Calhoun
Institutional Archive of the Naval Postgraduate School

Author(s)	Lowe, Gary B.
Title	Digital computer program for flexural analysis of beams
Publisher	Monterey, California: U.S. Naval Postgraduate School
Issue Date	1963
URL	http://hdl.handle.net/10945/12011

This document was downloaded on May 22, 2015 at 07:54:49

DUDLEY KNOX LIBRARY

http://www.nps.edu/library

Calhoun is a project of the Dudley Knox Library at NPS, furthering the precepts and goals of open government and government transparency. All information contained herein has been approved for release by the NPS Public Affairs Officer.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/

NPS ARCHIVE
1963
LOWE, G.

DIGITAL COMPUTER PROGRAM FOR FLEXURAL ANALYSIS OF BEAMS

GARY B. LOWE

LIBRARY
U.S. NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

DIGITAL COMPUTER PROGRAM

FOR

FLEXURAL ANALYSIS OF BEAMS

* * * *

Gary B. Lowe

DIGITAL COMPUTER PROGRAM

For

FLEXURAL ANALYSIS OF BEAMS

by

Gary B. Lowe

Lieutenant, United States Navy

Submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

United States Naval Postgraduate School
Monterey, California

1963

LIBRARY
U.S. NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

DIGITAL COMPUTER PROGRAM
FOR
FLEXURAL ANALYSIS OF BEAMS

* * * *

Gary B. Lowe

This work is accepted as fulfilling
the thesis requirements for the degree of
MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING
from the
United States Naval Postgraduate School

TABLE OF CONTENTS

Abstract.

This paper describes a solution to the flexural analysis of a beam by digital computer. Subroutines were written which calculate shear, moment, slope and deflection at any specified point on a beam if the loading is known. Shear deflection may be included for many support conditions. Neither elastic supports nor column effects are considered. The subroutines solve problems of variable beam cross-section and loading by utilizing an extension of "McCaulay's Method", a generalized step function.

The main program provides input-output facilities and also solves for indeterminate reactions on the beam.

Problems and their solutions, showing the versatility of this program, are also included.

Acknowledgment.

The author wishes to thank Professor Gilles Cantin for his continued advice, direction, and encouragement. The director and staff of the computer center were very generous with their help and time. Professor John E. Brock's assistance was also appreciated.

List of Illustrations

1. Introduction.

The routine solution to engineering problems by digital computer appears to be limited only by the scope of the available programs. A flexible program permitting the solution of routine problems in flexural analysis of beams would seemingly fill a need in the field of digital computer applications to Mechanics. The present work is devoted to such an idea.

Five subroutines, "LOAD", "SHEAR", "MOMENT", "SLOPE", and "DEFLECT", were written which will calculate the load, shear, moment, slope, and deflection, respectively, at any indicated point on a beam if the indeterminate quantities are calculated elsewhere. The use of generalized step functions in this project allows for piecewise linear variation of distributed loading, bending compliance, $\frac{1}{EI}$, and shear compliance, $\frac{K}{AG}$.

The main program, "BEAM3", first generates and then solves a set of simultaneous linear equations, calculating the indeterminate quantities of the beam. The above mentioned subroutines are then utilized to calculate the remaining flexural quantities. This program also provides input-output facility.

2. Definition of the problem.

The problem of structural beam analysis is one in which the external loading applied to the beam is well defined, as is the geometric configuration and the nature of the beam itself. Where information about such a beam is required, this information will most likely fall into the following categories:

1. Internal shear forces, V, and moments, M.
2. Slope, θ.
3. Deflection, y.
4. Redundant reactions, either forces, Ry, or moments, RM_z.

"Flexural analysis" is used here as meaning the process by which this information is generated from the known data.

In attacking this problem of flexural analysis, the intention was to limit the scope as little as possible. However, several customary simplifications and limitations were made. First, elastic supports and beam column effects were not considered. Limitations concerning the type and number of external loads and the geometry of the beam will be discussed later. Also the basic relations which follow, inherently restrict the deflection of the beam to small values and the internal stresses to values below the elastic limit of the structural material. Shear deflection, which is generally neglected in a problem of this type, has been included in some problems.

a. Bending deflection (y_1).

The Bernoulli-Euler-Navier equation governing the elastic curve of a beam due to bending, consistent with the sign convention shown in Appendix B is:

$$\frac{M}{EI} = \text{curvature} = (d^2y/dx^2)\left[1+(dy/dx)^2\right]^{-3/2}$$

and we use the approximation, curvature $\approx d^2y/dx^2$.

The following equations, derived from the above equation and shown in Timoshenko /5/,[1] with differences due to the chosen sign convention, describe the flexure of a beam (See Appendix B for notation).

$$W = q(x) \quad ,$$

$$V = \int W dx \quad ,$$

$$M = \int V dx \quad ,$$

$$\theta_1 = \frac{dy_1}{dx} = \int \frac{M}{EI} dx \quad ,$$

$$y_1 = \int \theta_1 dx \quad ,$$

b. Shear deflection (y_2).

The equation governing the shear deflection of a beam, also given in Timoshenko /5/, compatible with the stated sign convention becomes:

$$\frac{dy_2}{dx} = -\frac{KV}{AG}$$

It is evident that:

$$y_2 = \int \left(-\frac{KV}{AG}\right) dx$$

c. Total deflection (y).

The equations used in this thesis are the result of the sum of the above shear and bending deflections. They are:

$$W = q(x) \quad ,$$

[1]Numbers so indicated refer to the bibliography.

$$V = \int_0^x W dx + \sum F_i *$$

$$M = \int_0^x V dx + \sum M_i *$$

$$\theta = \int_0^x \frac{M}{EI} dx - \frac{KV}{AG} + C_1 **$$

$$Y = \int_0^x \theta \, dx + C_2 **$$

Note that step-discontinuities may appear in all of the above equations, except in the equation for deflection. These discontinuities limit the methods that can be used in handling the equations.

Three methods that could be used to solve the equations are a pure analytical approach, a step-by-step numerical integration scheme, and the "curly bracket" method. Of these methods, the analytical approach is generally restricted to a specific problem and is unsuited for adaptation to digital computer programming. The numerical integration scheme can be used in the computer solution of flexural analysis and may have been programmed; however, our search of literature has not uncovered any such program. The "curly bracket" method was chosen for use in this thesis because it was felt that it would result in a more exact solution requiring less

*In these equations, $\sum F_i$ and $\sum M_i$, represent the sum of the point forces and moments, respectively, applied to the beam.

**C_1 and C_2 are constants of integration.

3. "Curly bracket" method

The "curly bracket" method is based upon the properties of a convenient notation defined as follows:

$$\{x-a\}^n = 0 \qquad \text{if } x<a$$

$$= (x-a)^n \qquad \text{if } x>a$$

where n is an integer $\geqq 0$.

Appendix A contains further properties of expressions containing curly brackets. The notation is generally credited to McCaulay /1/, but similar, though less useful notations can be traced to A. Föppl and St. Venant. Also, the ideas involved are related to the use of the Laplace Transformation with the result that this method or approximations to it are reinvented frequently /8/ /9/.

Using this method allows one equation, valid over the entire length of the beam, to be expressed for each of the basic equations of flexure. The particular value of this method is that it permits the integration of these equations to be carried out easily and has the virtue of adjusting the constants of integration most conveniently.

In addition, for use with a digital computer, the "switching" property of the curly bracket can be easily handled in FORTRAN by use of an "IF" statement.

A simple example should convince most readers of the convenience of this method. Examine the following problem, which consists of a uniform beam, simply supported at each end with a concentrated load at the center of the span.

stant variation of bending compliance, $\frac{1}{EI}$. In this thesis the method has been further expanded to include beams with piecewise linear variation of both bending compliance and shear compliance, $\frac{K}{AG}$.

4. Implementation.

Although the formulas shown in Appendix A could be expanded further, a piecewise linear variation of distributed loads, bending compliance, and shear compliance was considered capable of providing sufficient accuracy and flexibility for the resulting program.

Fig. 2(a) illustrates a distributed load on a beam whose origin is at x=0. Fig. 2(b) shows an element of the distributed load.

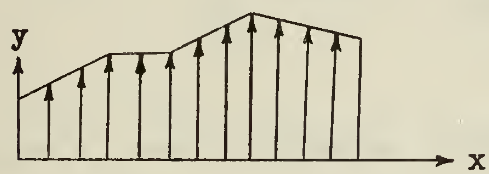


Fig. 2(a)

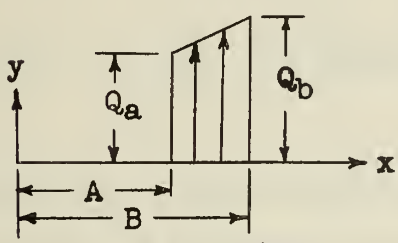


Fig. 2(b)

The analytical expression for this element is:

$$Q(x) = Q_a \{x-A\}^0 + \frac{Q_b - Q_a}{B-A}\{x-A\}^1 - Q_b\{x-B\}^0 - \frac{Q_b-Q_a}{B-A}\{x-B\}^1$$

The same relationship is used for an element, f(x), of bending compliance and an element, g(x), of shear compliance upon substituting f for Q and g for Q respectively.

The basic equations of flexure can now be expressed in the same manner as they were utilized in programming. They are:[2]

$$W(x) = \sum_{i=1}^{NOL} Q_i(x) \tag{1}$$

$$V(x) = \int_0^x W(x)dx + \sum_{i=1}^{NOF} F_{y_i}\{x-x_{f_i}\}^0 + \sum_{i=1}^{NOR} R_{y_i}\{x-x_{r_i}\}^0 \tag{2}$$

where F_{y_i} are upward, known forces at $x=x_{f_i}$ and R_{y_i} are upward, unknown reactions at $x=x_{r_i}$

$$M(x) = \int_0^x V(x)dx + \sum_{i=1}^{NOM} M_{z_i}\{x-x_{m_i}\}^0 + \sum_{i=1}^{NORM} RM_{z_i}\{x-x_{rm_i}\}^0 \tag{3}$$

where M_{z_i} are clockwise, known moments at $x=x_{m_i}$ and RM_{z_i} are clockwise, unknown reactive moments at $x=x_{rm_i}$

$$\theta = \frac{dy}{dx} = \int_0^x \left[M(x) \sum_{i=1}^{NOI} f_i(x)\right]dx - \left[V(x) \sum_{i=1}^{NOK} g_i(x)\right] + C_1 \tag{4}$$

where C_1 is the constant of integration.

Appendix A indicates how the product and the integration of the product of the curly bracket symbols are formed.

$$y = \int_0^x\int_0^x \left[M(\xi)\sum_{i=1}^{NOI} f_i(\xi)\right] d\xi\, dx - \int_0^x \left[V(x)\sum_{i=1}^{NOK} g_i(x)\right]dx + C_1x + C_2 \tag{5}$$

where C_2 is the constant of integration.

Equations (1) through (5) when properly coded in FORTRAN

[2]The numbers, NOL, NOF, NOR, NORM, NOI and NOK, appearing above the summation signs are program input parameters indicating the number of terms for each problem.

form the core of the five basic subroutines: LOAD, SHEAR MOMENT, SLOPE and DEFLECT. It must be noted before discussing the subroutines that equations (2) through (5) involve the unknown values C_1, C_2, R_{y_i}, and RM_{z_i}, where there may be as many as ten values each of R_y and RM_z. The calculation of these values is done in the main program BEAM3 and will be discussed in section 6.

5. The basic subroutines.

Subroutines LOAD, SHEAR and MOMENT.

