factored negative moment in first span</td>
<td>-1160626304.00 N*mm</td>
</tr>
<tr>
<td>Maximum factored positive moment in 2nd span</td>
<td>1015341824.00 N*mm</td>
</tr>
<tr>
<td>Maximum factored negative moment in 2nd span</td>
<td>-1160626304.00 N*mm</td>
</tr>
<tr>
<td>Factored shear at left of first span</td>
<td>633844.81 N</td>
</tr>
<tr>
<td>Factored shear at right of first span</td>
<td>-792798.44 N</td>
</tr>
<tr>
<td>Factored shear at middle of first span</td>
<td>206271.75 N</td>
</tr>
<tr>
<td>Factored shear at left of second span</td>
<td>794799.13 N</td>
</tr>
<tr>
<td>Required area of steel for positive moment</td>
<td>3703.27 mm²</td>
</tr>
<tr>
<td>Provided area of steel for positive moment</td>
<td>4003.93 mm²</td>
</tr>
<tr>
<td>Required area of steel for negative moment</td>
<td>4165.76 mm²</td>
</tr>
<tr>
<td>Provided area of steel for negative moment</td>
<td>4488.83 mm²</td>
</tr>
<tr>
<td>Provided positive moment in first span</td>
<td>1287816576.00 N*mm</td>
</tr>
<tr>
<td>Ductility for positive moment reinforcement</td>
<td>0.0414</td>
</tr>
<tr>
<td>Ratio of reinforcement for positive moment</td>
<td>0.0061</td>
</tr>
<tr>
<td>Provided negative moment in first span</td>
<td>1294092800.00 N*mm</td>
</tr>
<tr>
<td>Ductility for negative moment reinforcement</td>
<td>0.2671</td>
</tr>
<tr>
<td>Ratio of reinforcement for negative moment</td>
<td>0.0085</td>
</tr>
<tr>
<td>Provided positive moment in second span</td>
<td>1287816576.00 N*mm</td>
</tr>
<tr>
<td>Ductility for positive moment reinforcement</td>
<td>0.0414</td>
</tr>
</tbody>
</table>
Ratio of reinforcement for positive moment = 0.0061
Provided negative moment in second span = 1294092800.00 N\*mm
Ductility for negative moment reinforcement = 0.2671
Ratio of reinforcement for negative moment = 0.0085
Required shear at middle of first span = 206271.75 N
Bar size of stirrup in first span = 10
Stirrup spacing at middle of first span = 504.23 mm
Stirrup spacing at left of first span = 119.06 mm
Stirrup spacing at right of first span = 119.06 mm
Required shear at middle of second span = 722136.88 N
Bar size of stirrup in second span = 10
Stirrup spacing at middle of second span = 504.23 mm
Stirrup spacing at left of second span = 119.06 mm
Stirrup spacing at right of second span = 119.06 mm

Maximum live load deflection in first span = 2.8899 mm
Maximum dead load deflection in first span = 9.1986 mm
Flexural rigidity EI for first span = 7.297079*10^{14} mm^4

Maximum live load deflection in second span = 2.7449 mm
Maximum dead load deflection in second span = 8.3380 mm
Flexural rigidity EI for second span = 7.297079*10^{14} mm^4

Beam Flexural Capacity Checks:
Maximum positive moment in first span = OK
Maximum negative moment in first span = OK
Maximum positive moment in second span = OK
Maximum negative moment in second span = OK

Beam Shear Capacity Checks:
Shear capacity at middle of first span = OK
Shear capacity at left end of second span = OK

Beam Fatigue Checks:
Fatigue check for positive moment in first span = OK
Fatigue check for negative moment in first span = OK
Fatigue check for positive moment in second span = OK
Fatigue check for negative moment in second span = OK

Beam Cracking Control Checks:
Cracking check for positive moment in first span = OK

Appendix B  Results for The Three Test Structures
Cracking check for negative moment in first span = OK
Cracking check for positive moment in 2nd span = OK
Cracking check for negative moment in 2nd span = OK

B.2 Test Structure Two — Three Span T-beam Bridge

B.2.1 Structure Description

The second test structure is also a reinforced concrete T-beam bridge. It has three spans of 10000 mm, 12000 mm and 10000 mm for the beam, four equal spans of 2950 mm for the deck, a 1190 mm long overhanging cantilever span and a skew angle of 40 degrees. Other data are same as those for the first test structure. This structure was used to compare the deck and beam design results obtained using the new AASHTO Specifications.

B.2.2 Results

General Description of the Bridge

| Number of decks | 4 |
| Deck spacing     | 2950.00 mm |
| Length of overhanging cantilever | 1190.00 mm |
| Width of bridge  | 13420.00 mm |

Deck Structure

| Number of beams | 3 |
| Beam spacing    | 10000.00 mm |
|                 | 12000.00 mm |
|                 | 10000.00 mm |
Skew angle of bridge = 40.00 deg
Length of bridge = 32000.00 mm

Beam Structure

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of design lanes</td>
<td>3</td>
</tr>
<tr>
<td>Multiple presence factor</td>
<td>0.85</td>
</tr>
<tr>
<td>Load modifier</td>
<td>0.9476</td>
</tr>
<tr>
<td>Compressive strength of concrete, $f_c$</td>
<td>30.00  MPa</td>
</tr>
<tr>
<td>Yield strength of steel, $f_y$</td>
<td>400.00 MPa</td>
</tr>
<tr>
<td>Modulus of concrete, $E_c$</td>
<td>27691.47 MPa</td>
</tr>
<tr>
<td>Modulus of steel, $E_s$</td>
<td>200000.00 MPa</td>
</tr>
<tr>
<td>Deck slab thickness, $h_1$</td>
<td>200.00 mm</td>
</tr>
<tr>
<td>Depth of beam, $h$</td>
<td>990.00 mm</td>
</tr>
<tr>
<td>Width of beam web, $b$</td>
<td>350.00 mm</td>
</tr>
</tbody>
</table>

Deck Analysis and Design Results

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of decks</td>
<td>4</td>
</tr>
<tr>
<td>Deck spacing</td>
<td>2950.00 mm</td>
</tr>
<tr>
<td>Length of overhanging cantilever</td>
<td>1190.00 mm</td>
</tr>
<tr>
<td>Width of bridge</td>
<td>13420.00 mm</td>
</tr>
<tr>
<td>Deck structural thickness</td>
<td>200.00 mm</td>
</tr>
<tr>
<td>Deck slab thickness</td>
<td>213.00 mm</td>
</tr>
<tr>
<td>Thickness of overhanging cantilever</td>
<td>240.00 mm</td>
</tr>
<tr>
<td>Thickness of future wearing surface</td>
<td>75.00 mm</td>
</tr>
<tr>
<td>Deck top cover to reinforcement</td>
<td>60.00 mm</td>
</tr>
<tr>
<td>Deck bottom cover to reinforcement</td>
<td>25.00 mm</td>
</tr>
<tr>
<td>Reaction at first support under DC load</td>
<td>21.03 N / mm</td>
</tr>
<tr>
<td>$-M$ at first support under DC load</td>
<td>-8939.06 N*mm / mm</td>
</tr>
<tr>
<td>$+M$ at middle of first span under DC load</td>
<td>-1028.86 N*mm / mm</td>
</tr>
</tbody>
</table>
-M at second support under DC load = -2260.50 N*mm / mm

