An Interactive Graphics Oriented Beam Analysis Program

by
Jack Zecher
Indiana University - Purdue University at Indianapolis

ABSTRACT
An interactive graphics based beam analysis program was developed which automates the analysis of beam problems usually encountered in the undergraduate Strength of Materials course. The program provides a highly intuitive graphical user interface that assists in the definition of supports, loads, length and cross-sectional properties of the beam. The program then automatically calculates shear, moment and deflection values along the length of the beam. The results of the analysis are presented in the form of graphical displays of the shear and moment diagrams as well as the deflection diagram. This paper describes the techniques used in the development of the program, as well as the impact that the use of this program can have when used in conjuction with teaching traditional shear and moment diagram techniques.

INTRODUCTION

Techniques for constructing shear and moment diagrams are first introduced during the sophomore level Strength of Materials course. These techniques are then used extensively in the process of calculating stresses and deflections in beams. Students are first introduced to the method of calculating resisting shear and moment values by using the basic principles of static equilibrium. Graphical techniques are then introduced, to further reinforce the relationship between shear and moment values along the beam. When first introduced, students usually struggle to understand the concept. It is almost always necessary to show the students numerous cases involving different loading and support situations until they begin to feel comfortable solving problems on their own.

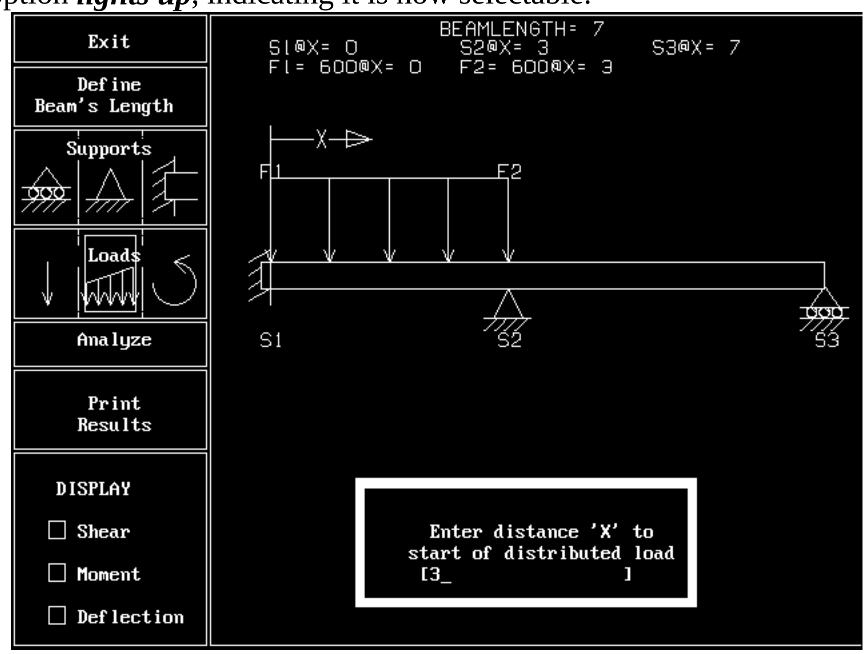
Beam deflections are calculated using both the moment-area method and the principle of superposition. Although the moment-area method is fairly quick, it is easily prone to errors due to the complicated sign conventions. The principle of superposition is performed by solving for the displacement at a single point along the beam by using formulas which are based on simpler beam problems, and then adding the results. This technique is also prone to arithmetic errors because of the large number of variables involved in the formulas. In addition to these arithmetic problems, students often never get to the stage of constructing moment diagrams or calculating deflections, because they make a mistake at a previous stage. For example, if a mistake is made during the calculation of a reaction forces, it then becomes impossible to proceed with correctly constructing the shear and subsequent moment diagram.

It was decided to develop a computer program which could be used in conjunction with the traditional solution techniques. The objective was to develop a program that would allow concentrated forces and moments as well as linearly distributed loads to be applied anywhere along the length of the beam. Pinned, roller and cantilevered supports would be able to be positioned at any location along the beam. The program would also be able to handle both statically determinate and statically indeterminate cases. The program was to serve as a computational tool which would present the overall solution, however, there would still be enough unknown values that the student would have to master the manual computational techniques in order to solve for these values.

