

DESIGN AND IMPLEMENTATION OF VERSATILE BEAM ANALYZER

By S. T. Mau,¹ Fellow, ASCE, and X. Tao²

ABSTRACT: A versatile beam analyzer is presented in this note. It displays shear, moment, and deflection diagrams as well as influence line diagrams of a two-dimensional beam. The computing engine of the program was carefully designed to handle changing segmental flexural rigidity easily as well as to minimize computing time. Coupled with an effective graphical user interface, the analyzer responds instantaneously to a user's on-screen play. The analyzer can be used both as a learning tool and as a tool for preliminary design of continuous beams. The methodology used for the development of this beam analyzer can be extended to other specialty computer programs.

INTRODUCTION

The introduction of object-oriented programming (OOP) makes it possible to develop sophisticated programs in a relatively short time through the use of "objects" that are highly portable (Ross et al. 1992a; 1992b). The first writer of this paper developed a simple program for the display of shear, moment, and deflection diagrams of a three-support beam under a single concentrated load using graphical user interface (GUI) and OOP (Mau and Aruna 1994). It was for the NeXT station. Recently, a powerful beam program for Macintosh computers was published as part of a comprehensive computer-aided education program for mechanics of materials (Cooper and Miller 1996).

In this note a versatile beam analyzer is presented. It is developed for use on IBM compatible PCs using the Borland C++ compiler (Cantu and Tendon 1992). It expands the scope of the previously developed three-support single-load analyzer to include a variety of support, connection, and loading combinations and allows the change of beam flexural rigidity from segment to segment. In addition to the shear, moment, and deflection diagrams, the influence line diagrams can also be displayed upon user command. A distinguished feature of this program is the way the changing segmental flexural rigidity is handled, as shall be described. The resulting computing engine is very efficient and consumes little time; coupled with an effective GUI, it produces an instantaneous response to any user on-screen play. In the following, the design process of this analyzer is described. Some examples are given to illustrate the versatility of the program.

ATTRIBUTES OF BEAM ANALYZER

Within the limits of a two-dimensional, linear-elastic, small-deflection engineering beam theory, the beam analyzer is to show shear, moment, and transverse deflection diagrams or influence-line diagrams for a variety of support, connection, and loading conditions. Change of any conditions should be effected through on-screen graphics interface, and the effect of any change should be shown instantaneously.

The beam section property is to be represented only by its flexural rigidity, EI . Along the beam length, however, the EI

value is allowed to vary in a piecewise constant fashion, i.e., the beam can be divided into a number of segments, each with a constant but different EI (see Fig. 1).

The left and right ends of the beam can be supported in any way the user wishes. The user can choose from among three general categories of support—free, hinged, or fixed (Fig. 1). For each category, the support conditions are divided into vertical and rotational conditions and the user is prompted to supply further data on vertical displacement and rotation. Based on the user input, the program is to internally generate a support type as well as its associated support value.

Internal supports and internal hinges can also be included (Fig. 1). Each internal support can be either a support with a specified vertical displacement (including zero) or a spring with specified flexibility.

Three types of transverse loading are included: concentrated load, distributed load of constant intensity, and concentrated moment.

COMPUTING ENGINE

Referring to Fig. 1, the basic unknowns can be identified as the following: four end reactions F_R , M_R , F_L , and M_L ; reaction forces at internal supports F_j ($j = 1, 2, \dots, S$); deflection and rotation at left end W_L and θ_L ; and relative rotations at internal hinges θ_j ($j = 1, 2, \dots, C$). The reason for the selection of these parameters as unknowns will become clear when the control equations are described. Thus, a total of $4 + S + C + 2$ equations are needed, where S and C are the numbers of internal supports and internal hinges, respectively. These equations are listed in Table 1 for the case of constant flexural