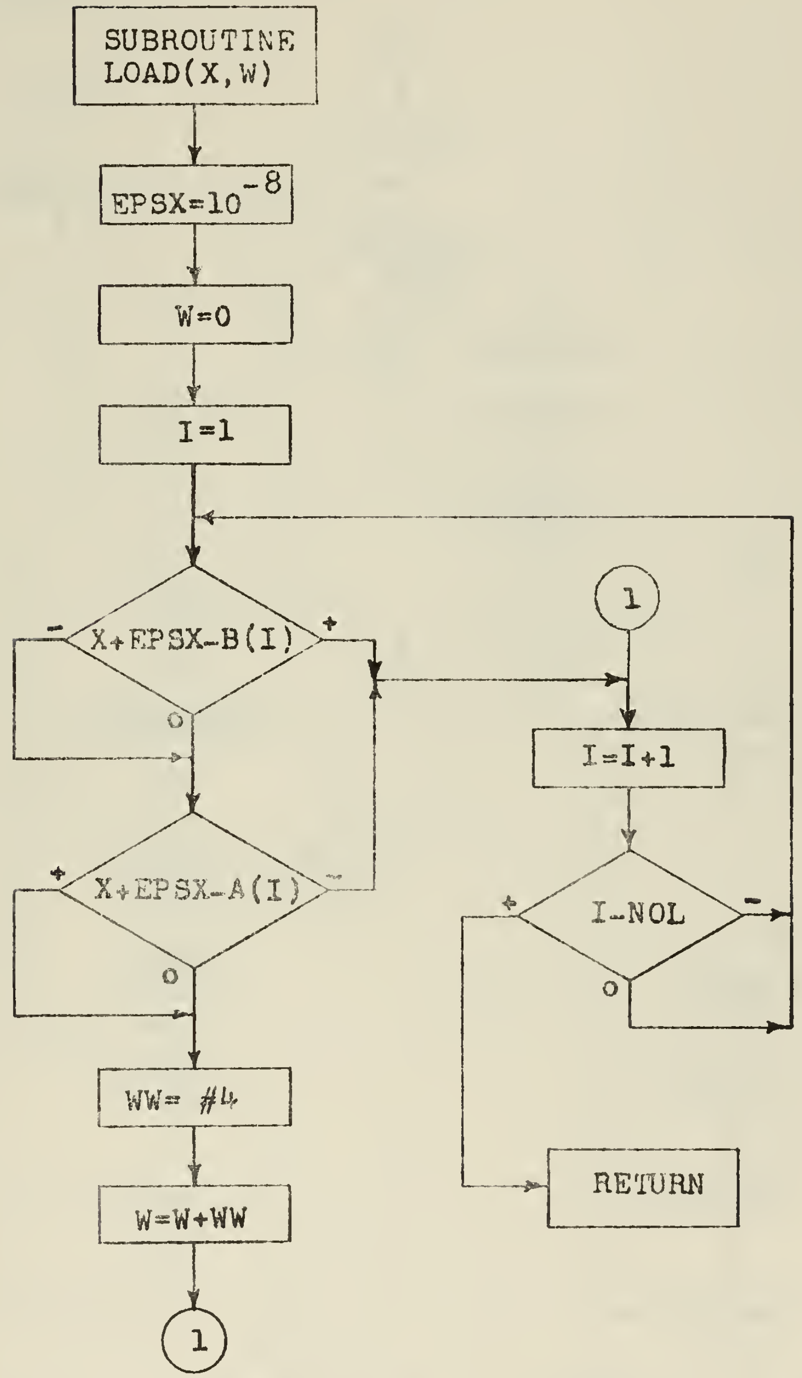
The above named subroutines, as their names imply, are employed to calculate the total distributed load, shear and moment, respectively, at any desired point on the beam. The expanded forms of equations (1), (2) and (3) were used in writing these subroutines. Utilizing the distance, x_1, as the entering argument, the subroutines select and sum the appropriate terms in the corresponding equation by inspecting all loading terms and discarding those in which the value within the curly bracket is negative. Actually the load, shear and moment are calculated a small increment, $\mathcal{E} = 10^{-8}$, to the right of x_1 in order to obtain the right side value of any step-discontinuities occuring at the point, x_1 (See flow charts on pages 13, 14 and 15).

Subroutine SLOPE.

This subroutine is equation (4), excluding the constant of integration, C_1, in FORTRAN language (See flow chart on page 17). Note that the equation can be expressed as a summation of terms, which are constants times the multiplication of two step-functions of the form $C\{x-a\}^n\{x-b\}^m$, or the summation of the integral of the same type of terms, where n equals either 0 or 1.

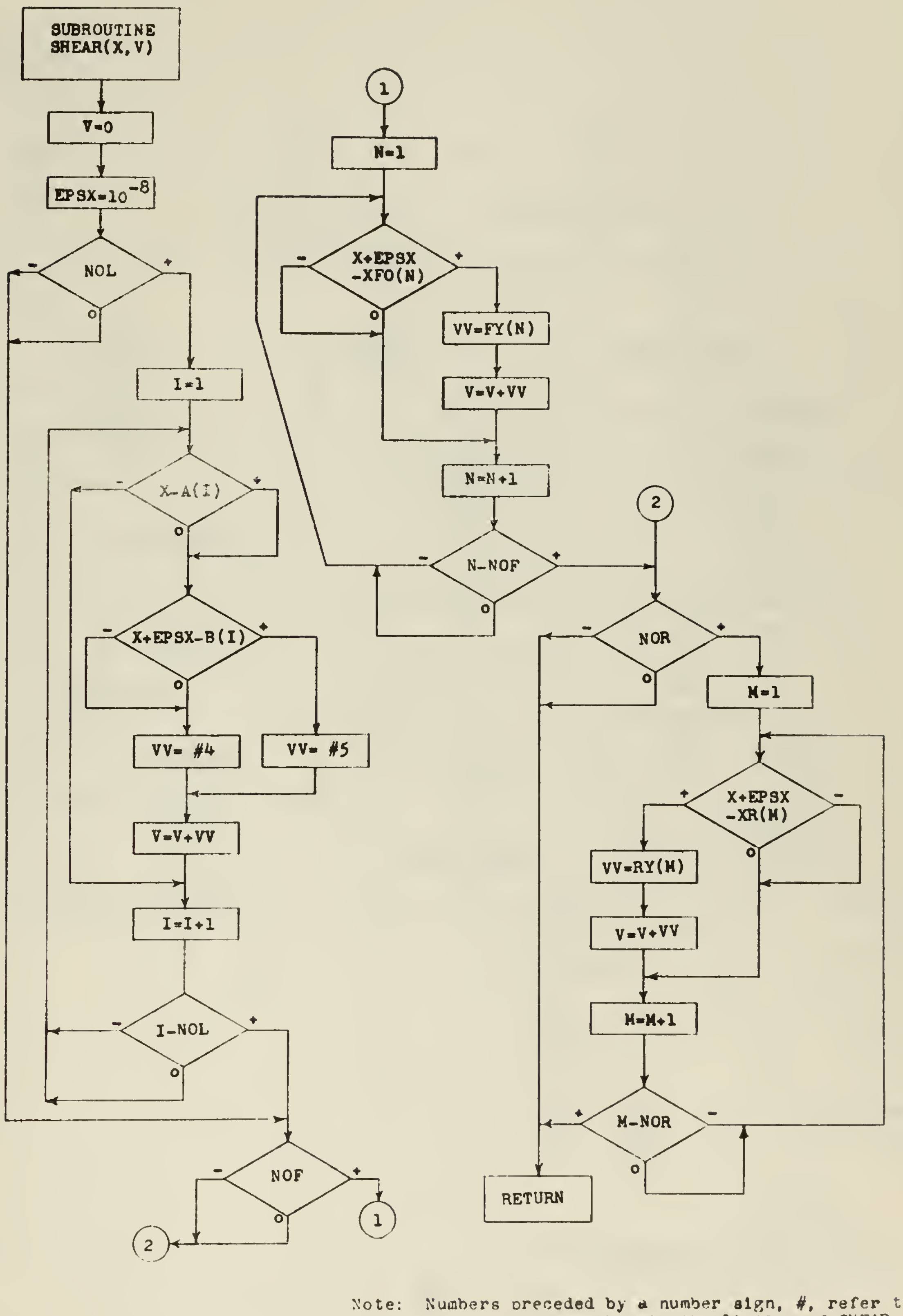
Function subroutines, FT0N and FT1N (see Appendix A), will calculate the value of $\{x-a\}^o\{x-b\}^m$ and $\{x-a\}^1\{x-b\}^n$, respectively. Function subroutines, ENT0N and ENT1N (See

Flow Chart of LOAD



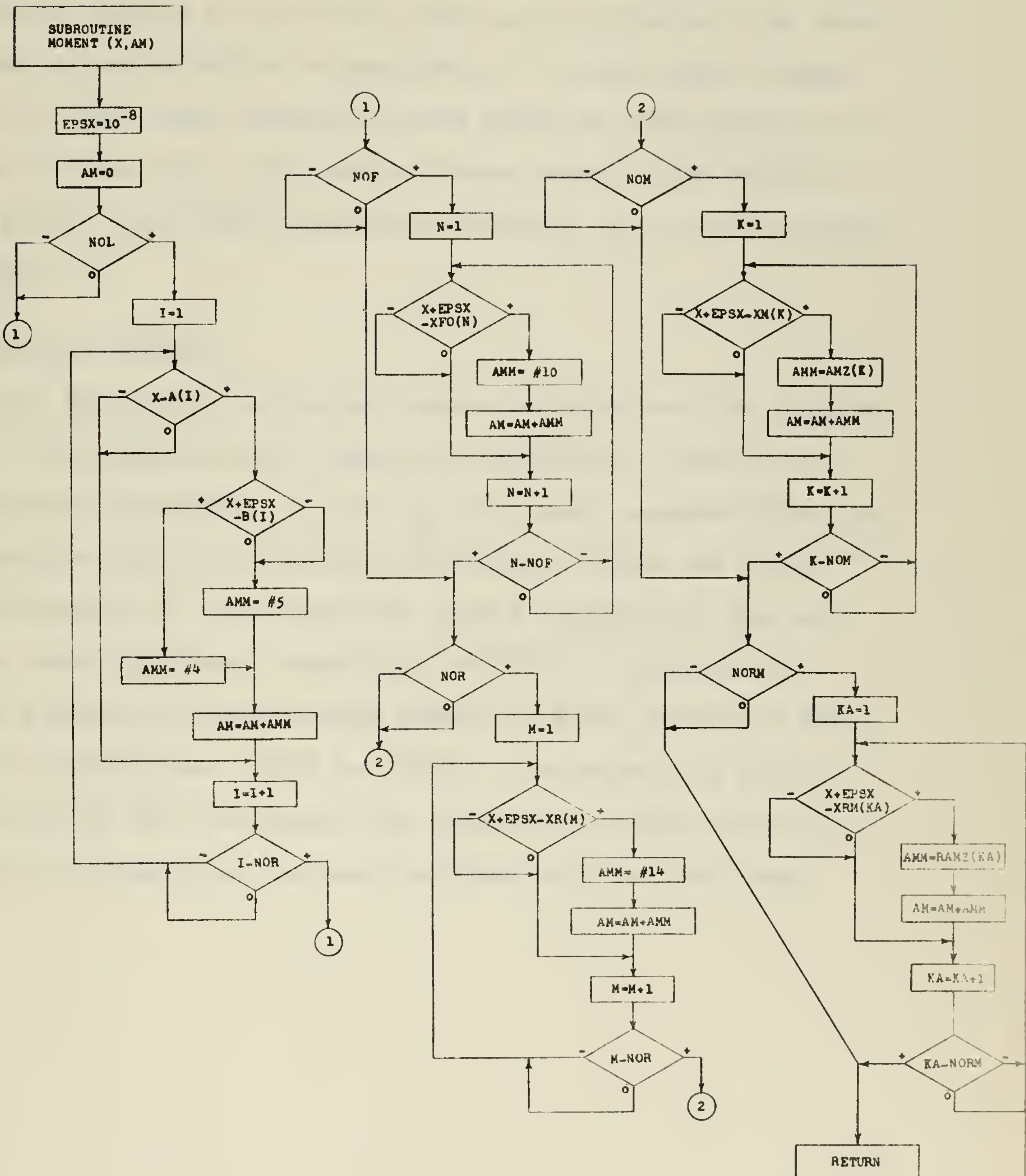
Note: Numbers preceded by a number sign, #, refer to the statement number in the listing of LOAD.

Flow Chart of SHEAR



Note: Numbers preceded by a number sign, #, refer to the statement number in the listing of SHEAR.

Flow Chart of MOMENT



Note: Numbers preceded by a number sign, #, refer to the statement number in the listing of MOMENT.

Appendix A), will calculate the first integral of these functions. Equation (4) can now be expressed as the summation of constants multiplied by the appropriate function subroutines.

SLOPE calculates all combinations of M(x) and f(x) by stepping through all the f(x) terms and varying all the terms in the moment equation at each step. It then steps through all the g(x) terms varying all the terms in the equation for shear at each step. The sum of these terms is the value of the slope, minus the integration constant, C_1, at the entered distance, x.

Subroutine DEFLECT.

At this point the author wishes to point out the similarity of the equations for slope and deflection. Since the integration constants, C_1 and C_2, are again excluded from the subroutine, and the function subroutines, DITON and DIT1N (See Appendix A), calculate the second integral of the multiple step-functions, subroutine DEFLECT is identical to SLOPE except for substituting ENTON for FTON, ENT1N for FT1N, DITON for ENTON and DIT1N for ENT1N. The result is a subroutine that will calculate the deflection, minus the constants of integration, at any required point on the beam.

6. Program BEAM3.

The main program, BEAM3, can be divided into four distinct steps. These steps are: reading the input data, calculation of indeterminate quantities, calculation of the flexural quantities, and the output of the solution.

The program first reads the input data which includes the length of the beam, all external loading, the location of all reactions, the flexural properties of the beam, and a group of parameters giving the number of each type of load or reaction. It also provides for reading a predetermined value of deflection, PDEFT, at each reaction and a predetermined value of slope, PSLP, at each moment reaction.

The program then proceeds to solve for the indeterminate quantities of the beam by utilizing the two equations of static equilibrium plus equations obtained from the known boundary condition at each reaction or reactive moment. The first two equations are simply the summation of the forces on the beam equals zero and the summation of the moments about the right end of the beam equals zero. The remainder of the equations are obtained from the fact that the slope at each reactive moment or the deflection at each reaction is known. Using this fact in the appropriate one of either equation (4) or (5) results in a set of linear equations for the indeterminate quantities.

The equations thus obtained are generated in a matrix form that is compatible with subroutine GAUSS2 (See Appendix E). This subroutine is then called to solve the equations simultaneously. If the matrix is singular, GAUSS2 exits to the calling program which in turn prints "MATRIX SINGULAR" and then stops. If not, the subroutine solves the equations and returns the calculated values to the main program. The most serious limitations of this program are imposed by the generation or solution of the simultaneous equations, therefore it is appropriate to include and explain several of them at this point.