CONCEPTUAL DESIGN

A primary goal from the outset was to provide a graphical user interface similar to those currently found in windows based programs. One way in which this was to be achieved, was to provide the process of ***pointing and clicking*** at the option the user wishes to execute. In addition, it was also desirable to provide a convenient means (and some sort of guidance to the user) of moving through the various options in the program. This was done by changing the ***selectability*** status of the various options as more and more parameters are defined.

The screen layout used in the program consists of a tall narrow rectangular area on the left 20% of the screen that contains the program's options, with the remaining portion of the screen being used to display the beam and the resulting shear, moment or deflection diagram. The various options of the program are displayed in icon form and are selectable by a point and click process. Options are ***grayed out*** (in other words, unselectable) until the user has provided enough information to make the selection of a particular option possible. As the user progresses through the definition of a beam problem additional options become selectable. Figure 1 shows the progression through the definition of a beam problem. After the beam's length and section properties are defined, the two options - Supports and Loads - ***light up***, thus indicating that they are now selectable. After defining the necessary supports, to insure stability of the beam, and at least one load has been applied, the Analyze option ***lights up***, indicating it is now selectable.



Figures 1. Various options become selectable as more parameters are defined

In addition to the options displayed along the left edge of the screen, Pop-up windows are incorporated to gather any numerical data that the user must key-in. These values include the length of the beam, magnitude of the loads, position of the supports, etc. As the user progresses through the program, defining loads, supports, etc., these numerical values are listed above the graphical representation of the beam, as shown in Figure 1. In order to allow deletion of incorrectly defined loads or supports, a traditional CAD type interface is used. This allows the user to graphically ***point and click*** at the unwanted load or support.

The analytical solution scheme of the program is based on the finite element method. The decision to use this method was made based on the finite element method's general purpose capability to solve both statically determinate as well as statically indeterminate problems. In addition, the author had previously developed software[1] that is based on this technique. The majority of the program development effort consisted of interfacing with the user, and then storing and converting the conventional beam problem data that the user enters into a format that the finite element analysis solution scheme can process.

PROGRAM ORGANIZATION

All of the pc based labs within the school are equipped with IBM or compatible machines. Due to the fact that some of these are still DOS based, the program had to be able to run in both Windows and DOS based environments. In order to fulfill this requirement, the program was written in QuickBASIC 4.5 rather than Visual BASIC or Visual C. Since QuickBASIC does not provide routines which directly support the mouse as a graphical pointing device, subroutines were constructed based on interrupt calls[2] to perform this function. Pop-up windows were supported by using Bit-Block transfers processes usually associated with computer graphic animation techniques[3].

As the user progresses through the process of defining the beam, the program builds an internal database of the associated values. The database keeps track of the type of support (roller, pinned, or cantilever) or load (concentrated, distributed, or moment) and their locations along the beam. After completing the definition of the beam and associated loads and supports, this database is used to prepare finite element data for the subsequent solution.

INTERFACING TO THE FINITE ELEMENT ANALYSIS

The finite element method is a well known analytical tool that can be used to solve structural analysis, heat transfer, fluid flow and many other classes of engineering problems. The majority of finite element models presented in the current literature or seen in advertisements make use of 2-dimensional quadrilateral or 3-dimensional hexahedral or tetrahedral elements. These elements are used to calculate parameters (such as displacements, temperatures, etc.) at numerous positions within a 2-dimensional or 3-dimensional continuum. However, in order to solve for shear values, moments and displacements along the length of a horizontal beam, a much simpler model can be used. In fact, most general purpose finite element programs have within their library of elements a beam element which has the capability to calculate the values mentioned above. In most cases, the finite element beam element uses the direct stiffness method to arrive at the stiffness matrix of the element, which is the identical representation used in strength of material or structural analysis books[4]. These finite element beams produce results at the nodal points, that are located at both ends of the element as shown in Figure 2.