Reaction at first support under DW load = 3.49 N / mm
-M at first support under DW load = -543.07 N*mm / mm
+M at middle of first span under DW load = 844.99 N*mm / mm
-M at second support under DW load = -1396.30 N*mm / mm

Reaction at first support under live load = 80.56 N / mm
-M at first support under live load = -28354.52 N*mm / mm
+M at middle of first span under live load = 22728.57 N*mm / mm
-M at second support under live load = -22643.82 N*mm / mm

Load modifier = 0.9476
Maximum reaction in deck = 207.55 N / mm
Maximum positive moment in deck = 50112.44 N*mm / mm
Maximum negative moment in deck = -73899.17 N*mm / mm

Designation number of bar in deck for +M = 15
Bar spacing for positive moment in deck = 200.00 mm
Required steel area for positive moment = 0.8774 mm^2
Provided steel area for positive moment = 1.0053 mm^2
Provided positive moment in deck = 57585.63 N*mm / mm
Ductility of positive moment reinforcement = 0.1130

Designation number of bar in deck for +M = 15
Bar spacing for negative moment in deck = 125.00 mm
Required steel area for negative moment = 1.4902 mm^2
Provided steel area for negative moment = 1.6085 mm^2
Provided negative moment in deck = 76658.27 N*mm / mm
Ductility of negative moment reinforcement = 0.2082

Distribution reinforcement = 0.6736 mm^2
Shrinkage and temperature reinforcement = 0.1997 mm^2

Deck Flexural Capacity Checks:
Ductility and capacity check for maximum positive moment = OK
Ductility and capacity check for maximum negative moment = OK

Deck Crack Control Checks:
Crack control check for maximum positive moment = OK
Crack control check for maximum negative moment = OK
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of beams =</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Beam spacing =</td>
<td>10000.00 mm</td>
<td>mm</td>
</tr>
<tr>
<td></td>
<td>12000.00 mm</td>
<td>mm</td>
</tr>
<tr>
<td></td>
<td>10000.00 mm</td>
<td>mm</td>
</tr>
<tr>
<td>Skew angle of bridge =</td>
<td>40.00 deg</td>
<td></td>
</tr>
<tr>
<td>Maximum positive service moment in beam =</td>
<td>734430208.00 N*mm</td>
<td>N*mm</td>
</tr>
<tr>
<td>Maximum negative service moment in beam =</td>
<td>-820200448.00 N*mm</td>
<td>N*mm</td>
</tr>
<tr>
<td>Maximum factored positive moment in first span =</td>
<td>1119378816.00 N*mm</td>
<td>N*mm</td>
</tr>
<tr>
<td>Maximum factored negative moment in first span =</td>
<td>-1191798784.00 N*mm</td>
<td>N*mm</td>
</tr>
<tr>
<td>Maximum factored positive moment in 2nd span =</td>
<td>1058275904.00 N*mm</td>
<td>N*mm</td>
</tr>
<tr>
<td>Maximum factored negative moment in 2nd span =</td>
<td>-1191798784.00 N*mm</td>
<td>N*mm</td>
</tr>
<tr>
<td>Factored shear at left of first span =</td>
<td>733627.44 N</td>
<td></td>
</tr>
<tr>
<td>Factored shear at right of first span =</td>
<td>-924540.38 N</td>
<td></td>
</tr>
<tr>
<td>Factored shear at middle of first span =</td>
<td>246493.70 N</td>
<td></td>
</tr>
<tr>
<td>Factored shear at left of second span =</td>
<td>927681.38 N</td>
<td></td>
</tr>
<tr>
<td>Required area of steel for positive moment =</td>
<td>3824.68 mm^2</td>
<td>mm^2</td>
</tr>
<tr>
<td>Provided area of steel for positive moment =</td>
<td>4003.93 mm^2</td>
<td>mm^2</td>
</tr>
<tr>
<td>Required area of steel for negative moment =</td>
<td>4257.38 mm^2</td>
<td>mm^2</td>
</tr>
<tr>
<td>Provided area of steel for negative moment =</td>
<td>4488.83 mm^2</td>
<td>mm^2</td>
</tr>
<tr>
<td>Provided positive moment in first span =</td>
<td>1286525312.00 N*mm</td>
<td>N*mm</td>
</tr>
<tr>
<td>Ductility for positive moment reinforcement =</td>
<td>0.0437</td>
<td></td>
</tr>
<tr>
<td>Ratio of reinforcement for positive moment =</td>
<td>0.0062</td>
<td></td>
</tr>
<tr>
<td>Provided negative moment in first span =</td>
<td>1294092800.00 N*mm</td>
<td>N*mm</td>
</tr>
<tr>
<td>Ductility for negative moment reinforcement =</td>
<td>0.2671</td>
<td></td>
</tr>
<tr>
<td>Ratio of reinforcement for negative moment =</td>
<td>0.0086</td>
<td></td>
</tr>
<tr>
<td>Provided positive moment in second span =</td>
<td>1286525312.00 N*mm</td>
<td>N*mm</td>
</tr>
<tr>
<td>Ductility for positive moment reinforcement =</td>
<td>0.0437</td>
<td></td>
</tr>
<tr>
<td>Ratio of reinforcement for positive moment =</td>
<td>0.0062</td>
<td></td>
</tr>
<tr>
<td>Provided negative moment in second span =</td>
<td>1294092800.00 N*mm</td>
<td>N*mm</td>
</tr>
<tr>
<td>Ductility for negative moment reinforcement =</td>
<td>0.2671</td>
<td></td>
</tr>
<tr>
<td>Ratio of reinforcement for negative moment =</td>
<td>0.0086</td>
<td></td>
</tr>
<tr>
<td>Required shear at middle of first span =</td>
<td>246493.70 N</td>
<td></td>
</tr>
<tr>
<td>Bar size of stirrup in first span =</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Stirrup spacing at middle of first span =</td>
<td>504.23 mm</td>
<td></td>
</tr>
</tbody>
</table>