First, at least one reaction, even if it is later forced to equal zero, must be included with the input, or a singular matrix will be generated.

Including shear deflection in the solution must also be greatly limited at this point. Since in general the shear includes step discontinuities, they will also appear in the slope if shear deflection is included. The program generates an equation for slope at each moment reaction. If by chance, either a reaction or point force occurs at the same point as a moment reaction, the slope at this point is double valued. The program is unable to distinguish which of the two values is required except when it occurs at the extreme ends of the beam. For example a beam "built in" at both ends is acceptable. If the beam is "built in" at only one end, it must be solved in a manner such that the "built in" is at the origin of the beam.

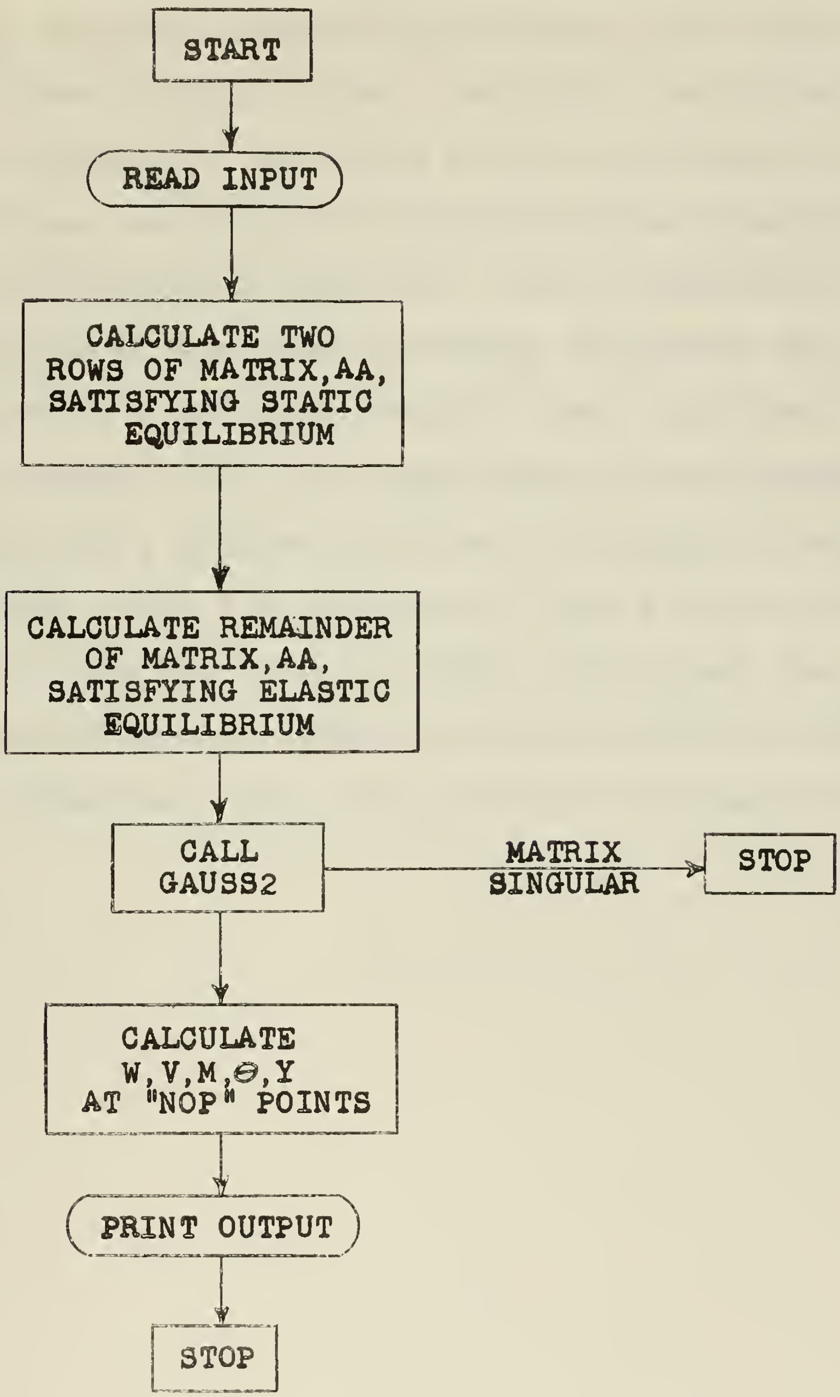
The program then divides the beam into a number of increments, NOP-1,* and calls the subroutines, LOAD, SHEAR,

*NOP is another parameter read in at the start of BEAM3.

MOMENT, SLOPE and DEFLECT to calculate the flexural quantities at each of these increments.

The output consists of printing the distance from the left end of the beam to the point in question and the values of the load, shear, moment, slope and deflection at that point. The values of all of the indeterminate quantities are also printed out.

Fig. 7 Simplified Flow Chart of BEAM3



7. Testing of Program.

The testing of the program and the subroutines could, in theory, continue indefinitely because of the unlimited number of beam configurations. The author has attempted to show (See Appendix D) solutions to test problems of simple configurations and constant bending and shear compliance to compare with classical solutions. Then a statically determinate problem with a varying bending compliance was solved and compared to a solution obtained from a numerical integration scheme. The most rigid test for the program was in constructing a problem for a beam of unlikely configuration (See Problems 6 and 7 of Appendix D), and solving the flexural analysis of this beam from both ends. Note that the only significant difference between the two solutions is the opposite signs for shear and slope, which would of course, be expected.

8. Conclusions and Recommendations.

The thesis shows a method of including piecewise linear variation of both bending compliance and shear compliance in the flexural analysis of beams by extending "McCaulay's Method". Adapting this idea to the digital computer resulted in a program capable of analyzing a very general type of beam. However, had time permitted, several changes would have been made to the program. First, internal monitoring of the input data would be an asset to any user. Also, using the same general method of attack, this program would have been expanded to include bending compliance and shear compliance of a piecewise parabolic nature.

Other additions to the program, much larger in scope, would be including elastic supports and/or beam column effect.

It would be interesting to compare the results of this program to those obtained from a program utilizing the numerical integration method. Also, a more exhaustive study of the program could be made to determine accuracy of, and time requirement for, solutions.

Bibliography

1. McCaulay, W. H., Note on the Deflections of Beams, Messenger of Mathematics, Vol XLVIII-No. 9, Jan. 1919.

2. Anthony, C. B., Digital Computer Analysis of Rigid Body Problems, Thesis, United States Naval Postgraduate School, 1963.

3. Quinlan, P. M., Elastic Beam Theory for the Electronic Computer--A Recurrence Approach, Report submitted to the Air Research and Development Command (European Office) of the United States Air Force under contract no. AF61(514)-1163, Technical note No. 9, June 1960.

4. DenHartog, J. P., Strength of Materials, McGraw-Hill Book Company, Inc., New York, 1949.

5. Timoshenko, S., Strength of Materials, Third Edition, D. Van Nostrand Company, Inc., New York, 1955.

6. Crandall, S. H. and Dahl, N. C., An Introduction to the Mechanics of Solids, McGraw-Hill Bood Company, Inc., New York, 1959.

7. CO OP Manual, Publication 067A, Vol. I, Control Data Corporation Computer Division, Minneapolis, Minnesota, Dec. 1960.

8. Brock, J. E. and Newton, R. E., A Method of Pedagogical Value in Elementary Bending Theory, Civ. Eng. Bul, ASEE, 17, 1952.

9. Hetenyi, M., Deflection of Beams of Varying Cross Section, Journal of Applied Mechanics, 4, 1937.

APPENDIX A

"Curly bracket" method, definitions and formulas

We define:

1. $\{x-a\}^n = (x-a)^n \quad , \quad x > a$

$\quad\quad\quad = 0 \quad , \quad x < a$

where n is an integer $\geqq 0$

From this definition, it is possible to establish the following formulas. Although it would be possible to express these results in terms of curly brackets, in most cases the resulting expressions would be rather complicated. The forms exhibited below have the advantage of being most readily conformable to the decision commands available in FORTRAN.

2. $\int_0^x \{x-a\}^n dx = \frac{(x-a)^{n+1}}{n+1} \quad , \quad x > a$

$\quad\quad\quad = 0 \quad , \quad \text{Otherwise}$

3. $\{x-a\}^0 \{x-b\}^n = (x-b)^n \quad , \quad x > a \text{ and } x > b$

$\quad\quad\quad = 0 \quad , \quad \text{Otherwise}$

4. $\{x-a\}^1 \{x-b\}^n = (x-b)^{n+1} + (b-a)(x-b)^n \quad , \quad x > a \text{ and } x > b$

$\quad\quad\quad = 0 \quad , \quad \text{Otherwise}$

5. $\int_0^x \{x-a\}^0 \{x-b\}^n dx = \frac{(x-b)^{n+1}}{n+1} - \frac{(a-b)^{n+1}}{n+1} \quad , \quad x > a > b$

$\quad\quad\quad = \frac{(x-b)^{n+1}}{n+1} \quad , \quad x > b \geq a$

$\quad\quad\quad = 0 \quad , \quad \text{Otherwise}$

6. $\int_0^x \{x-a\}^1 \{x-b\}^n dx$

$$= \frac{(x-b)^{n+2}}{n+2} + \frac{(b-a)(x-b)^{n+1}}{n+1} + \frac{(a-b)^{n+2}}{(n+1)(n+2)} \quad , \quad x > a > b$$

$$= \frac{(x-b)^{n+2}}{n+2} + \frac{(b-a)(x-b)^{n+1}}{n+1} \quad , \quad x > b > a$$

$$= 0 \quad , \quad \text{Otherwise}$$

7. $\int_0^x \int_0^x \{\xi - a\}^0 \{\xi - b\}^n d\xi \, dx$

$$= \frac{(x-b)^{n+2}}{(n+1)(n+2)} - \frac{(x-a)(a-b)^{n+1}}{(n+1)} - \frac{(a-b)^{n+2}}{(n+1)(n+2)} \quad , \quad x > a > b$$

$$= \frac{(x-b)^{n+2}}{(n+1)(n+2)} \quad , \quad x > b \geq a$$

$$= 0 \quad , \quad \text{Otherwise}$$

8. $\int_0^x \int_0^x \{\xi - a\}^1 \{\xi - b\}^n d\xi \, dx$

$$= \frac{(x-b)^{n+3}}{(n+2)(n+3)} + \frac{(b-a)(x-b)^{n+2}}{(n+1)(n+2)} + \frac{(x-a)(a-b)^{n+2}}{(n+1)(n+2)} + \frac{2(a-b)^{n+3}}{(n+1)(n+2)(n+3)} \quad , \quad x > a > b$$

$$= \frac{(x-b)^{n+3}}{(n+2)(n+3)} + \frac{(b-a)(x-b)^{n+2}}{(n+1)(n+2)} \quad , \quad x > b > a$$

$$= 0 \quad , \quad \text{Otherwise}$$

Function Subroutines FTON, FT1N, ENTON, ENT1N, DITON, DIT1N

The above named function subroutines were written and utilized to evaluate each of the preceeding formulas numbered 3 through 8. The values x, a, b and n were used as entering arguments to these functions. The function then became the variable as follows:

$$\text{FTON} = \{x-a\}^{0}\{x-b\}^{n}$$

$$\text{FT1N} = \{x-a\}^{1}\{x-b\}^{n}$$

$$\text{ENTON} = \int_{0}^{x} \{x-a\}^{0}\{x-b\}^{n}dx$$

$$\text{ENT1N} = \int_{0}^{x} \{x-a\}^{1}\{x-b\}^{n}dx$$

$$\text{DITON} = \int_{0}^{x}\int_{0}^{x} \{\xi-a\}^{0}\{\xi-b\}^{n}d\xi\,dx$$

$$\text{DIT1N} = \int_{0}^{x}\int_{0}^{x} \{\xi-a\}^{1}\{\xi-b\}^{n}d\xi\,dx$$

As examples, flow charts for FTON and ENTON are shown in Figs. 8 and 9.