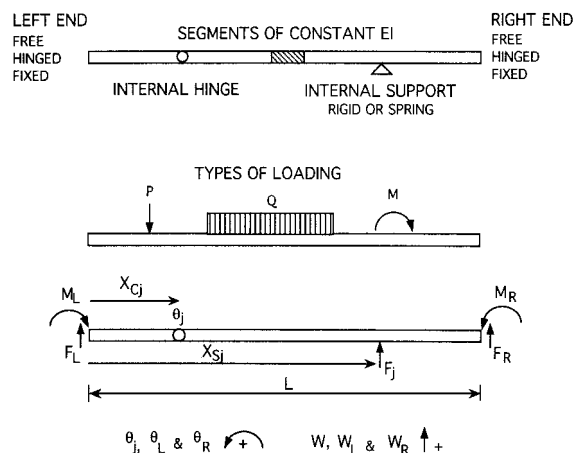


FIG. 1. Support, Connection, Loading, and Reaction Definitions of Beam

¹Dean, Newark College of Engrg., New Jersey Inst. of Technol., Newark, NJ 07032. E-mail: mau@admin.njit.edu; formerly, Prof., Dept. of Civ. and Envir. Engrg., Univ. of Houston, Houston, TX 77204-4791.

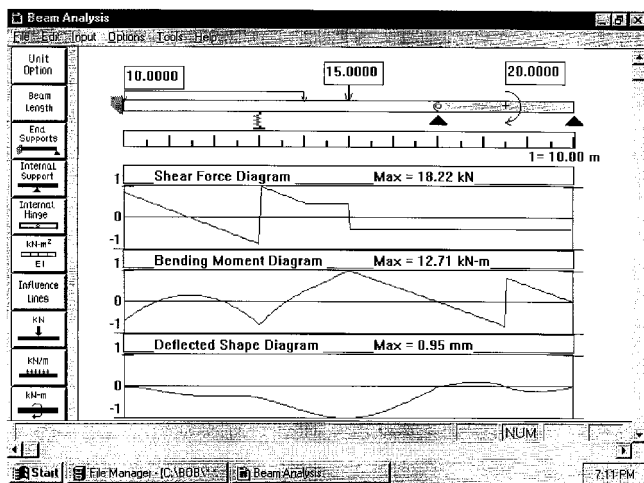
²Grad. Student, Dept. of Civ. and Envir. Engrg., Univ. of Houston, Houston, TX.

Note. Discussion open until June 1, 1999. To extend the closing date one month, a written request must be filed with the ASCE Manager of Journals. The manuscript for this technical note was submitted for review and possible publication on July 24, 1997. This technical note is part of the *Journal of Computing in Civil Engineering*, Vol. 13, No. 1, January, 1999. ©ASCE, ISSN 0877-3801/99/0001-0050-0052/\$8.00 + \$.50 per page. Technical Note No. 16278.

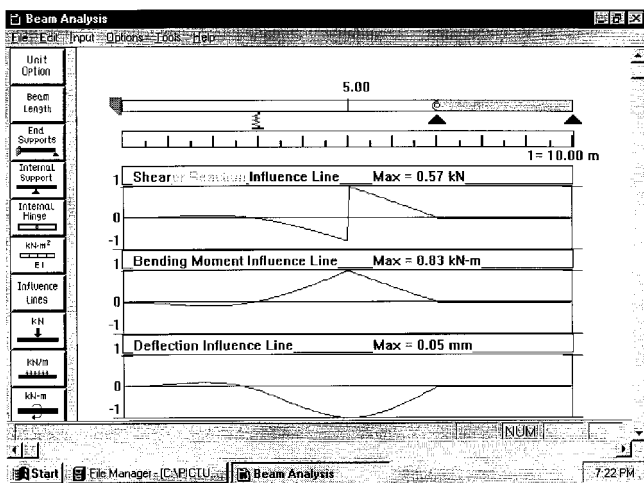
rigidity EI throughout the beam. Modifications needed to accommodate different EI for different segments of the beam are lengthy but easily derived. They are not included here, for brevity. A brief description of the nature of these equations is in order.

The first two equations are the force and moment equilibrium equations of the whole beam. The contributions of the applied concentrated load, distributed load, and concentrated moments are all lumped to the right-hand-side (RHS) of the equations. The next C equations are moment equilibrium equations of the free-body-diagram to the left of the i th internal hinge, i.e., the summation of moment about an internal hinge of all forces and moments to the left of the hinge is zero. The above $2 + C$ equations, being equilibrium equations, are not dependent on the flexural rigidity of the beam. These equations involve only the force unknowns.