Appendix B  Results for The Three Test Structures  127
<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stirrup spacing at left of first span =</td>
<td>99.02</td>
<td>mm</td>
</tr>
<tr>
<td>Stirrup spacing at right of first span =</td>
<td>99.02</td>
<td>mm</td>
</tr>
<tr>
<td>Required shear at middle of second span =</td>
<td>823928.94</td>
<td>N</td>
</tr>
<tr>
<td>Bar size of stirrup in second span =</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Stirrup spacing at middle of second span =</td>
<td>504.23</td>
<td>mm</td>
</tr>
<tr>
<td>Stirrup spacing at left of second span =</td>
<td>99.02</td>
<td>mm</td>
</tr>
<tr>
<td>Stirrup spacing at right of second span =</td>
<td>99.02</td>
<td>mm</td>
</tr>
<tr>
<td>Maximum live load deflection in first span =</td>
<td>2.8733</td>
<td>mm</td>
</tr>
<tr>
<td>Maximum dead load deflection in first span =</td>
<td>8.4755</td>
<td>mm</td>
</tr>
<tr>
<td>Flexural rigidity EI for first span =</td>
<td>$6.922091 \times 10^{14}$</td>
<td>mm$^4$</td>
</tr>
<tr>
<td>Maximum live load deflection in second span =</td>
<td>2.7398</td>
<td>mm</td>
</tr>
<tr>
<td>Maximum dead load deflection in second span =</td>
<td>7.7130</td>
<td>mm</td>
</tr>
<tr>
<td>Flexural rigidity EI for second span =</td>
<td>$6.922091 \times 10^{14}$</td>
<td>mm$^4$</td>
</tr>
</tbody>
</table>

Beam Flexural Capacity Checks:
- Maximum positive moment in first span = OK
- Maximum negative moment in first span = OK
- Maximum positive moment in second span = OK
- Maximum negative moment in second span = OK

Beam Shear Capacity Checks:
- Shear capacity at middle of first span = OK
- Shear capacity at left end of second span = OK

Beam Fatigue Checks:
- Fatigue check for positive moment in first span = OK
- Fatigue check for negative moment in first span = OK
- Fatigue check for positive moment in second span = OK
- Fatigue check for negative moment in second span = OK

Beam Cracking Control Checks:
- Cracking check for positive moment in first span = OK
- Cracking check for negative moment in first span = OK
- Cracking check for positive moment in 2nd span = OK
- Cracking check for negative moment in 2nd span = OK
B.3 Test Structure Three — Five Span T-beam Bridge

B.3.1 Structure Description

The third test structure is a five span symmetrical continuous beam having T sections. The beam spans are 8300 mm, 10000 mm, 12000 mm, 10000 mm and 8300 mm and the span ratio is 1.2. For this structure, continuous beam analysis results obtained using the influence line equations developed in the study and those obtained from commercial programs were compared.

B.3.2 Results

General Description of the Bridge

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of decks</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Deck spacing</td>
<td>2440.00</td>
<td>mm</td>
</tr>
<tr>
<td>Length of overhanging cantilever</td>
<td>990.00</td>
<td>mm</td>
</tr>
<tr>
<td>Width of bridge</td>
<td>13420.00</td>
<td>mm</td>
</tr>
</tbody>
</table>

Deck Structure

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of beams</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Beam spacing</td>
<td>8300.00</td>
<td>mm</td>
</tr>
<tr>
<td></td>
<td>10000.00</td>
<td>mm</td>
</tr>
<tr>
<td></td>
<td>12000.00</td>
<td>mm</td>
</tr>
<tr>
<td>Skew angle of bridge</td>
<td>0.00</td>
<td>deg</td>
</tr>
<tr>
<td>Length of bridge</td>
<td>48600.00</td>
<td>mm</td>
</tr>
</tbody>
</table>

Beam Structure
Number of design lanes = 3
Multiple presence factor = 0.85
Load modifier = 0.9476

Compressive strength of concrete, f_c = 30.00 MPa
Yield strength of steel, f_y = 400.00 MPa
Modulus of concrete, E_c = 27691.47 MPa
Modulus of steel, E_s = 200000.00 MPa

Deck slab thickness, h_d = 190.00 mm
Depth of beam, h = 990.00 mm
Width of beam web, b = 350.00 mm

Beam Analysis and Design Results

+max M at first span due to DC load in interior beam = 93686048.0 N*mm
+max M at first span due to DC load in exterior beam = 126692408.0 N*mm
-max M at 2nd support due to DC load in interior beam = -140841616.0 N*mm
-max M at 2nd support due to DC load in exterior beam = -190461264.0 N*mm
+max M at 2nd span due to DC load in interior beam = 58756928.0 N*mm
+max M at 2nd span due to DC load in exterior beam = 79457472.0 N*mm

+max M at first span due to live load in interior beam = 384886944.0 N*mm
+max M at first span due to live load in exterior beam = 384886944.0 N*mm
-max M at 2nd support due to live load in interior beam = -310682976.0 N*mm
-max M at 2nd support due to live load in exterior beam = -310682976.0 N*mm
+max M at 2nd span due to live load in interior beam = 368918688.0 N*mm
+max M at 2nd span due to live load in exterior beam = 372511008.0 N*mm

Shear at left of 1st span due to DC load in interior beam = 58341.2 N
Shear at left of 1st span due to DC load in exterior beam = 78895.3 N
Shear at right of 1st span due to DC load in interior beam = -92279.0 N
Shear at right of 1st span due to DC load in exterior beam = -124789.6 N
Shear at left of 2nd span due to DC load in interior beam = 85026.6 N
Shear at left of 2nd span due to DC load in exterior beam = 114982.2 N

Shear at left of 1st span due to live load in interior beam = 249512.9 N
Shear at left of 1st span due to live load in exterior beam = 230331.5 N
Shear at right of 1st span due to live load in interior beam = -289330.5 N

Appendix B  Results for The Three Test Structures
Shear at right of 1st span due to live load in exterior beam = -267088.2 N
Shear at left of 2nd span due to live load in interior beam = 300365.8 N
Shear at left of 2nd span due to live load in exterior beam = 277275.1 N
## Appendix C

### Comparison of Influence Line Coordinates With RISA-2D

Table C.1  Comparison of influence line coordinates for moment in three span beam