Fig. 8 Flow Chart of FTON

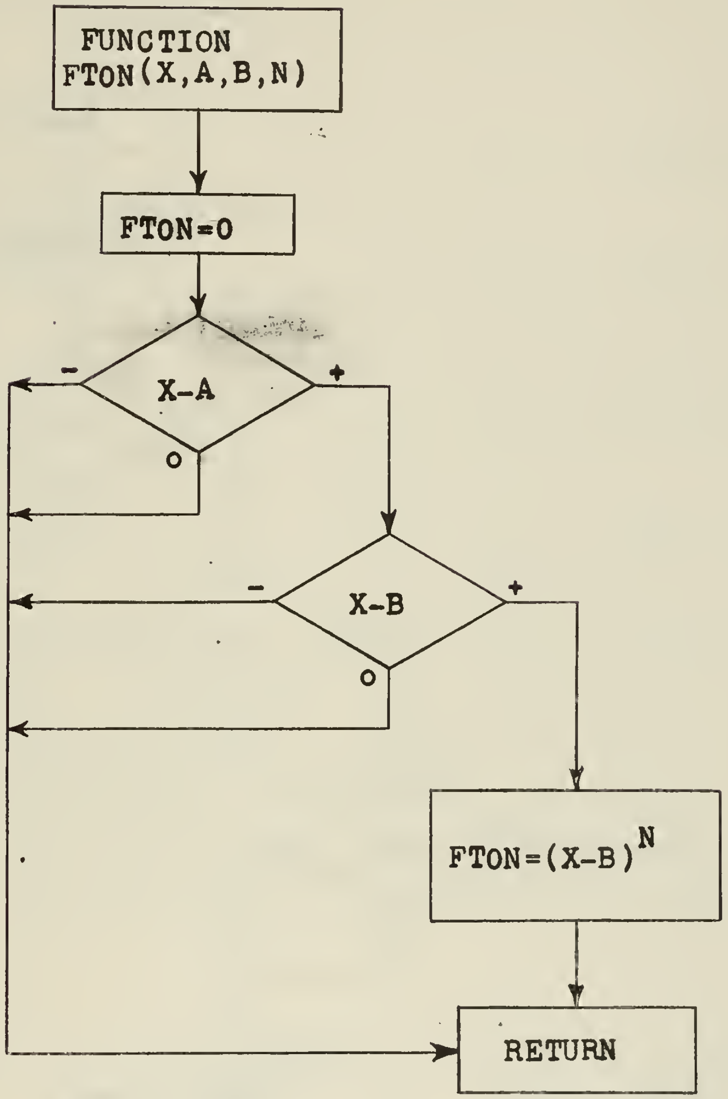
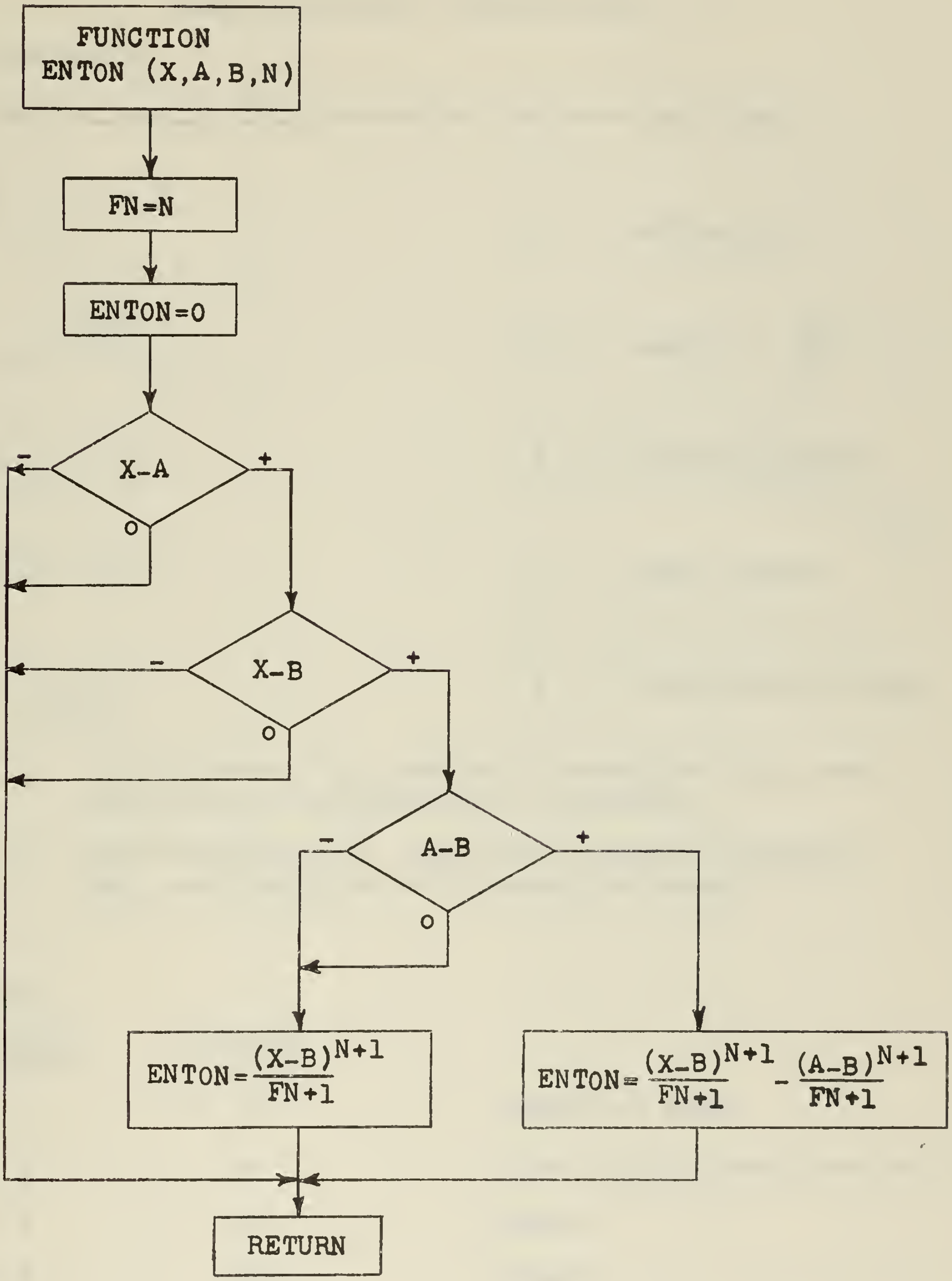


Fig. 9 Flow Chart of ENTON

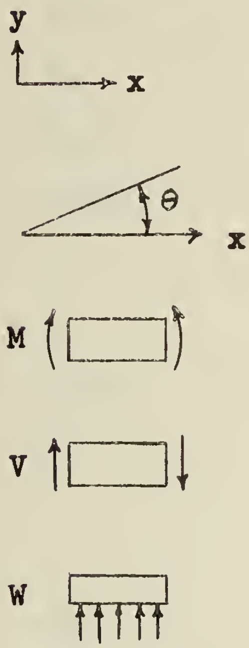


APPENDIX B

Sign Convention and Notation

Sign Convention

The following sign convention has been employed:



y = deflection

θ = slope = $\frac{dy}{dx}$

M = bending moment

V = shear force

W = distributed loads

Note: 1. Point forces, F_y, and point reactions, R_y, are positive in the positive y direction.

2. Point moments, M_z, and point reactive moments, RM_2, are positive in a clockwise sense.

Notation:

THESIS	PROGRAM	
	C	Length of beam
W	W	Total distributed load at a point
V	V	Shear
M	AM	Moment
θ	Y1	Slope
y	Y	Deflection

F_y	FY	Point load or force
R_y	RY	Point reaction
M_z	AMZ	Point moment
RM_z	RAMZ	Point reactive moment
Q(x)		Element of distributed load
Q_a	AQ	Distributed load (left end)
Q_b	QB	Distributed load (right end)
f(x)		Element of bending compliance $(\frac{1}{EI})$
	EA	Bending compliance (left end of element)
	EB	Bending compliance (right end of element)
g(x)		Element of shear compliance $(\frac{K}{AG})$
	GA	Shear compliance (left end of element)
	GB	Shear compliance (right end of element)
X_f	XFO	Distance to F_y
X_r	XR	Distance to R_y
X_m	XM	Distance to M_z
X_m	RXM	Distance to RM_z
A_g	AG	Distance to G_a
B_g	BG	Distance to G_b
A_e	AE	Distance to E_a
B_e	BE	Distance to E_b
A	A	Distance to Q_a
B	B	Distance to Q_b
NOL	NOL	Number of distributed loads (25)[3]

[3]Numbers following explanation of terms NOL--NOK indicate the arbitarily chosen maximum value of each for this program.

NOF	NOF	Number of point forces (100)
NOR	NOR	Number of point reactions (10)
NOM	NOM	Number of point moments (50)
NORM	NORM	Number of point reactive moments (10)
NOI	NOI	Number of bending compliance elements (25)
NOK	NOK	Number of shear compliance elements (25)
PDEFT	PDEFT	Predetermined deflection at a reaction
PSLP	PSLP	Predetermined slope at a reactive moment
NOP	NOP	Number of points at which the flexural quantities will be calculated.

APPENDIX C

General Information Concerning Use of Program

The composite program, BEAM3 plus subroutines, was written in FORTRAN-60, compiled and tested on a Control Data Corporation 1604 digital computer. It consists of 412 FORTRAN statements requiring about 580 cards. The computer storage requirement is about 12,500 cells, however, this number can be reduced by changing the dimension statements which would of course, reduce either the maximum number of external loads, elements of shear compliance or bending compliance, or number of points at which flexural quantities could be calculated.

The FORTRAN statements used in this program are: GO TO, computed GO TO, IF, STOP, DO, CONTINUE, FORMAT, READ, PRINT, WRITE OUTPUT TAPE, FUNCTION, SUBROUTINE, RETURN, CALL, DIMENSION, COMMON, and END.

There are no conversion constants incorporated in this program. As such all input must be in a consistant set of units.

"FORMAT" for Input Data

The input data for a problem is read by the program with a FORMAT of I4 for all "fixed point" variables and F20.0 for all floating point variables. The arrangement of the data on the input cards is shown as follows:

Group no.	No. of cards in group	Fields	Entries
1	1	F20.0	C
2	1	8I4	NOP, NOL, NOF, NOR NOM, NORM, NOI, NOK
3	NOL	4F20.0	A, B, A_a, A_b
4	NOF	2F20.0	x_f, F_y
5	NOR	3F20.0	x_r, 0.0, PDEFT
6	NOM	2F20.0	x_m, M_z
7	NORM	3F20.0	x_m, 0.0, PSLP
8	NOI	4F20.0	A_e, B_e, E_a, E_b
9	NOK	4F20.0	A_g, B_g, G_a, G_b

As an example the input data cards for test problem #1 in Appendix D was as follows:

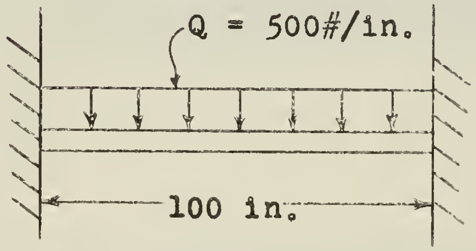
CARD 1:	100							
CARD 2:	21	1	0	2	0	2	1	0
CARD 3:	0.	100	-500.	-500.				
CARD 4:	0.	0.	.0					
CARD 5:	100	0.	.0					
CARD 6:	0.	0.	0.					
CARD 7:	100	0.	0.					
CARD 8:	0.	100.	.000000001	.000000001				

APPENDIX D

Test problems and solutions

Test problem 1

This problem consists of a beam of uniform cross-section, "built-in" at both ends, and loaded with a uniformly distributed weight of 500#/in. (See Fig. 10)



$$\frac{1}{EI} = 10^{-9} \frac{1}{\#\text{in.}^2}$$

$$\frac{K}{AG} = 0$$

Fig. 10

Test problem 2

This problem is identical to problem 1 except that the slope at either end of the beam is forced to values other than zero.

$$\theta_{\text{left end}} = -.002$$

$$\theta_{\text{right end}} = .002$$

SOLUTION TO TEST PROBLEM 1

DISTANCE	LOAD	SHEAR	MOMENT	SLOPE	DEFLECTION
.00	-500.00	25000.0	-416667.	-.00000000	.00000000
5.00	-500.00	22500.0	-297917.	-.00178125	-.00470052
10.00	-500.00	20000.0	-191667.	-.00300000	-.01687500
15.00	-500.00	17500.0	-97917.	-.00371875	-.03386719
20.00	-500.00	15000.0	-16667.	-.00400000	-.05333333
25.00	-500.00	12500.0	52083.	-.00390625	-.07324219
30.00	-500.00	10000.0	108333.	-.00350000	-.09187500
35.00	-500.00	7500.0	152083.	-.00284375	-.10782552
40.00	-500.00	5000.0	183333.	-.00200000	-.12000000
45.00	-500.00	2500.0	202083.	-.00103125	-.12761719
50.00	-500.00	-.0	208333.	.00000000	-.13020833
55.00	-500.00	-2500.0	202083.	.00103125	-.12761719
60.00	-500.00	-5000.0	183333.	.00200000	-.12000000
65.00	-500.00	-7500.0	152083.	.00284375	-.10782552
70.00	-500.00	-10000.0	108333.	.00350000	-.09187500
75.00	-500.00	-12500.0	52083.	.00390625	-.07324219
80.00	-500.00	-15000.0	-16667.	.00400000	-.05333333
85.00	-500.00	-17500.0	-97917.	.00371875	-.03386719
90.00	-500.00	-20000.0	-191667.	.00300000	-.01687500
95.00	-500.00	-22500.0	-297917.	.00178125	-.00470052
100.00	-500.00	-25000.0	-416667.	.00000000	.00000000