The next S equations are compatibility equations requiring that the total contribution to the deflection at the i th internal support be zero, when calculated by integration from the left end. If a prescribed support settlement is given, it is moved to the right-hand-side (denoted as W'_i in Table 1). Because the deflection at any point is calculated by integrating from the beam equation using the left end as the origin, it depends on the deflection and rotation of the left end as well as those at any internal support to the left of the point in question. Thus,



(a)



(b)

FIG. 2. (a) Example Problem Showing Shear, Moment, and Deflection Diagrams; (b) Influence Lines of Shear, Moment, and Deflection of Middle Section (5 m from Left End)

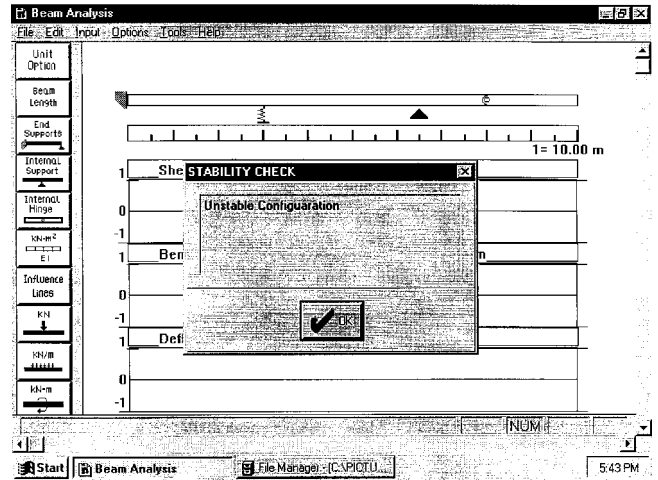


FIG. 3. Warning of Unstable Configuration

it is necessary to include the left-end deflection and rotation, W_L and θ_L , and the rotational jump at any internal hinges, θ_i , as unknown parameters, while the right-end deflection and rotation, W_R and θ_R , need not be.

The nature of the last four equations depends on how the right end and the left end are constrained. The first two are related to the vertical and rotational constraints, respectively, of the right end, and the last two are for the same of the left end. Each equation has three possibilities as shown in Table 1. For example, if the vertical or rotational constraint type is designated as free ($rs_type=0$, $ls_type=0$, or $rr_type=0$, $lr_type=0$ in Table 1), then the correct condition becomes support reaction force or moment be zero. If the right-end vertical or rotational constraint is designated as rigid ($rs_type=1$ or $rr_type=1$ in Table 1), then the right-end vertical deflection or rotation must equal to the prescribed value, which could be nonzero (rs_value or rr_value in Table 1).

The $2 + C + S + 4$ equations are solved as soon as the user specifies a stable configuration. Furthermore, the shear, moment, and deflection values at 201 equally spaced points along the beam are calculated. The user can also specify a particular location to show the value of the shear, moment, and deflection. A function is written to compute such information on demand. If the user specifies the influence line mode of the beam analyzer and gives a location value along the beam, the shear, moment, and deflection influence lines of the cross section are computed at the same 201 points by sweeping a unit load through. The shear influence line would automatically change to a reaction influence line if the specified location coincides with that of an internal support.

Examples

Shown in Fig. 2(a) is the on-screen display of an example problem and its shear, moment, and deflection diagrams. Shown in Fig. 2(b) are the influence line diagrams of a middle section of the same beam.

An interesting feature of the program is the ability to judge whether a configuration is stable. It will display a warning whenever the support conditions render the beam unstable (see Fig. 3), so that the user knows additional supports are needed or undesirable internal hinges must be deleted or relocated.

What cannot be illustrated in static figures is the dynamic nature of the GUI. Most of the on-screen objects can be manipulated by the mouse to effect an instant response. This beam analyzer is now available to the public free of charge. Interested readers may send a message by e-mail to the writer.