<table>
<thead>
<tr>
<th>Position</th>
<th>M104 Interpolating functions</th>
<th>RISA-2D</th>
<th>M200 Interpolating functions</th>
<th>RISA-2D</th>
<th>M205 Interpolating functions</th>
<th>RISA-2D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.0494</td>
<td>0.0494</td>
<td>-0.0264</td>
<td>-0.0264</td>
<td>-0.0099</td>
<td>-0.0099</td>
</tr>
<tr>
<td>0.2</td>
<td>0.0995</td>
<td>0.0995</td>
<td>-0.0512</td>
<td>-0.0512</td>
<td>-0.0192</td>
<td>-0.0192</td>
</tr>
<tr>
<td>0.3</td>
<td>0.1509</td>
<td>0.1509</td>
<td>-0.0728</td>
<td>-0.0728</td>
<td>-0.0273</td>
<td>-0.0273</td>
</tr>
<tr>
<td>0.4</td>
<td>0.2042</td>
<td>0.2042</td>
<td>-0.0896</td>
<td>-0.0896</td>
<td>-0.0336</td>
<td>-0.0336</td>
</tr>
<tr>
<td>0.5</td>
<td>0.1600</td>
<td>0.1600</td>
<td>-0.1000</td>
<td>-0.1000</td>
<td>-0.0375</td>
<td>-0.0375</td>
</tr>
<tr>
<td>0.6</td>
<td>0.1190</td>
<td>0.1190</td>
<td>-0.1024</td>
<td>-0.1024</td>
<td>-0.0384</td>
<td>-0.0384</td>
</tr>
<tr>
<td>0.7</td>
<td>0.0819</td>
<td>0.0819</td>
<td>-0.0952</td>
<td>-0.0952</td>
<td>-0.0357</td>
<td>-0.0357</td>
</tr>
<tr>
<td>0.8</td>
<td>0.0493</td>
<td>0.0493</td>
<td>-0.0768</td>
<td>-0.0768</td>
<td>-0.0288</td>
<td>-0.0288</td>
</tr>
<tr>
<td>0.9</td>
<td>0.0218</td>
<td>0.0218</td>
<td>-0.0456</td>
<td>-0.0456</td>
<td>-0.0171</td>
<td>-0.0171</td>
</tr>
<tr>
<td>1.1</td>
<td>-0.0156</td>
<td>-0.0156</td>
<td>-0.0390</td>
<td>-0.0390</td>
<td>0.0230</td>
<td>0.0230</td>
</tr>
<tr>
<td>1.2</td>
<td>-0.0256</td>
<td>-0.0256</td>
<td>-0.0640</td>
<td>-0.0640</td>
<td>0.0520</td>
<td>0.0520</td>
</tr>
<tr>
<td>1.3</td>
<td>-0.0308</td>
<td>-0.0308</td>
<td>-0.0770</td>
<td>-0.0770</td>
<td>0.0870</td>
<td>0.0870</td>
</tr>
<tr>
<td>1.4</td>
<td>-0.0320</td>
<td>-0.0320</td>
<td>-0.0800</td>
<td>-0.0800</td>
<td>0.1280</td>
<td>0.1280</td>
</tr>
<tr>
<td>1.5</td>
<td>-0.0300</td>
<td>-0.0300</td>
<td>-0.0750</td>
<td>-0.0750</td>
<td>0.1750</td>
<td>0.1750</td>
</tr>
<tr>
<td>1.6</td>
<td>-0.0256</td>
<td>-0.0256</td>
<td>-0.0640</td>
<td>-0.0640</td>
<td>0.1280</td>
<td>0.1280</td>
</tr>
<tr>
<td>1.7</td>
<td>-0.0196</td>
<td>-0.0196</td>
<td>-0.0490</td>
<td>-0.0490</td>
<td>0.0870</td>
<td>0.0870</td>
</tr>
<tr>
<td>1.8</td>
<td>-0.0128</td>
<td>-0.0128</td>
<td>-0.0320</td>
<td>-0.0320</td>
<td>0.0520</td>
<td>0.0520</td>
</tr>
<tr>
<td>1.9</td>
<td>-0.0060</td>
<td>-0.0060</td>
<td>-0.0150</td>
<td>-0.0150</td>
<td>0.0230</td>
<td>0.0230</td>
</tr>
<tr>
<td>2.1</td>
<td>0.0046</td>
<td>0.0046</td>
<td>0.0114</td>
<td>0.0114</td>
<td>-0.0171</td>
<td>-0.0171</td>
</tr>
<tr>
<td>2.2</td>
<td>0.0077</td>
<td>0.0077</td>
<td>0.0192</td>
<td>0.0192</td>
<td>-0.0288</td>
<td>-0.0288</td>
</tr>
<tr>
<td>2.3</td>
<td>0.0095</td>
<td>0.0095</td>
<td>0.0238</td>
<td>0.0238</td>
<td>-0.0357</td>
<td>-0.0357</td>
</tr>
<tr>
<td>2.4</td>
<td>0.0102</td>
<td>0.0102</td>
<td>0.0256</td>
<td>0.0256</td>
<td>-0.0384</td>
<td>-0.0384</td>
</tr>
<tr>
<td>2.5</td>
<td>0.0100</td>
<td>0.0100</td>
<td>0.0250</td>
<td>0.0250</td>
<td>-0.0375</td>
<td>-0.0375</td>
</tr>
<tr>
<td>2.6</td>
<td>0.0090</td>
<td>0.0090</td>
<td>0.0224</td>
<td>0.0224</td>
<td>-0.0336</td>
<td>-0.0336</td>
</tr>
<tr>
<td>2.7</td>
<td>0.0073</td>
<td>0.0073</td>
<td>0.0182</td>
<td>0.0182</td>
<td>-0.0273</td>
<td>-0.0273</td>
</tr>
<tr>
<td>2.8</td>
<td>0.0051</td>
<td>0.0051</td>
<td>0.0128</td>
<td>0.0128</td>
<td>-0.0192</td>
<td>-0.0192</td>
</tr>
<tr>
<td>2.9</td>
<td>0.0026</td>
<td>0.0026</td>
<td>0.0066</td>
<td>0.0066</td>
<td>-0.0099</td>
<td>-0.0099</td>
</tr>
<tr>
<td>Position</td>
<td>Interpolating functions</td>
<td>V100</td>
<td>Interpolating functions</td>
<td>V110</td>
<td>Interpolating functions</td>
<td>V200</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------</td>
<td>------</td>
<td>-------------------------</td>
<td>------</td>
<td>-------------------------</td>
<td>------</td>
</tr>
<tr>
<td></td>
<td>RISA-2D</td>
<td></td>
<td>RISA-2D</td>
<td></td>
<td>RISA-2D</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>0.8736</td>
<td>0.8736</td>
<td>-0.1264</td>
<td>-0.1264</td>
<td>0.0330</td>
<td>0.0330</td>
</tr>
<tr>
<td>0.2</td>
<td>0.7488</td>
<td>0.7488</td>
<td>-0.2512</td>
<td>-0.2512</td>
<td>0.0640</td>
<td>0.0640</td>
</tr>
<tr>
<td>0.3</td>
<td>0.6272</td>
<td>0.6272</td>
<td>-0.3728</td>
<td>-0.3728</td>
<td>0.0910</td>
<td>0.0910</td>
</tr>
<tr>
<td>0.4</td>
<td>0.5104</td>
<td>0.5104</td>
<td>-0.4896</td>
<td>-0.4896</td>
<td>0.1120</td>
<td>0.1120</td>
</tr>
<tr>
<td>0.5</td>
<td>0.4000</td>
<td>0.4000</td>
<td>-0.6000</td>
<td>-0.6000</td>
<td>0.1250</td>
<td>0.1250</td>
</tr>
<tr>
<td>0.6</td>
<td>0.2976</td>
<td>0.2976</td>
<td>-0.7024</td>
<td>-0.7024</td>
<td>0.1280</td>
<td>0.1280</td>
</tr>
<tr>
<td>0.7</td>
<td>0.2048</td>
<td>0.2048</td>
<td>-0.7952</td>
<td>-0.7952</td>
<td>0.1190</td>
<td>0.1190</td>
</tr>
<tr>
<td>0.8</td>
<td>0.1232</td>
<td>0.1232</td>
<td>-0.8768</td>
<td>-0.8768</td>
<td>0.0960</td>
<td>0.0960</td>
</tr>
<tr>
<td>0.9</td>
<td>0.0544</td>
<td>0.0544</td>
<td>-0.9456</td>
<td>-0.9456</td>
<td>0.0570</td>
<td>0.0570</td>
</tr>
<tr>
<td>1.1</td>
<td>-0.0390</td>
<td>-0.0390</td>
<td>-0.0390</td>
<td>-0.0390</td>
<td>0.9240</td>
<td>0.9240</td>
</tr>
<tr>
<td>1.2</td>
<td>-0.0640</td>
<td>-0.0640</td>
<td>-0.0640</td>
<td>-0.0640</td>
<td>0.8320</td>
<td>0.8320</td>
</tr>
<tr>
<td>1.3</td>
<td>-0.0770</td>
<td>-0.0770</td>
<td>-0.0770</td>
<td>-0.0770</td>
<td>0.7280</td>
<td>0.7280</td>
</tr>
<tr>
<td>1.4</td>
<td>-0.0800</td>
<td>-0.0800</td>
<td>-0.0800</td>
<td>-0.0800</td>
<td>0.6160</td>
<td>0.6160</td>
</tr>
<tr>
<td>1.5</td>
<td>-0.0750</td>
<td>-0.0750</td>
<td>-0.0750</td>
<td>-0.0750</td>
<td>0.5000</td>
<td>0.5000</td>
</tr>
<tr>
<td>1.6</td>
<td>-0.0640</td>
<td>-0.0640</td>
<td>-0.0640</td>
<td>-0.0640</td>
<td>0.3840</td>
<td>0.3840</td>
</tr>
<tr>
<td>1.7</td>
<td>-0.0490</td>
<td>-0.0490</td>
<td>-0.0490</td>
<td>-0.0490</td>
<td>0.2720</td>
<td>0.2720</td>
</tr>
<tr>
<td>1.8</td>
<td>-0.0320</td>
<td>-0.0320</td>
<td>-0.0320</td>
<td>-0.0320</td>
<td>0.1680</td>
<td>0.1680</td>
</tr>
<tr>
<td>1.9</td>
<td>-0.0150</td>
<td>-0.0150</td>
<td>-0.0150</td>
<td>-0.0150</td>
<td>0.0760</td>
<td>0.0760</td>
</tr>
<tr>
<td>2.1</td>
<td>0.0114</td>
<td>0.0114</td>
<td>0.0114</td>
<td>0.0114</td>
<td>-0.0570</td>
<td>-0.0570</td>
</tr>
<tr>
<td>2.2</td>
<td>0.0192</td>
<td>0.0192</td>
<td>0.0192</td>
<td>0.0192</td>
<td>-0.0960</td>
<td>-0.0960</td>
</tr>
<tr>
<td>2.3</td>
<td>0.0238</td>
<td>0.0238</td>
<td>0.0238</td>
<td>0.0238</td>
<td>-0.1190</td>
<td>-0.1190</td>
</tr>
<tr>
<td>2.4</td>
<td>0.0256</td>
<td>0.0256</td>
<td>0.0256</td>
<td>0.0256</td>
<td>-0.1280</td>
<td>-0.1280</td>
</tr>
<tr>
<td>2.5</td>
<td>0.0250</td>
<td>0.0250</td>
<td>0.0250</td>
<td>0.0250</td>
<td>-0.1250</td>
<td>-0.1250</td>
</tr>
<tr>
<td>2.6</td>
<td>0.0224</td>
<td>0.0224</td>
<td>0.0224</td>
<td>0.0224</td>
<td>-0.1120</td>
<td>-0.1120</td>
</tr>
<tr>
<td>2.7</td>
<td>0.0182</td>
<td>0.0182</td>
<td>0.0182</td>
<td>0.0182</td>
<td>-0.0910</td>
<td>-0.0910</td>
</tr>
<tr>
<td>2.8</td>
<td>0.0128</td>
<td>0.0128</td>
<td>0.0128</td>
<td>0.0128</td>
<td>-0.0640</td>
<td>-0.0640</td>
</tr>
<tr>
<td>2.9</td>
<td>0.0066</td>
<td>0.0066</td>
<td>0.0066</td>
<td>0.0066</td>
<td>-0.0330</td>
<td>-0.0330</td>
</tr>
</tbody>
</table>