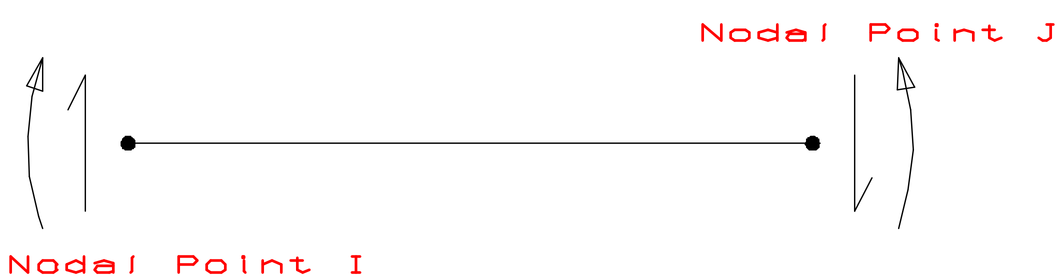


Figure 2. Shear and moment forces are calculated at

both ends of a typical finite element beam

Since values are calculated only at the ends of the finite element beams, the actual beam being analyzed needs to be defined with a large number of finite element beams in order to obtain a somewhat continuous representation of the different values. Also of concern, is that a support or a concentrated load can only be applied at a nodal point. It was decided that the beam be divided into 50 finite element beams, which are initially positioned between 51 evenly spaced nodal points. These nodal points are then adjusted if necessary so that they align with the location of all loads and supports

In order to start the calculation process, the user simply clicks-on the Analyze option. The amount of time required to analyze these models ranges from 18 seconds on a 386DX-20 to less than a second on a Pentium 133 machine.

PRESENTATION OF RESULTS

The shear forces, bending moments and deflection at each nodal point are displayed in a graphical format similar to the traditional techniques used to plot shear, moment and deflection diagrams. As shown in figure 3, the displays consist of vertical lines drawn from the horizontal datum line up or down an amount that corresponds to the value calculated at each nodal point. The ends of the vertical lines are then connected, which gives a continuous appearance to the diagram. There is a possible chance of slightly missing the maximum value of shear or moment, if it were to occur between the nodal points. However, this error would be small due to the relatively fine spacing of the nodal points.

Along with the individual diagram, the numeric maximum value of the parameter being displayed is also listed. Only one diagram is presented at a time, with the user controlling the type of diagram displayed with a single mouse click. The scale of each display is normalized so that regardless of the magnitude of the results, the height and length of the display will always be the same.

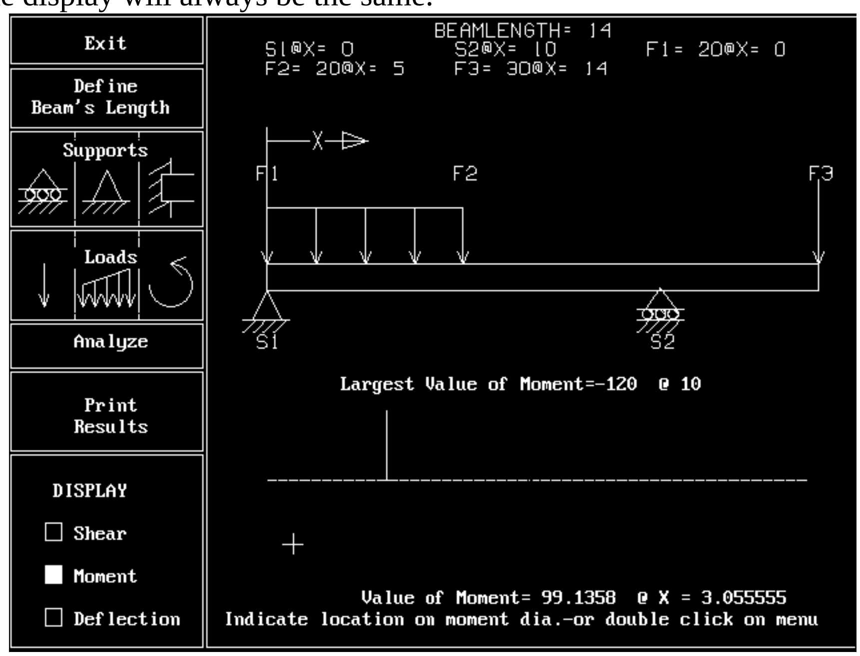


Figure 3. Typical moment diagram

CONCLUSION

Specialized computer software such as the program described in this paper can be a very important tool in problem solving classes. This particular program gives students a graphical solution to the overall problem along with the numerical value of the maximum value of the parameter being displayed. However, it still leaves enough unknown values that the student must master the manual computational techniques in order to solve for these remaining values. The program's graphical user interface makes it easy to learn, so that a minimal amount of time needs to be devoted to instructing students on how to use the program.

For students just beginning to learn shear and moment diagrams, the program acts as a tool that can be used to check their assigned problems. It also serves as a resource which can help them out, when they get stuck with a problem and are not able to contact their instructor until the next day or two. The program can also be used to introduce students to topics which are beyond the scope of what is normally covered in the course, such as, statically indeterminate beams.