RY(1)= 25000.
RY(2)= 25000.
RAMZ(1)= -416667.
RAMZ(2)= 416667.
C1= .00000000
C2= .00000000

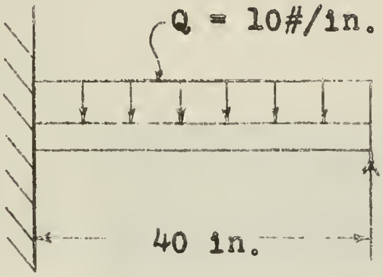
SOLUTION TO TEST PROBLEM 2

DISTANCE	LOAD	SHEAR	MOMENT	SLCPE	DEFLECTION
.00	-500.00	25000.0	-376667.	-.00200000	.00000000
5.00	-500.00	22500.0	-257917.	-.00358125	-.01420052
10.00	-500.00	20000.0	-151667.	-.00460000	-.03487500
15.00	-500.00	17500.0	-57917.	-.00511875	-.05936719
20.00	-500.00	15000.0	23333.	-.00520000	-.08533333
25.00	-500.00	12500.0	92083.	-.00490625	-.11074219
30.00	-500.00	10000.0	148333.	-.00430000	-.13387500
35.00	-500.00	7500.0	192083.	-.00344375	-.15332552
40.00	-500.00	5000.0	223333.	-.00240000	-.16800000
45.00	-500.00	2500.0	242083.	-.00123125	-.17711719
50.00	-500.00	-.0	248333.	.00000000	-.18020833
55.00	-500.00	-2500.0	242083.	.00123125	-.17711719
60.00	-500.00	-5000.0	223333.	.00240000	-.16800000
65.00	-500.00	-7500.0	192083.	.00344375	-.15332552
70.00	-500.00	-10000.0	148333.	.00430000	-.13387500
75.00	-500.00	-12500.0	92083.	.00490625	-.11074219
80.00	-500.00	-15000.0	23333.	.00520000	-.08533333
85.00	-500.00	-17500.0	-57917.	.00511875	-.05936719
90.00	-500.00	-20000.0	-151667.	.00460000	-.03487500
95.00	-500.00	-22500.0	-257917.	.00358125	-.01420052
100.00	-500.00	-25000.0	-376667.	.00200000	.00000000

RY(1)= 25000.
RY(2)= 25000.
RAMZ(1)= -376667.
RAMZ(2)= 376667.
C1=-.00200000
C2= .00000000

Test problem 3

This problem consists of a beam of uniform cross-section, "built-in" at one end and simply supported at the other, and loaded with a uniformly distributed weight. (See Fig. 11)



$$\frac{1}{EI} = .1332 \times 10^{-7} \frac{1}{\#in.^2}$$

$$\frac{K}{AG} = 0$$

Fig. 11

Test problem 4

This problem is identical to problem number 3 except shear deflection is included in the solution.

$$\frac{K}{AG} = .159 \times 10^{-6} \ 1/\#in.$$

SOLUTION TO TEST PROBLEM 3

DISTANCE	LOAD	SHEAR	MOMENT	SLOPE	DEFLECTION
.00	-10.00	250.00	-2000.00	-.00000000	.00000000
2.00	-10.00	230.00	-1520.00	-.00004680	-.00004893
4.00	-10.00	210.00	-1080.00	-.00008134	-.00017902
6.00	-10.00	190.00	-680.00	-.00010470	-.00036683
8.00	-10.00	170.00	-320.00	-.00011793	-.00059105
10.00	-10.00	150.00	.00	-.00012210	-.00083250
12.00	-10.00	130.00	280.00	-.00011828	-.00107412
14.00	-10.00	110.00	520.00	-.00010754	-.00130101
16.00	-10.00	90.00	720.00	-.00009093	-.00150036
18.00	-10.00	70.00	880.00	-.00006953	-.00166154
20.00	-10.00	50.00	1000.00	-.00004440	-.00177600
22.00	-10.00	30.00	1080.00	-.00001661	-.00183736
24.00	-10.00	10.00	1120.00	.00001279	-.00184136
26.00	-10.00	-10.00	1120.00	.00004271	-.00178586
28.00	-10.00	-30.00	1080.00	.00007211	-.00167086
30.00	-10.00	-50.00	1000.00	.00009990	-.00149850
32.00	-10.00	-70.00	880.00	.00012503	-.00127304
34.00	-10.00	-90.00	720.00	.00014643	-.00100086
36.00	-10.00	-110.00	520.00	.00016304	-.00069051
38.00	-10.00	-130.00	280.00	.00017378	-.00035262
40.00	-10.00	-150.00	.00	.00017760	-.00000000

RY(1)= 250.00
RY(2)= 150.00
RAMZ(1)= -2000.00
C1= .00000000
C2= .00000000

SOLUTION TO TEST PROBLEM 4

DISTANCE	LOAD	SHEAR	MOMENT	SLOPE	DEFLECTION
.00	-10.00	248.91	-1956.22	-.00003958	.00000000
2.00	-10.00	228.91	-1478.41	-.00008206	-.00012375
4.00	-10.00	208.91	-1040.59	-.00011234	-.00032009
6.00	-10.00	188.91	-642.78	-.00013149	-.00056570
8.00	-10.00	168.91	-284.97	-.00014058	-.00083936
10.00	-10.00	148.91	32.84	-.00014067	-.00112203
12.00	-10.00	128.91	310.65	-.00013283	-.00139677
14.00	-10.00	108.91	548.46	-.00011812	-.00164877
16.00	-10.00	88.91	746.27	-.00009760	-.00186536
18.00	-10.00	68.91	904.08	-.00007235	-.00203602
20.00	-10.00	48.91	1021.89	-.00004343	-.00215232
22.00	-10.00	28.91	1099.70	-.00001190	-.00220799
24.00	-10.00	8.91	1137.51	.00002117	-.00219889
26.00	-10.00	-11.09	1135.32	.00005471	-.00212300
28.00	-10.00	-31.09	1093.14	.00008766	-.00198044
30.00	-10.00	-51.09	1010.95	.00011896	-.00177345
32.00	-10.00	-71.09	888.76	.00014753	-.00150642
34.00	-10.00	-91.09	726.57	.00017232	-.00118585
36.00	-10.00	-111.09	524.38	.00019225	-.00082039
38.00	-10.00	-131.09	282.19	.00020626	-.00042080
40.00	-10.00	-151.09	.00	.00021329	-.00000000

```
RY( 1)=     248.91
RY( 2)=     151.09
RAMZ( 1)=  -1956.22
C1=-.00000000
C2= .00000000
```

Test problem 5

This problem consists of a homogeneous circular shaft, three inches in diameter at the center and tapered to one and a half inches in diameter at both ends. The total weight of the shaft is 100# and a uniformly distributed load totaling 200# is placed on the central section. The shaft is supported by a uniformly distributed reaction at each end. Points A and B are points of zero deflection. This problem was approximated for the program as shown in Fig. 12.

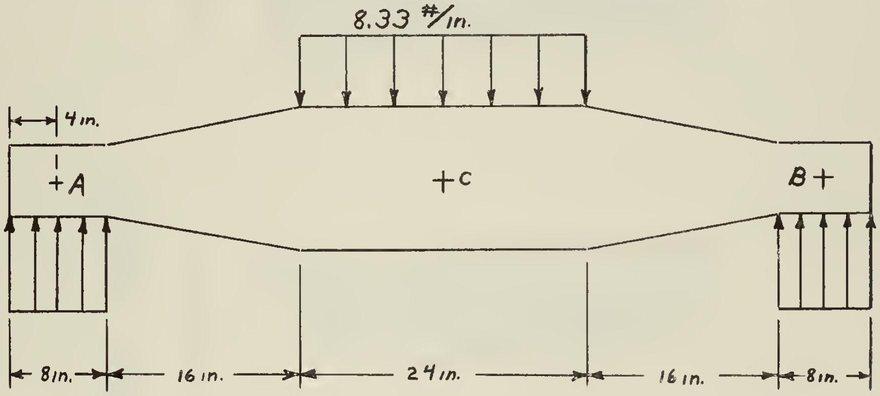


Fig. 12

Weight of shaft:

0" - 8"	-0.5315#/in.	
8" - 24"	-0.5315 to -2.081#/in.	(linear)
24" - 36"	-2.081#/in.	

Bending compliance $\left(\frac{1}{EI}\right)$:

0" - 8"	0.1420×10^{-6}	$1/\#\text{-in}^2$		
8" - 12"	0.1420×10^{-6} to 0.0583×10^{-6}	$1/\#\text{-in}^2$	(linear)
12" - 16"	0.0583×10^{-6} to 0.0281×10^{-6}	$1/\#\text{-in}^2$	(linear)
16" - 20"	0.0281×10^{-6} to 0.0152×10^{-6}	$1/\#\text{-in}^2$	(linear)
20" - 24"	0.0152×10^{-6} to 0.0089×10^{-6}	$1/\#\text{-in}^2$	(linear)
24" - 36"	0.0089×10^{-6}	$1/\#\text{-in}^2$		

Shear compliance $\left(\frac{K}{AG}\right) = 0.$

SOLUTION TO TEST PROBLEM 5

DISTANCE	LOAD	SHEAR	MOMENT	SLOPE	DEFLECTION
.00	18.22	.00	.00	-.00144040	.00573400
4.00	18.22	72.96	145.75	-.00141280	-.00000000
8.00	-.53	145.83	583.33	-.00121954	-.00534754
12.00	-.92	142.93	1161.37	-.00088582	-.00953806
16.00	-1.31	138.48	1724.71	-.00064188	-.01256778
20.00	-1.69	132.48	2267.15	-.00047103	-.01477502
24.00	-10.41	124.93	2782.50	-.00035015	-.01640427
28.00	-10.41	83.29	3198.94	-.00024330	-.01759611
32.00	-10.41	41.64	3448.80	-.00012461	-.01833491
36.00	-10.41	-.00	3532.09	-.00000000	-.01858513
40.00	-10.41	-41.64	3448.80	.00012461	-.01833491
44.00	-10.41	-83.29	3198.94	.00024330	-.01759611
48.00	-2.08	-124.93	2782.50	.00035015	-.01640427
52.00	-1.69	-132.48	2267.15	.00047103	-.01477502
56.00	-1.31	-138.48	1724.71	.00064188	-.01256778
60.00	-.92	-142.93	1161.37	.00088582	-.00953806
64.00	18.22	-145.83	583.33	.00121954	-.00534754
68.00	18.22	-72.87	145.75	.00141280	.00000000
72.00	18.22	-.00	-.00	.00144040	.00573400

RY(1)= .08
RY(2)= .08
C1=-.00144040
C2= .00573400

Test problem 6

This hyperstatic problem consists of a beam with a very unlikely configuration as shown in Fig. 13. The bending compliance varies as shown in Fig. 14. Shear deflection has been omitted, i.e., $\frac{K}{AG} = 0$.

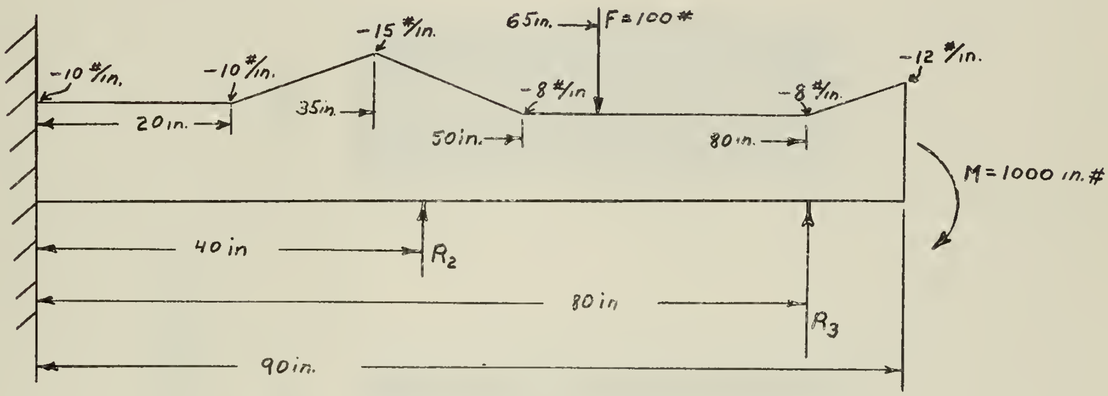


Fig. 13

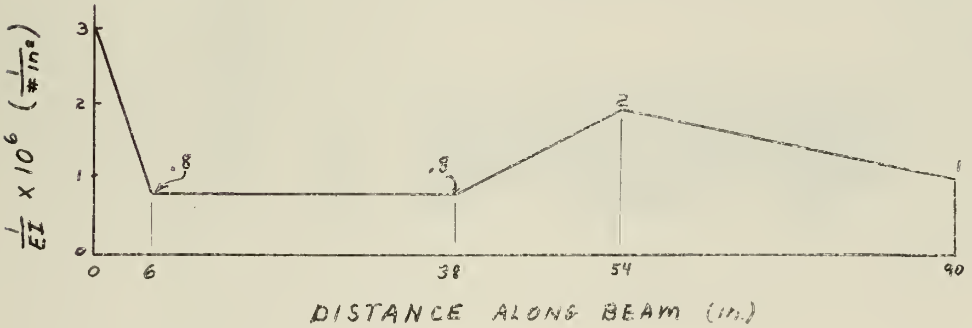


Fig. 14

Test problem 7

This problem is identical to problem number 6 except the beam was turned end for end.