Table C.2  Comparison of influence line coordinates for shear in three span beam
<table>
<thead>
<tr>
<th>Position</th>
<th>(\text{Interpolating functions} )</th>
<th>(\text{RISA-2D} )</th>
<th>(\text{Percent Difference} )</th>
<th>(\text{Interpolating functions} )</th>
<th>(\text{RISA-2D} )</th>
<th>(\text{Percent Difference} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>-0.1264</td>
<td>-0.1264</td>
<td>0.0</td>
<td>0.0330</td>
<td>0.0330</td>
<td>0.0</td>
</tr>
<tr>
<td>0.2</td>
<td>-0.2512</td>
<td>-0.2512</td>
<td>0.0</td>
<td>0.0640</td>
<td>0.0640</td>
<td>0.0</td>
</tr>
<tr>
<td>0.3</td>
<td>-0.3728</td>
<td>-0.3728</td>
<td>0.0</td>
<td>0.0910</td>
<td>0.0910</td>
<td>0.0</td>
</tr>
<tr>
<td>0.4</td>
<td>-0.5104</td>
<td>-0.5104</td>
<td>0.0</td>
<td>0.1120</td>
<td>0.1120</td>
<td>0.0</td>
</tr>
<tr>
<td>0.5</td>
<td>0.3935</td>
<td>0.4000</td>
<td>1.63</td>
<td>0.1250</td>
<td>0.1250</td>
<td>0.0</td>
</tr>
<tr>
<td>0.6</td>
<td>0.2976</td>
<td>0.2976</td>
<td>0.0</td>
<td>0.1280</td>
<td>0.1280</td>
<td>0.0</td>
</tr>
<tr>
<td>0.7</td>
<td>0.2061</td>
<td>0.2048</td>
<td>0.63</td>
<td>0.1190</td>
<td>0.1190</td>
<td>0.0</td>
</tr>
<tr>
<td>0.8</td>
<td>0.1232</td>
<td>0.1232</td>
<td>0.0</td>
<td>0.0960</td>
<td>0.0960</td>
<td>0.0</td>
</tr>
<tr>
<td>0.9</td>
<td>0.0531</td>
<td>0.0544</td>
<td>2.39</td>
<td>0.0570</td>
<td>0.0570</td>
<td>0.0</td>
</tr>
<tr>
<td>1.1</td>
<td>-0.0390</td>
<td>-0.0390</td>
<td>0.0</td>
<td>-0.0813</td>
<td>-0.0760</td>
<td>6.97</td>
</tr>
<tr>
<td>1.2</td>
<td>-0.0640</td>
<td>-0.0640</td>
<td>0.0</td>
<td>-0.1720</td>
<td>-0.1680</td>
<td>2.38</td>
</tr>
<tr>
<td>1.3</td>
<td>-0.0770</td>
<td>-0.0770</td>
<td>0.0</td>
<td>-0.2720</td>
<td>-0.2720</td>
<td>0.0</td>
</tr>
<tr>
<td>1.4</td>
<td>-0.0800</td>
<td>-0.0800</td>
<td>0.0</td>
<td>-0.3813</td>
<td>-0.3840</td>
<td>0.7</td>
</tr>
<tr>
<td>1.5</td>
<td>-0.0750</td>
<td>-0.0750</td>
<td>0.0</td>
<td>-0.5000</td>
<td>-0.5000</td>
<td>0.0</td>
</tr>
<tr>
<td>1.6</td>
<td>-0.0640</td>
<td>-0.0640</td>
<td>0.0</td>
<td>0.3813</td>
<td>0.3840</td>
<td>0.7</td>
</tr>
<tr>
<td>1.7</td>
<td>-0.0490</td>
<td>-0.0490</td>
<td>0.0</td>
<td>0.2720</td>
<td>0.2720</td>
<td>0.0</td>
</tr>
<tr>
<td>1.8</td>
<td>-0.0320</td>
<td>-0.0320</td>
<td>0.0</td>
<td>0.1720</td>
<td>0.1680</td>
<td>2.38</td>
</tr>
<tr>
<td>1.9</td>
<td>-0.0150</td>
<td>-0.0150</td>
<td>0.0</td>
<td>0.0813</td>
<td>0.0760</td>
<td>6.97</td>
</tr>
<tr>
<td>2.1</td>
<td>0.0114</td>
<td>0.0114</td>
<td>0.0</td>
<td>-0.0570</td>
<td>-0.0570</td>
<td>0.0</td>
</tr>
<tr>
<td>2.2</td>
<td>0.0192</td>
<td>0.0192</td>
<td>0.0</td>
<td>-0.0960</td>
<td>-0.0960</td>
<td>0.0</td>
</tr>
<tr>
<td>2.3</td>
<td>0.0238</td>
<td>0.0238</td>
<td>0.0</td>
<td>-0.1190</td>
<td>-0.1190</td>
<td>0.0</td>
</tr>
<tr>
<td>2.4</td>
<td>0.0256</td>
<td>0.0256</td>
<td>0.0</td>
<td>-0.1280</td>
<td>-0.1280</td>
<td>0.0</td>
</tr>
<tr>
<td>2.5</td>
<td>0.0250</td>
<td>0.0250</td>
<td>0.0</td>
<td>-0.1250</td>
<td>-0.1250</td>
<td>0.0</td>
</tr>
<tr>
<td>2.6</td>
<td>0.0224</td>
<td>0.0224</td>
<td>0.0</td>
<td>-0.1120</td>
<td>-0.1120</td>
<td>0.0</td>
</tr>
<tr>
<td>2.7</td>
<td>0.0182</td>
<td>0.0182</td>
<td>0.0</td>
<td>-0.0910</td>
<td>-0.0910</td>
<td>0.0</td>
</tr>
<tr>
<td>2.8</td>
<td>0.0128</td>
<td>0.0128</td>
<td>0.0</td>
<td>-0.0640</td>
<td>-0.0640</td>
<td>0.0</td>
</tr>
<tr>
<td>2.9</td>
<td>0.0066</td>
<td>0.0066</td>
<td>0.0</td>
<td>-0.0330</td>
<td>-0.0330</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Appendix C  Comparison of Influence Line Coordinates with RISA-2D  134
Table C.3  Comparison of influence line coordinates for moment in five span beam