TABLE 1. Matrix of Control Equations and Right-Hand-Side (RHS) Vector

Equations and unknowns (1)	F_R (2)	M_R (3)	F_L (4)	M_L (5)	$F_1 \dots F_i \dots F_s$ (6)	W_L (7)	θ_L (8)	$\theta_1 \dots \theta_j \dots \theta_c$ (9)	RHS (10)
$\Sigma F_y=0$	1	—	1	—	1 ... 1 ... 1	—	—	—	$\Sigma F'$
$\Sigma M_R=0$	—	-1	L	1	$L - X_s$	—	—	—	$\Sigma M'_R$
$M_i=0$ · $M_i=0$ · $M_c=0$	—	—	X_{Ci} · · ·	1 · · 1	$(X_{Ci} - X_{Si})^a$ · · ·	—	—	—	$\Sigma M'_i$
$W_i=0$ · $W_i=0$ · $W_s=0$	—	—	$\frac{X_i^3}{6EI}$ · · ·	$\frac{X_i^2}{2EI}$ · · ·	f_{ij}^b · · ·	1 · 1 · 1	X_{Si} · · ·	$(X_{Si} - X_{Cj})^a$ · · ·	W'_i
rs_type=0 (free), $F_R=0$, OR rs_type=1 (rigid), $W_R=-rs_value$, OR rs_type=2 (spring), $W_R+rs_value^a F_R=0$	1 0 rs_value	—	0 $L^3/6EI$ $L^3/6EI$	0 $L^2/2EI$ $L^2/2EI$	0 $(L - X_i)^3/6EI$ $(L - X_i)^3/6EI$	0 1 1	0 L L	0 ... 0 ... 0 L- X_{Cj} L- X_{Ci}	0 A-rs_value A
rr_type=0 (free), $M_R=0$, OR rr_type=1 (rigid), $\theta_R=-rr_value$, OR rr_type=2 (spring), $\theta_R-rr_value^a M_R=0$	—	1 0 -rr_value	0 $L^2/2EI$ $L^2/2EI$	0 L/EI L/EI	0 $(L - X_i)^2/2EI$ $(L - X_i)^2/2EI$	—	0 1 1	0 ... 0 ... 0 1 ... 1 ... 1 1 ... 1 ... 1	0 B-rr_value B
ls_type=0 (free), $F_L=0$, OR ls_type=1 (rigid), $W_L=-ls_value$, OR ls_type=2 (spring), $W_L+ls_value^a F_L=0$	—	—	1 0 ls_value	—	—	0 1 1	—	—	0 -ls_value 0
lr_type=0 (free), $M_L=0$, OR lr_type=1 (rigid), $\theta_L=-lr_value$, OR lr_type=2 (spring), $\theta_L-lr_value^a M_L=0$	—	—	—	1 0 -lr_value	—	—	0 1 1	—	0 -lr_value 0

Note: A = downward deflection at right end due to external loadings; B = clockwise rotation at right end due to external loadings. rs_type, rr_type, ls_type, lr_type = support types of right or left (r or l) end for vertical or rotational constraints (s or r). rs_value, rr_value, ls_value, lr_value = displacement or rotation value (if type=1), vertical or rotational spring flexibility (if type=2).

^aOnly if $X_{Si} - X_{Cj} > 0$, otherwise zero.

^b $f_{ij} = f_j + (X_{Si} - X_{Cj})^3/6EI$ if i not less than j , otherwise zero, where f_j = spring flexibility.

APPENDIX. REFERENCES

Cantu, M., and Tendon, S. (1992). *Borland C++ 3.1 object-oriented programming*. Bantam, New York.
 Cooper, S. C., and Miller, G. R. (1996). "A suite of computer-based tools for teaching mechanics of materials." *Comp. Appl. in Engrg. Educ.*, 4(1), 41-49.
 Mau, S. T., and Aruna, V. (1994). "Application of OOP and GUI to the development of educational software in structural mechanics. Proc., 1st

Congr. on Comp. in Civ. Engrg., *J. Comp. in Civ. Engrg.*, ASCE, Reston, Va., 8(2), 1273-1279.
 Ross, T. J., Wagner, L. R., and Luger, G. F. (1992a). "Object-oriented programming for scientific codes. I: thoughts and concepts." *J. Comp. in Civ. Engrg.*, ASCE, 6(4), 480-496.
 Ross, T. J., Wagner, L. R., and Luger, G. F. (1992b). "Object-oriented programming for scientific codes. II: examples in C++." *J. Comp. in Civ. Engrg.*, ASCE, 6(4), 497-514.