SOLUTION TO TEST PROBLEM 6

DISTANCE	LOAD	SHEAR	MOMENT	SLOPE	DEFLECTION
.00	-10.00	218.61	-1672.36	-.00000000	.00000000
4.50	-10.00	173.61	-789.88	-.00576422	-.01429420
9.00	-10.00	128.61	-109.89	-.00804624	-.04663760
13.50	-10.00	83.61	367.59	-.00716320	-.08194572
18.00	-10.00	38.61	642.57	-.00377395	-.10733002
22.50	-10.83	-7.43	714.18	.00112008	-.11363611
27.00	-12.33	-59.56	565.97	.00612216	-.09705796
31.50	-13.83	-118.43	168.01	.00927688	-.06127469
36.00	-14.53	-183.66	-509.94	.00796860	-.02022418
40.50	-12.43	208.18	-1250.21	.00018866	.00036344
45.00	-10.33	156.95	-432.21	-.00544019	-.01398517
49.50	-8.23	115.18	176.53	-.00610080	-.04120624
54.00	-8.00	79.12	613.58	-.00456710	-.06571866
58.50	-8.00	43.12	888.61	-.00181456	-.08044870
63.00	-8.00	7.12	1001.65	.00163650	-.08100193
67.50	-8.00	-128.88	702.68	.00495289	-.06575553
72.00	-8.00	-164.88	41.72	.00634141	-.03945104
76.50	-8.00	-200.88	-781.25	.00505885	-.01268946
81.00	-8.40	91.80	-1437.40	.00065355	.00124103
85.50	-10.20	49.95	-1115.42	-.00435623	-.00670383
90.00	-12.00	.00	-1000.00	-.01445347	-.04653770

RY(1)= 218.61
RY(2)= 452.51
RY(3)= 328.88
RAMZ(1)= -1672.36
C1= .00000000
C2= .00000000

SOLUTION TO TEST PROBLEM 7

DISTANCE	LOAD	SHEAR	MOMENT	SLOPE	DEFLECTION
.00	-12.00	.00	-1000.00	.01445346	-.04653768
4.50	-10.20	-49.95	-1115.42	.00435623	-.00670377
9.00	-8.40	-91.80	-1437.40	-.00065355	.00124103
13.50	-8.00	200.88	-781.25	-.00505885	-.01268946
18.00	-8.00	164.88	41.72	-.00634141	-.03945104
22.50	-8.00	128.88	702.68	-.00495289	-.06575552
27.00	-8.00	-7.12	1001.65	-.00163650	-.08100193
31.50	-8.00	-43.12	888.61	.00181456	-.08044870
36.00	-8.00	-79.12	613.58	.00456710	-.06571865
40.50	-8.23	-115.18	176.53	.00610080	-.04120624
45.00	-10.33	-156.95	-432.21	.00544019	-.01398517
49.50	-12.43	-208.18	-1250.21	-.00018866	.00036344
54.00	-14.53	183.66	-509.94	-.00796860	-.02022419
58.50	-13.83	118.43	168.01	-.00927688	-.06127469
63.00	-12.33	59.56	565.97	-.00612216	-.09705796
67.50	-10.83	7.43	714.18	-.00112008	-.11363612
72.00	-10.00	-38.61	642.57	.00377395	-.10733003
76.50	-10.00	-83.61	367.59	.00716320	-.08194571
81.00	-10.00	-128.61	-109.89	.00804624	-.04663761
85.50	-10.00	-173.61	-789.88	.00576422	-.01429420
90.00	-10.00	-218.61	-1672.36	.00000000	.00000002

RY(1)= 328.88
RY(2)= 452.51
RY(3)= 218.61
RAMZ(1)= 1672.36
C1= .01445346
C2=-.04653768

APPENDIX E

Subroutine GAUSS2

This routine is a modified form of the GAUSS2 used by C. B. Bailey /7/ in his program, LINEQN, for solving simultaneous linear equations by Gaussian elimination. Bailey was solving the matrix equation, Ax=B, with a possibility of 50 vectors B for each matrix A, but this program requires only one vector B with each matrix A. Also since the maximum number of equations in BEAM3 is 22, the size of matrix A was reduced from 100x101 to 22x23.

At each step of the elimination, the value of the diagonal element is compared to a defined "zero". If it is smaller than this quantity, the matrix is considered singular and an error output is returned to the calling program. This is the "matrix singular" test used by BEAM3.