<table>
<thead>
<tr>
<th>Position</th>
<th>M104 Interpolating functions</th>
<th>RISA-2D</th>
<th>M200 Interpolating functions</th>
<th>RISA-2D</th>
<th>M205 Interpolating functions</th>
<th>RISA-2D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.0494</td>
<td>0.0494</td>
<td>-0.0265</td>
<td>-0.0265</td>
<td>-0.0097</td>
<td>-0.0097</td>
</tr>
<tr>
<td>0.2</td>
<td>0.0994</td>
<td>0.0994</td>
<td>-0.0515</td>
<td>-0.0515</td>
<td>-0.0188</td>
<td>-0.0188</td>
</tr>
<tr>
<td>0.3</td>
<td>0.1507</td>
<td>0.1507</td>
<td>-0.0732</td>
<td>-0.0732</td>
<td>-0.0268</td>
<td>-0.0268</td>
</tr>
<tr>
<td>0.4</td>
<td>0.2040</td>
<td>0.2040</td>
<td>-0.0900</td>
<td>-0.0900</td>
<td>-0.0330</td>
<td>-0.0330</td>
</tr>
<tr>
<td>0.5</td>
<td>0.1598</td>
<td>0.1598</td>
<td>-0.1005</td>
<td>-0.1005</td>
<td>-0.0368</td>
<td>-0.0368</td>
</tr>
<tr>
<td>0.6</td>
<td>0.1188</td>
<td>0.1188</td>
<td>-0.1029</td>
<td>-0.1029</td>
<td>-0.0377</td>
<td>-0.0377</td>
</tr>
<tr>
<td>0.7</td>
<td>0.0817</td>
<td>0.0817</td>
<td>-0.0957</td>
<td>-0.0957</td>
<td>-0.0350</td>
<td>-0.0350</td>
</tr>
<tr>
<td>0.8</td>
<td>0.0491</td>
<td>0.0491</td>
<td>-0.0772</td>
<td>-0.0772</td>
<td>-0.0283</td>
<td>-0.0283</td>
</tr>
<tr>
<td>0.9</td>
<td>0.0217</td>
<td>0.0217</td>
<td>-0.0458</td>
<td>-0.0458</td>
<td>-0.0168</td>
<td>-0.0168</td>
</tr>
<tr>
<td>1.1</td>
<td>-0.0155</td>
<td>-0.0155</td>
<td>-0.0387</td>
<td>-0.0387</td>
<td>0.0226</td>
<td>0.0226</td>
</tr>
<tr>
<td>1.2</td>
<td>-0.0254</td>
<td>-0.0254</td>
<td>-0.0634</td>
<td>-0.0634</td>
<td>0.0511</td>
<td>0.0511</td>
</tr>
<tr>
<td>1.3</td>
<td>-0.0304</td>
<td>-0.0304</td>
<td>-0.0761</td>
<td>-0.0761</td>
<td>0.0856</td>
<td>0.0856</td>
</tr>
<tr>
<td>1.4</td>
<td>-0.0315</td>
<td>-0.0315</td>
<td>-0.0788</td>
<td>-0.0788</td>
<td>0.1262</td>
<td>0.1262</td>
</tr>
<tr>
<td>1.5</td>
<td>-0.0294</td>
<td>-0.0294</td>
<td>-0.0736</td>
<td>-0.0736</td>
<td>0.1729</td>
<td>0.1729</td>
</tr>
<tr>
<td>1.6</td>
<td>-0.0250</td>
<td>-0.0250</td>
<td>-0.0625</td>
<td>-0.0625</td>
<td>0.1257</td>
<td>0.1257</td>
</tr>
<tr>
<td>1.7</td>
<td>-0.0190</td>
<td>-0.0190</td>
<td>-0.0475</td>
<td>-0.0475</td>
<td>0.0848</td>
<td>0.0848</td>
</tr>
<tr>
<td>1.8</td>
<td>-0.0123</td>
<td>-0.0123</td>
<td>-0.0308</td>
<td>-0.0308</td>
<td>0.0502</td>
<td>0.0502</td>
</tr>
<tr>
<td>1.9</td>
<td>-0.0057</td>
<td>-0.0057</td>
<td>-0.0145</td>
<td>-0.0145</td>
<td>0.0220</td>
<td>0.0220</td>
</tr>
<tr>
<td>2.0</td>
<td>0.0042</td>
<td>0.0042</td>
<td>0.0104</td>
<td>0.0104</td>
<td>-0.0156</td>
<td>-0.0156</td>
</tr>
<tr>
<td>2.1</td>
<td>0.0068</td>
<td>0.0068</td>
<td>0.0170</td>
<td>0.0170</td>
<td>-0.0255</td>
<td>-0.0255</td>
</tr>
<tr>
<td>2.2</td>
<td>0.0082</td>
<td>0.0082</td>
<td>0.0204</td>
<td>0.0204</td>
<td>-0.0306</td>
<td>-0.0306</td>
</tr>
<tr>
<td>2.3</td>
<td>0.0085</td>
<td>0.0085</td>
<td>0.0211</td>
<td>0.0211</td>
<td>-0.0317</td>
<td>-0.0317</td>
</tr>
<tr>
<td>2.4</td>
<td>0.0079</td>
<td>0.0079</td>
<td>0.0197</td>
<td>0.0197</td>
<td>-0.0296</td>
<td>-0.0296</td>
</tr>
<tr>
<td>2.5</td>
<td>0.0067</td>
<td>0.0067</td>
<td>0.0168</td>
<td>0.0168</td>
<td>-0.0252</td>
<td>-0.0252</td>
</tr>
<tr>
<td>2.6</td>
<td>0.0051</td>
<td>0.0051</td>
<td>0.0128</td>
<td>0.0128</td>
<td>-0.0191</td>
<td>-0.0191</td>
</tr>
<tr>
<td>2.7</td>
<td>0.0033</td>
<td>0.0033</td>
<td>0.0083</td>
<td>0.0083</td>
<td>-0.0124</td>
<td>-0.0124</td>
</tr>
<tr>
<td>2.8</td>
<td>0.0015</td>
<td>0.0015</td>
<td>0.0038</td>
<td>0.0038</td>
<td>-0.0057</td>
<td>-0.0057</td>
</tr>
</tbody>
</table>
Table C.4  Comparison of influence line coordinates for shear in five span beam