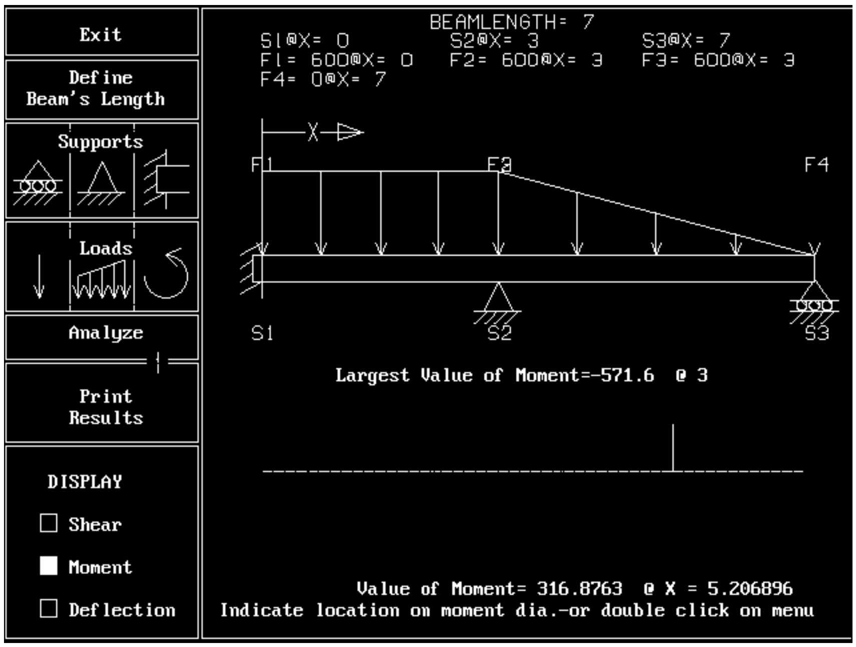


Figure 4. A statically indeterminate problem

As students progress through higher level courses, such as Design of Machine Elements, and Senior Design Projects, the program can serve as a valuable resource to perform calculations on shafts, frame rails, etc.

REFERENCES

1) BeamBoss - Computer Program, Software Consulting Specialists, Ft. Wayne, IN 1985
2) *Microsoft Mouse Programmer's Reference*, by Microsoft Press, 1989
3) Zecher, J. E., *Computer Graphics for CAD/CAM Systems*, Marcel Dekker, 1994
4) Cook, R. D., *Finite Element Modeling for Stress Analysis*, John Wiley & Sons, 1995

Jack Zecher is a registered professional engineer in the state of Indiana and Professor of Mechanical Engineering Technology at Indiana University - Purdue University at Indianapolis. He has worked on the

development of interactive computer aided analysis tools in both industry and academia during the past 20 years. He holds both a B.S., from Miami Univ. and a M.S., from Western Michigan Univ.

An interactive beam analysis program called GT BEAM, that has been developed at Georgia Tech for use in undergraduate mechanics courses, is presented. The development objectives of the program are discussed and how the program has been used effectively to increase the learning and understanding of indeterminate beam structures is presented. Practical experience with the program in two elementary structures courses is also discussed. The paper shows how graphically oriented programs can be beneficial to the education of first year engineering students by allowing them to perform 'what-if&a ABSTRACT An interactive graphics based beam analysis program was developed which automates the analysis of beam problems usually encountered in the undergraduate Strength of Materials course. The program provides a highly intuitive graphical user interface that assists in the definition of supports, loads, length and cross-sectional properties of the beam. The program then automatically calculates shear, moment and deflection values along the length of the beam.Â Zecher, J. (1996, June), An Interactive Graphics Oriented Beam Analysis Program Paper presented at 1996 Annual Conference, Washington, District of Columbia. 10.18260/1-2--6136. Ã—. APA. The theory of the design and analysis of time-invariant beam-formers is well documented. Several standard algorithms such as delay-and-sum, Minimum Variance Distortionless Response (MVDR) [5] and beam pattern synthesis are commonly used and are relatively straightforward to implement. Performance metrics such as directivity index, front-back ratio and white noise gain are also well understood, and many standard textbooks e.g. [1, 2, 6, 7, 8] provide gures depicting the inuence of design parameters on per-formance metrics.Â This paper describes BFGUI [9], an interactive graphical tool programmed in MATLAB for simulating microphone arrays, syn-thesizing beamformers, and investigating performance metrics as design parameters are varied.