APPENDIX F

Listing of program and subroutines

```
      PROGRAM BEAM3
      DIMENSION W(1001),V(1001),AM(1001),Y1(1001),Y(1001),XX(1001),
     1 A(25),B(25),QA(25),QB(25),XFO(100),FY(100),XR(10),RY(10),
     2XM(50),AMZ(50),RXM(10),RAMZ(10),AE(25),BE(25),EA(25),EB(25),
     3AA(22,23),XA(22),PDEFT(10),PSLP(10)
     4 ,ITITLE(10),GA(25),GB(25),AG(25),BG(25)
      COMMON A,B,QA,QB,XFO,FY,XR,RY,XM,AMZ,RXM,RAMZ,AE,BE,EA,EB,NOL,
     1 NOR,NOF,NOM,NORM,NOI,NOK,GA,GB,AG,BG
      READ  70,C
   70 FORMAT (F20.0)
      READ 71,NOP,NOL,NOF,NOR,NOM,NORM,NOI,NOK
   71 FORMAT(8I4)
      IF(NOL) 51,51,50
   50 READ 72,(A(I),B(I),QA(I),QB(I),I=1,NOL)
   72 FORMAT(4F20.0)
   51 IF(NOF)53,53,52
   52 READ 73,( XFO(N),FY(N),N=1,NOF)
   73 FORMAT (2F20.0)
   53 IF(NOR)55,55,54
   54 READ 74,(XR(M),RY(M),PDEFT(M),M=1,NOR)
   74 FORMAT (3F20.0)
   55 IF(NOM)57,57,56
   56 READ 73,(XM(K),AMZ(K),K=1,NOM)
   57 IF(NORM)59,59,58
   58 READ 74,(RXM(KA),RAMZ(KA),PSLP(KA),KA=1,NORM)
   59  IF(NOI)61,61,60
   60 READ 72,(AE(IA),BE(IA),EA(IA),EB(IA),IA=1,NOI)
   61  IF(NOK)1,1,62
   62  READ 72,(AG(IB),BG(IB),GA(IB),GB(IB),IB=1,NOK)
    1 NOR1 = NOR+1
      C1=0.
      C2= 0.
      NORM2 =NORM + 2
      NORM3 =NORM + 3
      NOQ = NOR +NORM
      NOQ1 = NOQ + 1
      NOQ2 = NOQ + 2
```

```
      NOQ3 = NOQ + 3
      EPSX=10.**(-8)
       DO 2 K=1,NOR
  2    AA(1,K) = 1.
       DO 3 K= NOR1,NOQ2
  3    AA(1,K) = C.
       AAA=0.
      IF(NOF)45,45,35
 35   DO 4 N=1,NOF
  4    AAA= AAA + FY(N)
 45   IF(NOL)48,48,46
 46   DO 5 I=1,NOL
  5    AAA = AAA + ((QA(I)+QB(I))/2.)* (B(I)-A(I))
 48   AA(1,NOQ3)=-AAA
       DO 6 K =1,NOR
  6    AA(2,K) = C-XR(K)
       IF(NORM)9,9,7
  7    DO  8 K=NOR1,NOQ
  8   AA(2,K)=1.
  9    DO 10  K=NOQ1,NOQ2
 10    AA(2,K) = 0.
       CALL MOMENT (C,AAA)
       AA(2,NOQ3) =-AAA
       IF (NORM)17,17,11
 11    DO 16 J=3,NORM2
       DO 13 K=1,NOR
       AA(J,K)=0.
       DO 12 IA=1,NOI
      AAA=EA(IA)*ENTON(RXM(J-2),AE(IA),XR(K),1) - EB(IA)*ENTON(RXM(J-2),
     1BE(IA),XR(K),1) +((EB(IA)-EA(IA))/(BE(IA)-AE(IA)))*(ENT1N(RXM(J-2)
     2,AE(IA),XR(K),1)-ENT1N(RXM(J-2),BE(IA),XR(K),1))
 12    AA(J,K) = AA(J,K) + AAA
      IF (NOK) 13,13,900
900   DO 910  IB=1,NOK
      IF(NORM-1)902,901,902
902   IF(J-NORM2)901,904,901
901   X=RXM(J-2) -EPSX
```

```
      AAA = GA(IB)* FTON(  X      ,AG(IB),XR(K),0) - GB(IB)* FTON(X
     1     ,BG(IB),XR(K),0) + ((GB(IB)-GA(IB))/(BG(IB)-AG(IB)))* ( FT1N(
     2  X      ,AG(IB),XR(K),0)- FT1N(  X      ,BG(IB),XR(K),0))
      GO TO 910
  904 X=RXM(J-2) +EPSX
      AAA = GA(IB)* FTON(  X      ,AG(IB),XR(K),0) - GB(IB)* FTON(X
     1     ,BG(IB),XR(K),0) + ((GB(IB)-GA(IB))/(BG(IB)-AG(IB)))* ( FT1N(
     2  X      ,AG(IB),XR(K),0)- FT1N(  X      ,BG(IB),XR(K),0))
  910 AA(J,K) =AA(J,K) -AAA
   13  CONTINUE
      IF(NOQ-NOR1)155,135,135
  135 DO 15 K=NOR1,NOQ
       AA(J,K)= 0.
       DO 14 IA=1,NOI
      AAA= EA(IA)*ENTON(RXM(J-2),AE(IA),RXM(K-NOR),0) -EB(IA)* ENTON
     1 (RXM(J-2),BE(IA),RXM(K-NOR),0) + ((EB(IA)-EA(IA))/(BE(IA)-AE(IA))
     2) *(ENT1N(RXM(J-2),AE(IA),RXM(K-NOR),0) - ENT1N(RXM(J-2),BE(IA),
     3 RXM(K-NOR),0))
   14  AA(J,K) = AA(J,K) + AAA
   15  CONTINUE
  155 AA(J,NOQ1)=1.
       AA(J,NOQ2)= 0.
      IF(J-NORM2)156,157,156
  156 X=RXM(J-2) - EPSX
      CALL SLOPE(X,AAA)
      GO TO 16
  157 X=RXM(J-2) + EPSX
      CALL SLOPE(X,AAA)
   16  AA(J,NOQ3) =-AAA + PSLP(J-2)
   17  DO 20 J=NORM3,NOQ2
       DO 18 K=1,NOR
       AA(J,K) =0.
      DO 181  IA=1,NOI
       AAA = EA(IA)* DITON(XR(J-NORM2),AE(IA),XR(K),1) -EB(IA)*DITON
     1 (XR(J-NORM2),BE(IA),XR(K),1) +((EB(IA)-EA(IA))/ (BE(IA)-AE(IA)))
     2 *(DIT1N(XR(J-NORM2),AE(IA),XR(K),1) - DIT1N(XR(J-NORM2),BE(IA),
     3 XR(K),1))
```

```
181 AA(J,K) = AA(J,K) + AAA
    IF (NOK)18,18,920
920 DO 930  IB=1,NOK
    AAA =GA(IB)* ENTON(XR(J-NORM2),AG(IB),XR(K),0) - GB(IB)*ENTON
   1 (XR(J-NORM2),BG(IB),XR(K),0) +((GB(IB)-GA(IB))/(BG(IB)-AG(IB)))
   2 *(ENT1N(XR(J-NORM2),AG(IB) ,XR(K),0) -ENT1N(XR(J-NORM2),BG(IB),
   3 XR(K),0))
930 AA(J,K) =AA(J,K) -AAA
 18 CONTINUE
    IF(NOQ-NOR1)195,185,185
185 DO 19 K=NOR1,NOQ
     AA(J,K) = 0.
     DO 19 IA=1,NOI
     AAA = EA(IA)* DITON(XR(J-NORM2),AE(IA),RXM(K-NOR),0) - EB(IA)*
   1 DITON(XR(J-NORM2),BE(IA),RXM(K-NOR),0) +((EB(IA)-EA(IA)) / (BE(IA
   2 )- AE(IA))) * (DIT1N(XR(J-NORM2),AE(IA),RXM(K-NOR),0) - DIT1N
   3 (XR(J-NORM2),BE(IA),RXM(K-NOR),0))
 19  AA(J,K) = AA(J,K) + AAA
195 AA(J,NOQ1) =XR(J-NORM2)
     AA(J,NOQ2) = 1.
     CALL DEFLECT (XR(J-NORM2),AAA)
 20  AA(J,NOQ3) =-AAA+ PDEFT(J-NORM2)
     CALL GAUSS2(NOQ2,.00000001,AA,XA,K1)
     GO TO (22,21),K1
 21  WRITE OUTPUT TAPE 4,77
 77  FORMAT (16H MATRIX SINGULAR)
    STOP
 22  DO  23 K=1,NOR
 23  RY(K) = XA(K)
     IF(NORM)26,26,24
 24  DO  25 K=NOR1,NOQ
 25  RAMZ(K-NOR) = XA(K)
 26  C1 = XA(NOQ1)
     C2 = XA(NOQ2)
     X=0.
    NOPE =NOP-1
     FNOP = NOP-1
```

```
      DO  27  J=1,NCPE
       CALL LOAD (X,W(J))
       CALL SHEAR (X,V(J))
       CALL MOMENT (X,AM(J))
      X1 =X+EPSX
      CALL SLOPE(X1,Y1(J))
      Y1(J)=Y1(J) +C1
       CALL DEFLECT (X,Y(J))
      Y(J)=Y(J)+C1*X+C2
       XX(J) = X
 27    X = C/FNOP +X
      XX(NOP)=X
        X=X-3.*EPSX
      CALL LOAD(X,W(NOP))
      CALL SHEAR(X,V(NOP))
      CALL MOMENT(X,AM(NOP))
      CALL SLOPE(X,Y1(NOP))
      Y1(NOP)=Y1(NOP)+C1
      CALL DEFLECT(X,Y(NOP))
      Y(NOP)=Y(NOP)+C1*X +C2
      WRITE OUTPUT TAPE 4,76
 76   FORMAT ( 3X,1HJ,7X,4HLOAD,10X,5HSHEAR,10X,6HMOMENT,8X,5HSLOPE,
     1 9X,10HDEFLECTION,8X,1HX)
      WRITE OUTPUT TAPE 4,78,(J,W(J),V(J),AM(J),Y1(J),Y(J),XX(J),J=1,
     1NOP)
 78   FORMAT(I4,6E15.8)
      IF (NCR)282,282,280
 280  PRINT 281,(M,RY(M),M=1,NOR)
 281  FORMAT ( 3HRY(,I2,2H)=,E20.8)
 282  IF(NORM)288,288,283
 283  PRINT 284,(KA,RAMZ(KA),KA=1,NCRM)
 284  FORMAT (5HRAMZ(,I2,2H)=,E20.8)
 288  PRINT 286,C1
 286  FORMAT(5X,3HC1=,E20.8)
      PRINT 287,C2
 287  FORMAT(5X,3HC2=,E20.8)
 285  CONTINUE
```

```
STOP
END
```

```
      SUBROUTINE LOAD(X,W)
      DIMENSION A(25),B(25),QA(25),QB(25),XFO(100),FY(100),XR(10),RY(10)
     1,XM(50),AMZ(50),RXM(10),RAMZ(10),AE(25),BE(25),EA(25),EB(25)
     2,GA(25),GB(25),AG(25),BG(25)
      COMMON A,B,QA,QB,XFO,FY,XR,RY,XM,AMZ,RXM,RAMZ,AE,BE,EA,EB,NOL,
     1 NOR,NOF,NOM,NORM,NOI,NOK,GA,GB,AG,BG
      EPSX=10.**(-8)
      W=0.
    1 DO 5 I=1,NOL
    2 IF(X+EPSX-B(I))3,3,5
    3 IF(X+EPSX-A(I))5,4,4
    4 WW=QA(I)+((QB(I)-QA(I))*(X-A(I)))/(B(I)-A(I))
      W=W+WW
    5 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SHEAR(X,V)
      DIMENSION A(25),B(25),QA(25),QB(25),XFO(100),FY(100),XR(10),RY(10)
     1,XM(50),AMZ(50),RXM(10),RAMZ(10),AE(25),BE(25),EA(25),EB(25)
     2,GA(25),GB(25),AG(25),BG(25)
      COMMON A,B,QA,QB,XFO,FY,XR,RY,XM,AMZ,RXM,RAMZ,AE,BE,EA,EB,NOL,
     1 NOR,NOF,NOM,NORM,NOI,NOK,GA,GB,AG,BG
      EPSX=10.**(-8)
      V=0.
      IF(NOL)7,7,1
    1 DO 6 I=1,NOL
      IF(X-A(I))6,2,2
    2 IF(X+EPSX-B(I))4,4,3
    3 VV= QA(I)*(X-A(I) )+((QB(I)-QA(I))*((X-A(I))**2))/(2.*(B(I)-A(I)))
     1-QB(I)*(X-B(I)) -((QB(I)-QA(I))*((X-B(I))**2))/(2.*(B(I)-A(I)))
      GO TO 5
    4 VV=+QA(I)*(X-A(I) )+((QB(I)-QA(I))*((X-A(I))**2))/(2.*(B(I)-A(I)))
    5 V=V+VV
    6 CONTINUE
    7 IF(NOF)10,10,75
   75 DO 9 N=1,NOF
      IF(X+EPSX-XFO(N)) 9,9,8
    8 VV=FY(N)
      V=V+VV
    9 CONTINUE
   10 IF(NOR)13,13,105
  105 DO 12 M=1,NOR
      IF(X+EPSX-XR(M))12,12,11
   11 VV= RY(M)
      V=V+VV
   12 CONTINUE
   13 CONTINUE
       RETURN
      END
```

```
      SUBROUTINE MOMENT(X,AM)
      DIMENSION A(25),B(25),QA(25),QB(25),XFO(100),FY(100),XR(10),RY(10)
     1,XM(50),AMZ(50),RXM(10),RAMZ(10),AE(25),BE(25),EA(25),EB(25)
     2,GA(25),GB(25),AG(25),BG(25)
      COMMON A,B,QA,QB,XFO,FY,XR,RY,XM,AMZ,RXM,RAMZ,AE,BE,EA,EB,NOL,
     1 NOR,NOF,NOM,NORM,NOI,NOK,GA,GB,AG,BG
      EPSX=10.**(-8)
      AM=0.
      IF(NOL)8,8,2
    2 DO 7  I=1,NOL
      IF(X-A(I))7,3,3
    3 IF(X+EPSX-B(I))5,5,4
    4 AMM=(+QA(I)/2.)*((X-A(I))**2)+((QB(I)-QA(I))*((X-A(I))**3))/(6.*(
     1B(I)-A(I)))-(QB(I)/2.)*((X-B(I))**2)- (QB(I)-QA(I))*((X-B(I))**3)
     2/(6.*(B(I)-A(I)))
      GO TO 6
    5 AMM=(+QA(I)/2.)*((X-A(I))**2)+((QB(I)-QA(I))*((X-A(I))**3))/(6.*
     1(B(I)-A(I)))
    6 AM=AM+AMM
    7 CONTINUE
    8 IF(NOF)12,12,9
    9 DO 11 N=1,NOF
      IF(X+EPSX-XFO(N))11,11,10
   10 AMM= FY(N)*(X-XFO(N))
      AM=AM+AMM
   11 CONTINUE
   12 IF(NOR)16,16,13
   13 DO 15 M=1,NOR
      IF(X+EPSX-XR(M))15,15,14
   14 AMM= RY(M)*(X-XR(M))
      AM=AM+AMM
   15 CONTINUE
   16 IF(NOM)28,28,17
   17 DO 19 K=1,NOM
      IF(X+EPSX-XM(K))19,19,18
   18 AMM=AMZ(K)
      AM=AM+AMM
```

```
19 CONTINUE
28 IF(NORM) 32,32,29
29 DO 31 KA=1,NORM
   IF(X+EPSX-RXM(KA))31,31,30
30 AMM=RAMZ(KA)
   AM=AM+AMM
31 CONTINUE
32 CONTINUE
   RETURN
   END
```

```
      SUBROUTINE SLOPE(X,Y1)
      DIMENSION A(25),B(25),QA(25),QB(25),XFC(100),FY(100),XR(10),RY(10)
     1,XM(5C),AMZ(50),RXM(10),RAMZ(10),AE(25),BE(25),EA(25),EB(25)
     2,GA(25),GB(25),AG(25),BG(25)
      COMMON A,B,QA,QB,XFO,FY,XR,RY,XM,AMZ,RXM,RAMZ,AE,BE,EA,EB,NOL,
     1 NOR,NOF,NOM,NORM,NOI,NOK,GA,GB,AG,BG
      Y1=0.
      DO 17  IA=1,NOI
      IF(X-AE(IA))17,17,2
    2 IF(NOL)5,5,3
    3 DO  4  I=1,NOL
   20 EYY = EA(IA)*((QA(I)/2.)*ENTON(X,AE(IA),A(I),2) +((QB(I)-QA(I))/
     1(6.*(B(I)-A(I))))*(ENTON(X,AE(IA),A(I),3)-ENTON(X,AE(IA),B(I),3))
     2 -(QB(I)/2.)*ENTON(X,AE(IA),B(I),2))  +((EB(IA)-EA(IA))/(BE(IA)-
     3 AE(IA)))*((QA(I)/2.)*ENT1N(X,AE(IA),A(I),2) +((QB(I)-QA(I)) /
     4(6.*(B(I)-A(I))))*(ENT1N(X,AE(IA),A(I),3)-ENT1N(X,AE(IA),B(I),3))