<table>
<thead>
<tr>
<th>Position</th>
<th>V100</th>
<th>V110</th>
<th>V200</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Interpolating functions</td>
<td>RISA-2D Interpolating functions</td>
<td>RISA-2D Interpolating functions</td>
</tr>
<tr>
<td>0.1</td>
<td>0.8735</td>
<td>0.8735</td>
<td>-0.1265</td>
</tr>
<tr>
<td>0.2</td>
<td>0.7486</td>
<td>0.7486</td>
<td>-0.2515</td>
</tr>
<tr>
<td>0.3</td>
<td>0.6269</td>
<td>0.6269</td>
<td>-0.3732</td>
</tr>
<tr>
<td>0.4</td>
<td>0.5100</td>
<td>0.5100</td>
<td>-0.4900</td>
</tr>
<tr>
<td>0.5</td>
<td>0.3995</td>
<td>0.3995</td>
<td>-0.6005</td>
</tr>
<tr>
<td>0.6</td>
<td>0.2971</td>
<td>0.2971</td>
<td>-0.7029</td>
</tr>
<tr>
<td>0.7</td>
<td>0.2044</td>
<td>0.2044</td>
<td>-0.7957</td>
</tr>
<tr>
<td>0.8</td>
<td>0.1228</td>
<td>0.1228</td>
<td>-0.8772</td>
</tr>
<tr>
<td>0.9</td>
<td>0.0542</td>
<td>0.0542</td>
<td>-0.9458</td>
</tr>
<tr>
<td>1.1</td>
<td>-0.0387</td>
<td>-0.0387</td>
<td>-0.0387</td>
</tr>
<tr>
<td>1.2</td>
<td>-0.0634</td>
<td>-0.0634</td>
<td>-0.0634</td>
</tr>
<tr>
<td>1.3</td>
<td>-0.0761</td>
<td>-0.0761</td>
<td>-0.0761</td>
</tr>
<tr>
<td>1.4</td>
<td>-0.0788</td>
<td>-0.0788</td>
<td>-0.0788</td>
</tr>
<tr>
<td>1.5</td>
<td>-0.0736</td>
<td>-0.0736</td>
<td>-0.0736</td>
</tr>
<tr>
<td>1.6</td>
<td>-0.0625</td>
<td>-0.0625</td>
<td>-0.0625</td>
</tr>
<tr>
<td>1.7</td>
<td>-0.0475</td>
<td>-0.0475</td>
<td>-0.0475</td>
</tr>
<tr>
<td>1.8</td>
<td>-0.0308</td>
<td>-0.0308</td>
<td>-0.0308</td>
</tr>
<tr>
<td>1.9</td>
<td>-0.0143</td>
<td>-0.0143</td>
<td>-0.0143</td>
</tr>
<tr>
<td>2.1</td>
<td>0.0104</td>
<td>0.0104</td>
<td>0.0104</td>
</tr>
<tr>
<td>2.2</td>
<td>0.0170</td>
<td>0.0170</td>
<td>0.0170</td>
</tr>
<tr>
<td>2.3</td>
<td>0.0204</td>
<td>0.0204</td>
<td>0.0204</td>
</tr>
<tr>
<td>2.4</td>
<td>0.0211</td>
<td>0.0211</td>
<td>0.0211</td>
</tr>
<tr>
<td>2.5</td>
<td>0.0197</td>
<td>0.0197</td>
<td>0.0197</td>
</tr>
<tr>
<td>2.6</td>
<td>0.0168</td>
<td>0.0168</td>
<td>0.0168</td>
</tr>
<tr>
<td>2.7</td>
<td>0.0128</td>
<td>0.0128</td>
<td>0.0128</td>
</tr>
<tr>
<td>2.8</td>
<td>0.0083</td>
<td>0.0083</td>
<td>0.0083</td>
</tr>
<tr>
<td>2.9</td>
<td>0.0038</td>
<td>0.0038</td>
<td>0.0038</td>
</tr>
</tbody>
</table>