     5 -(QB(I)/2.)*ENT1N(X,AE(IA),B(I),2))
      Y1=Y1+EYY
   21 EYY= -EB(IA)*((QA(I)/2.)*ENTON(X,BE(IA),A(I),2) +((QB(I)-QA(I))/
     1(6.*(B(I)-A(I))))*(ENTON(X,BE(IA),A(I),3)-ENTON(X,BE(IA),B(I),3))
     2 -(QB(I)/2.)*ENTON(X,BE(IA),B(I),2))  -((EB(IA)-EA(IA))/(BE(IA)-
     3 AE(IA)))*((QA(I)/2.)*ENT1N(X,BE(IA),A(I),2) +((QB(I)-QA(I)) /
     4(6.*(B(I)-A(I))))*(ENT1N(X,BE(IA),A(I),3)-ENT1N(X,BE(IA),B(I),3))
     5 -(QB(I)/2.)*ENT1N(X,BE(IA),B(I),2))
      Y1=Y1+EYY
    4 CONTINUE
    5 IF(NOF)8,8,6
    6 DO  7 N=1,NOF
   22 EYY= EA(IA)*FY(N)*ENTON(X,AE(IA),XFO(N),1) -EB(IA)*FY(N)*ENTON(X,
     1BE(IA),XFO(N),1) + ((EB(IA)-EA(IA))/(BE(IA)-AE(IA)))*FY(N)*
     2(ENT1N(X,AE(IA),XFO(N),1) - ENT1N(X,BE(IA),XFO(N),1))
      Y1=Y1+EYY
    7 CONTINUE
    8 IF(NOR)11,11,9
    9 DO 10 M=1,NOR
   23 EYY= EA(IA)*RY(M)*ENTON(X,AE(IA),XR(M),1)-EB(IA)*RY(M)* ENTON(X,
     1BE(IA),XR(M),1) +((EB(IA)-EA(IA))/(BE(IA)-AE(IA)))*RY(M) *
```

```
  2(ENTIN(X,AE(IA),XR(M),1)- ENTIN(X,BE(IA),XR(M),1))
   Y1=Y1+EYY
10 CONTINUE
11 IF(NOM) 14,14,12
12 DO 13  K=1,NOM
24 EYY= EA(IA)*AMZ(K)*ENTON(X,AE(IA),XM(K),0) -EB(IA)*AMZ(K)*ENTON(X,
  1BE(IA),XM(K),0) + ((EB(IA)-EA(IA))/(BE(IA)-AE(IA)))* AMZ(K) *
  2 (ENTIN(X,AE(IA),XM(K),0)- ENTIN(X,BE(IA),XM(K),0))
   Y1=Y1+EYY
13 CONTINUE
14 IF(NORM) 17,17,15
15 DO 16  KA= 1,NORM
25 EYY=EA(IA)*RAMZ(KA)*ENTON(X,AE(IA),RXM(KA),0)-EB(IA)*RAMZ(KA)*
  1 ENTON(X,BE(IA),RXM(KA),0) + ((EB(IA)-EA(IA))/(BE(IA)-AE(IA))) *
  2RAMZ(KA)*(ENTIN(X,AE(IA),RXM(KA),0)- ENTIN(X,BE(IA),RXM(KA),0))
   Y1=Y1+EYY
16 CONTINUE
17 CONTINUE
   IF(NOK)595,595,500
500 DO 590  IB=1,NOK
   IF (X -AG(IB))590,590,509
509 IF (NCL)530,530,510
510 DO 520  I=1,NOL
26 EYY = GA(IB)*(QA(I)*FTON(X,AG(IB),A(I),1) + ((QB(I) -QA(I) ) /
  1 (2.*(B(I)-A(I))))*(FTON(X,AG(IB),A(I),2)-FTON(X,AG(IB),B(I),2))
  2 -QB(I)*FTON(X,AG(IB),B(I),1)) + ((GB(IB)-GA(IB))/(BG(IB)-AG(IB)))
  3 *(QA(I)*FTIN(X,AG(IB),A(I),1) + ((QB(I)-QA(I))/(2.*(B(I)-A(I))))
  4* (FTIN(X,AG(IB),A(I),2) -FTIN(X,AG(IB),B(I),2)) - QB(I)* FTIN
  5 (X,AG(IB),B(I),2))
   Y1 =Y1 -EYY
27 EYY =-GB(IB)*(QA(I)*FTON(X,BG(IB),A(I),1) + ((QB(I) -QA(I)) /
  1 (2.*(B(I)-A(I))))*( FTON(X,BG(IB),A(I),2)-FTON(X,BG(IB),B(I),2))
  2-QB(I)* FTON(X,BG(IB),B(I),1))- ((GB(IB)-GA(IB))/(BG(IB)-AG(IB)))
  3*(QA(I)* FTIN(X,BG(IB),A(I),1)+ ((QB(I)-QA(I))/(2.*(B(I)-A(I))))
  4*( FTIN(X,BG(IB),A(I),2)- FTIN(X,BG(IB),B(I),2)) - QB(I)*  FTIN
  5 (X,BG(IB),B(I),2))
   Y1 =Y1 -EYY
```

```
520 CONTINUE
530 IF(NOF)560,560,540
540 DO 55C  N=1,NOF
 28 EYY=GA(IB)*FY(N)* FTON(X,AG(IB),XFO(N),0) -GB(IB)*FY(N)* FTON(X,
   1BG(IB),XFO(N),0) + ((GB(IB)-GA(IB)) / (BG(IB)-AG(IB))) *  FY(N)*
   2( FT1N(X,AG(IB),XFO(N),0) -  FT1N(X,BG(IB),XFO(N),C))
    Y1 =Y1 -EYY
550 CONTINUE
560 IF(NOR)590,590,570
570 DO 58C  M=1,NOR
 29 EYY =GA(IB)*RY(M)* FTON(X,AG(IB),XR(M),0) -GB(IB)*RY(M)* FTON(X,
   1BG(IB),XR(M),0) + ((GB(IB)-GA(IB)) / (BG(IB)-AG(IB))) * RY(M) *
   2( FT1N(X,AG(IB),XR(M),0) -  FT1N(X,BG(IB),XR(M),0))
    Y1 =Y1 -EYY
580 CONTINUE
590 CONTINUE
595 CONTINUE
    RETURN
    END
```

```
      SUBROUTINE DEFLECT(X,Y)
      DIMENSION A(25),B(25),QA(25),QB(25),XFO(100),FY(100),XR(10),RY(10)
     1,XM(50),AMZ(50),RXM(10),RAMZ(10),AE(25),BE(25),EA(25),EB(25)
     2,GA(25),GB(25),AG(25),BG(25)
      COMMON A,B,QA,QB,XFO,FY,XR,RY,XM,AMZ,RXM,RAMZ,AE,BE,EA,EB,NOL,
     1 NOR,NOF,NOM,NORM,NOI,NOK,GA,GB,AG,BG
      Y=0.
      DO 17  IA=1,NOI
      IF(X-AE(IA))17,17,2
    2 IF(NOL)5,5,3
    3 DO  4  I=1,NOL
   20 EYY = EA(IA)*((QA(I)/2.)*DITON(X,AE(IA),A(I),2) +((QB(I)-QA(I))/
     1(6.*(B(I)-A(I))))*(DITON(X,AE(IA),A(I),3)-DITON(X,AE(IA),B(I),3))
     2 -(QB(I)/2.)*DITON(X,AE(IA),B(I),2))  +((EB(IA)-EA(IA))/(BE(IA)-
     3 AE(IA)))*((QA(I)/2.)*DIT1N(X,AE(IA),A(I),2) +((QB(I)-QA(I)) /
     4(6.*(B(I)-A(I))))*(DIT1N(X,AE(IA),A(I),3)-DIT1N(X,AE(IA),B(I),3))
     5-(QB(I)/2.)*DIT1N(X,AE(IA),B(I),2))
      Y=Y+EYY
   21 EYY= -EB(IA)*((QA(I)/2.)*DITON(X,BE(IA),A(I),2) +((QB(I)-QA(I))/
     1(6.*(B(I)-A(I))))*(DITON(X,BE(IA),A(I),3)-DITON(X,BE(IA),B(I),3))
     2 -(QB(I)/2.)*DITON(X,BE(IA),B(I),2))  -((EB(IA)-EA(IA))/(BE(IA)-
     3AE(IA)))*((QA(I)/2.)*DIT1N(X,BE(IA),A(I),2) +((QB(I)-QA(I)) /
     4(6.*(B(I)-A(I))))*(DIT1N(X,BE(IA),A(I),3)-DIT1N(X,BE(IA),B(I),3))
     5 -(QB(I)/2.)*DIT1N(X,BE(IA),B(I),2))
      Y=Y+EYY
    4 CONTINUE
    5 IF(NOF)8,8,6
    6 DO  7 N=1,NOF
   22 EYY= EA(IA)*FY(N)*DITON(X,AE(IA),XFO(N),1) -EB(IA)*FY(N)*DITON(X,
     1 BE(IA),XFO(N),1) + ((EB(IA)-EA(IA))/(BE(IA)-AE(IA)))*FY(N)*
     2(DIT1N(X,AE(IA),XFO(N),1) - DIT1N(X,BE(IA),XFO(N),1))
      Y=Y+EYY
    7 CONTINUE
    8 IF(NOR)11,11,9
    9 DO 10 M=1,NOR
   23 EYY= EA(IA)*RY(M)*DITON(X,AE(IA),XR(M),1)-EB(IA)*RY(M)* DITON(X,
     1BE(IA),XR(M),1) +((EB(IA)-EA(IA))/(BE(IA)-AE(IA)))*RY(M) *
```

```
  2(DITIN(X,AE(IA),XR(M),1)- DITIN(X,BE(IA),XR(M),1))
   Y=Y+EYY
 10 CONTINUE
 11 IF(NOM)14,14,12
 12 DO 13  K=1,NOM
 24 EYY=EA(IA)*AMZ(K)*DITON(X,AE(IA),XM(K),0) -EB(IA)*AMZ(K)*DITON(X,
  1BE(IA),XM(K),0) + ((EB(IA)-EA(IA))/(BE(IA)-AE(IA)))*AMZ(K)*
  2 (DITIN(X,AE(IA),XM(K),0)- DITIN(X,BE(IA),XM(K),0))
   Y=Y+EYY
 13 CONTINUE
 14 IF(NORM) 17,17,15
 15 DO 16  KA= 1,NORM
 25 EYY=EA(IA)*RAMZ(KA)*DITON(X,AE(IA),RXM(KA),0)-EB(IA)*RAMZ(KA)*
  1 DITON(X,BE(IA),RXM(KA),0) + ((EB(IA)-EA(IA))/(BE(IA)-AE(IA))) *
  2RAMZ(KA)*(DITIN(X,AE(IA),RXM(KA),0)- DITIN(X,BE(IA),RXM(KA),C))
   Y=Y+EYY
 16 CONTINUE
 17 CONTINUE
   IF(NOK)595,595,500
500 DO 59C  IB=1,NOK
   IF (X -AG(IB))590,590,509
509 IF (NCL)530,530,510
510 DO 52C  I=1,NCL
 26 EYY=GA(IB)*(QA(I)* ENTON(X,AG(IB),A(I),1) + ((QB(I) -CA(I) ) /
  1 (2.*(B(I)-A(I))))*(ENTON(X,AG(IB),A(I),2)-ENTON(X,AG(IB),B(I),2))
  2-QB(I)*ENTON(X,AG(IB),B(I),1)) + ((GB(IB)-GA(IB))/(BG(IB)-AG(IB)))
  3*(QA(I)*ENTIN(X,AG(IB),A(I),1) + ((QB(I)-CA(I))/(2.*(B(I)-A(I))))
  4*(ENTIN(X,AG(IB),A(I),2)-ENTIN(X,AG(IB),B(I),2)) - QB(I)*ENTIN
  5 (X,AG(IB),B(I),2))
   Y=Y-EYY
 27 EYY=-GB(IB)*(QA(I)*ENTON(X,BG(IB),A(I),1) +((QB(I)-QA(I))/
  1(2.*(B(I)-A(I))))*(ENTON(X,BG(IB),A(I),2)-ENTON(X,BG(IB),B(I),2))
  2-QB(I)*ENTON(X,BG(IB),B(I),1))- ((GB(IB)-GA(IB))/(BG(IB)-AG(IB)))
  3*(QA(I)*ENTIN(X,BG(IB),A(I),1)+ ((QB(I)-QA(I))/(2.*(B(I)-A(I))))
  4*(ENTIN(X,BG(IB),A(I),2)-ENTIN(X,BG(IB),B(I),2)) - QB(I)* ENTIN
  5 (X,BG(IB),B(I),2))
   Y=Y-EYY
```

```
520 CONTINUE
530 IF(NOF)560,560,540
540 DO 550  N=1,NOF
 28 EYY=GA(IB)*FY(N)*ENTON(X,AG(IB),XFO(N),0) -GB(IB)*FY(N)*ENTON(X,
   1BG(IB),XFO(N),0) + ((GB(IB)-GA(IB)) / (BG(IB)-AG(IB))) *  FY(N)*
   2(ENT1N(X,AG(IB),XFO(N),0) - ENT1N(X,BG(IB),XFO(N),0))
    Y=Y-EYY
550 CONTINUE
560 IF(NOR)590,590,570
570 DO 580  M=1,NOR
 29 EYY =GA(IB)*RY(M)*ENTON(X,AG(IB),XR(M),0) -GB(IB)*RY(M)*ENTON(X,
   1BG(IB),XR(M),0) + ((GB(IB)-GA(IB)) / (BG(IB)-AG(IB))) * RY(M) *
   2(ENT1N(X,AG(IB),XR(M),0) - ENT1N(X,BG(IB),XR(M),0))
    Y=Y-EYY
580 CONTINUE
590 CONTINUE
595 CONTINUE
    RETURN
    END
```

```
      FUNCTION FTON(X,A,B,N)
      FTON = 0.
      IF(X-A)3,3,1
    1 IF(X-B)3,3,2
    2 FTON=(X-B)**N
    3 RETURN
      END
```

```
      FUNCTION FT1N (X,A,B,N)
      FT1N =0.
      IF (X-A)3,3,1
    1 IF (X-B)3,3,2
    2 FT1N = (X-B)**(N+1) + (B-A)*((X-B)**N)
    3 RETURN
      END
```

```
      FUNCTION  ENTON (X,A,B,N)
      FN=N
      ENTON = 0.
      IF(X-A)5,5,1
    1 IF(X-B)5,5,2
    2 IF(A-B)3,3,4
    3 ENTON = (X-B)**(N+1)/(FN+1.)
      GO TO 5
    4 ENTON = (X-B)**(N+1)/(FN+1.) - (A-B)**(N+1)/(FN+1.)
    5 RETURN
      END
```

```
      FUNCTION  ENT1N(X,A,B,N)
      FN=N
      ENT1N = 0.
      IF(X-A)5,5,1
    1 IF(X-B)5,5,2
    2 IF(A-B)3,3,4
    3 ENT1N = (X-B)**(N+2)/(FN+2.) +  (B-A)*(X-B)**(N+1)/(FN+1.)
      GO TO 5
    4 ENT1N = (X-B)**(N+2)/(FN+2.) + (B-A)*(X-B)**(N+1)/(FN+1.)
     1 -(A-B)**(N+2)/(FN+2.) -((B-A)*(A-B)**(N+1))/(FN+1.)
    5 RETURN
      END
```

```
      FUNCTION DITON(X,A,B,N)
      FN=N
      DITON =0.
      IF(X-A)5,5,1
    1 IF(X-B)5,5,2
    2 IF(A-B)3,3,4
    3 DITON = (X-B)**(N+2)/((FN+1.)*(FN+2.))
      GO TO 5
    4 DITON = (X-B)**(N+2)/((FN+1.)*(FN+2.)) -(X*(A-B)**(N+1))/(FN+1.)
     1-(A-B)**(N+2)/((FN+1.)*(FN+2.))  + (A*(A-B)**(N+1)))/(FN+1.)
    5 RETURN
      END
```

```
      FUNCTION  DIT1N(X,A,B,N)
      FN=N
      DIT1N=0.
      IF(X-A)5,5,1
    1 IF(X-B)5,5,2
    2 IF(A-B)3,3,4
    3 DIT1N =(X-B)**(N+3)/((FN+2.)*(FN+3.)) + ((B-A)*(X-B)**(N+2))/
     1((FN+1.)*(FN+2.))
      GO TO 5
    4 DIT1N = (X-B)**(N+3)/((FN+2.)*(FN+3.)) + ((B-A)*(X-B)**(N+2))/
     1((FN+1.)*(FN+2.))-(X*(A-B)**(N+2))/(FN+2.)-(X*(B-A)*(A-B)**(N+1))/
     2 (FN+1.) - (A-B)**(N+3)/((FN+2.)*(FN+3.)) - (B-A)*(A-B)**(N+2)/
     3 ((FN+1.)*(FN+2.)) + (A*(A-B)**(N+2))/(FN+2.) + (A*(B-A)*(A-B)**
     4 (N+1))/(FN+1.)
    5 RETURN
      END
```

```
      SUBROUTINE  GAUSS2(N,EP,A,X,KER)
      DIMENSION A(22,23) ,X(22)
      NPM=N+1
   10 DO 34 L=1,N
      KP=0
      Z=0.C
      DO 12 K=L,N
      IF(Z-ABSF(A(K,L)))11,12,12
   11 Z=ABSF(A(K,L))
      KP=K
   12 CONTINUE
      IF(L-KP)13,20,20
   13 DO 14 J=L,NPM
      Z=A(L,J)
      A(L,J)=A(KP,J)
   14 A(KP,J)=Z
   20 IF(ABSF(A(L,L))-EP)50,50,30
   30 IF(L-N)31,40,40
   31 LP1=L+1
      DO 34 K=LP1,N
      IF(A(K,L))32,34,32
   32 RATIO=A(K,L)/A(L,L)
      DO 33 J=LP1,NPM
   33 A(K,J)=A(K,J)-RATIO*A(L,J)
   34 CONTINUE
   40 DO 43 I=1,N
      II=N+1-I
      DO 43 J=1,1
      JPN=J+N
      S=0.0
      IF(II-N)41,43,43
   41 IIP1=II+1
      DO 42 K=IIP1,N
   42 S=S+A(II,K)*X(K)
   43 X(II)  =(A(II,JPN)-S)/A(II,II)
      KER=1
   75 RETURN
```

```
50 KER=2
   RETURN.
   END
```

thesL8573
Digital computer program for flexural an
3 2768 002 12686 4
DUDLEY KNOX LIBRARY