Appendix C  Comparison of Influence Line Coordinates with RISA-2D 136
### Table C.4 (continued)

<table>
<thead>
<tr>
<th>Position</th>
<th>Interpolating functions</th>
<th>RISA-2D</th>
<th>Percent Difference</th>
<th>Interpolating functions</th>
<th>RISA-2D</th>
<th>Percent Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>-0.1265</td>
<td>-0.1265</td>
<td>0.0</td>
<td>0.0336</td>
<td>0.0336</td>
<td>0.0</td>
</tr>
<tr>
<td>0.2</td>
<td>-0.2515</td>
<td>-0.2515</td>
<td>0.0</td>
<td>0.0652</td>
<td>0.0652</td>
<td>0.0</td>
</tr>
<tr>
<td>0.3</td>
<td>-0.3732</td>
<td>-0.3732</td>
<td>0.0</td>
<td>0.0927</td>
<td>0.0927</td>
<td>0.0</td>
</tr>
<tr>
<td>0.4</td>
<td>-0.5100</td>
<td>-0.5100</td>
<td>0.0</td>
<td>0.1141</td>
<td>0.1141</td>
<td>0.0</td>
</tr>
<tr>
<td>0.5</td>
<td>0.3938</td>
<td>0.3995</td>
<td>1.43</td>
<td>0.1274</td>
<td>0.1274</td>
<td>0.0</td>
</tr>
<tr>
<td>0.6</td>
<td>0.2971</td>
<td>0.2971</td>
<td>0.0</td>
<td>0.1305</td>
<td>0.1305</td>
<td>0.0</td>
</tr>
<tr>
<td>0.7</td>
<td>0.2047</td>
<td>0.2044</td>
<td>0.147</td>
<td>0.1219</td>
<td>0.1219</td>
<td>0.0</td>
</tr>
<tr>
<td>0.8</td>
<td>0.1213</td>
<td>0.1213</td>
<td>0.0</td>
<td>0.0978</td>
<td>0.0978</td>
<td>0.0</td>
</tr>
<tr>
<td>0.9</td>
<td>0.0542</td>
<td>0.0542</td>
<td>0.0</td>
<td>0.0581</td>
<td>0.0581</td>
<td>0.0</td>
</tr>
<tr>
<td>1.1</td>
<td>-0.0387</td>
<td>-0.0387</td>
<td>0.0</td>
<td>-0.0830</td>
<td>-0.0775</td>
<td>7.10</td>
</tr>
<tr>
<td>1.2</td>
<td>-0.0634</td>
<td>-0.0634</td>
<td>0.0</td>
<td>-0.1752</td>
<td>-0.1710</td>
<td>2.45</td>
</tr>
<tr>
<td>1.3</td>
<td>-0.0761</td>
<td>-0.0761</td>
<td>0.0</td>
<td>-0.2767</td>
<td>-0.2767</td>
<td>0.0</td>
</tr>
<tr>
<td>1.4</td>
<td>-0.0788</td>
<td>-0.0788</td>
<td>0.0</td>
<td>-0.3873</td>
<td>-0.3901</td>
<td>0.72</td>
</tr>
<tr>
<td>1.5</td>
<td>-0.0736</td>
<td>-0.0736</td>
<td>0.0</td>
<td>-0.4928</td>
<td>-0.4928</td>
<td>0.0</td>
</tr>
<tr>
<td>1.6</td>
<td>-0.0625</td>
<td>-0.0625</td>
<td>0.0</td>
<td>-0.3736</td>
<td>-0.3763</td>
<td>0.72</td>
</tr>
<tr>
<td>1.7</td>
<td>-0.0475</td>
<td>-0.0475</td>
<td>0.0</td>
<td>-0.2646</td>
<td>-0.2646</td>
<td>0.0</td>
</tr>
<tr>
<td>1.8</td>
<td>-0.0308</td>
<td>-0.0308</td>
<td>0.0</td>
<td>-0.1661</td>
<td>-0.1619</td>
<td>2.59</td>
</tr>
<tr>
<td>1.9</td>
<td>-0.0143</td>
<td>-0.0143</td>
<td>0.0</td>
<td>-0.0778</td>
<td>-0.0723</td>
<td>7.61</td>
</tr>
<tr>
<td>2.1</td>
<td>0.0104</td>
<td>0.0104</td>
<td>0.0</td>
<td>-0.0519</td>
<td>-0.0519</td>
<td>0.0</td>
</tr>
<tr>
<td>2.2</td>
<td>0.0170</td>
<td>0.0170</td>
<td>0.0</td>
<td>-0.0850</td>
<td>-0.0850</td>
<td>0.0</td>
</tr>
<tr>
<td>2.3</td>
<td>0.0204</td>
<td>0.0204</td>
<td>0.0</td>
<td>-0.1020</td>
<td>-0.1020</td>
<td>0.0</td>
</tr>
<tr>
<td>2.4</td>
<td>0.0211</td>
<td>0.0211</td>
<td>0.0</td>
<td>-0.1057</td>
<td>-0.1057</td>
<td>0.0</td>
</tr>
<tr>
<td>2.5</td>
<td>0.0197</td>
<td>0.0197</td>
<td>0.0</td>
<td>-0.0987</td>
<td>-0.0987</td>
<td>0.0</td>
</tr>
<tr>
<td>2.6</td>
<td>0.0168</td>
<td>0.0168</td>
<td>0.0</td>
<td>-0.0838</td>
<td>-0.0838</td>
<td>0.0</td>
</tr>
<tr>
<td>2.7</td>
<td>0.0128</td>
<td>0.0128</td>
<td>0.0</td>
<td>-0.0638</td>
<td>-0.0638</td>
<td>0.0</td>
</tr>
<tr>
<td>2.8</td>
<td>0.0083</td>
<td>0.0083</td>
<td>0.0</td>
<td>-0.0413</td>
<td>-0.0413</td>
<td>0.0</td>
</tr>
<tr>
<td>2.9</td>
<td>0.0038</td>
<td>0.0038</td>
<td>0.0</td>
<td>-0.0192</td>
<td>-0.0192</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Appendix C  Comparison of Influence Line Coordinates with RISA-2D  137
Vita

Yaling Li was born in Nanjing, China. She obtained a Bachelor of Civil Engineering degree in 1982 and a Master of Science in Structural Engineering degree in 1985 from the Southeast University, China. She worked as an engineer and a lecturer at Hehai University, China from 1985 to 1991. In January 1994, she enrolled in the Structures Division in the Department of Civil Engineering at Virginia Polytechnic Institute and State University where she was awarded as the member of the Honor Society of PHI KAPPA PHI and the member of the National Civil Engineering Honor Society, CHI EPSILON for her graduate study GPA being 4.0/4.0. In Virginia Tech, she earned the Master of Science in Civil Engineering degree in June 1995.