DEVELOPMENT OF MICRO-COMPUTER PROGRAMS FOR THE ANALYSIS OF AN OPEN SPANDRAL ARCH

by

JOHN JAY KOONTZ

B.S., United States Military Academy, 1971

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Civil Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1984

Approved by:

Major Professor

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

LIST OF TABLES AND FIGURES

CHAPTER 1

Introduction

The arrival of the micro-computer has been so rapid that some of the conclusions reached in the twelve month span of this research have been significantly changed. Increased usable memory, miniaturization, increased flexibility and decreased cost have made the micro-computer one of the most important tools available to the professional engineer today. The smallest firm can now afford to own a computer, and in fact needs to do so to stay competetive. The micro-computer has been accepted as a word processor and a valuable accounting tool for some time. Now with the recent increases in usable memory, even the most complicated analysis problems are within the capabilities of the micro. At this point in time, software to take advantage of these capabilities is still being developed. This research effort has produced one set of software to solve such a problem.

Until now, the analysis of a bridge composed of a continuous beam supported by a fixed arch has been restricted to either a very long approximate analysis or use of a mainframe program such as STRUDL. This effort produced a software package to accomplish such an analysis using a micro-computer. Matrices are applied to the displacement method for the solution of the problem. The use of this 'exact' theory has previously been restricted to mainframe computers because of the large memory requirements. In fact, it is the magnitude of the memory requirements that caused the greatest problems in the development of this package of software.

Equipment and Program Language

The first decision in the process was the selection of a language to be used. The language needed to be one that was universally available for microcomputers. The analysis software would then be adaptable to systems other than the one on which it was programmed. BASIC fills this requirement with additional benefits. BASIC is generally the first language available in a micro system. It is often the only language which is contained in ROM, that is, available without the requirement that it be stored on the disk and loaded into the memory. The language is user oriented and programs can be written to provide interaction between the user and the machine. For these reasons, most users will be familiar with the language. This familiarity benefits the user because he has the option of examining the program logic by reading the code. He can verify assumptions or procedures or make changes more readily. Because of it's almost universal availability and acceptance, BASIC is the language used in this package.

The Apple II micro-computer was selected from the two systems available because the author was familiar with it and had easy access to the hardware. The programs developed here are specifically written in Applesoft, a version of BASIC. Very little modification would be required to adapt the programs to any currently available system. Most of the necessary modifications would involve the DOS (Disk Operating System) commands and some input/output commands rather than language differences. The DOS is used in the storage and retrieval of data files and subprograms within the analysis package.

Assumptions and Limitations

The general problem which is addressed here is how the displacement method can be programmed on a micro-computer for the analysis of a large, complex structure. This is accomplished by writing a special program for a specific structure. This simplifies the solution by selecting the desired boundary conditions and the general geometry of a particular type of structure without invalidating the general solution procedure. This procedure, as developed, can then be adapted for other related structures, or, possibly, as a framework for more general cases. The specific structure selected is a continuous beam bridge supported by a fixed arch. The beam and arch are connected by vertical columns with rigid joints. This structure is most useful in spanning a canyon or other obstacle where mid-span supports are impractical, such as the one in Figure 1.1. Certain initial assumptions and limitations are accepted in order to simplify the structural considerations of this problem. These assumptions and limitations are;

(1) The analysis is two-dimensional. Only the effects due to the forces in the plane of the structure are of interest.

(2) Vibration and temperature analysis are not included in the development.

Typical Structure Fig. 1.1

(3) It is assumed that sufficient cross-bracing is present to eliminate the need of considering lateral sidesway.

(4) Wind load is not considered in the development because of limited time and because of (1) and (3) above.

(4) Torsion is not considered.

(5) Stresses will be in the elastic range.

(6) Displacements are assumed to be relatively small. Therefore the nonlinearity due to deformations can be ignored.

(7) The geometry of the structure is assumed to be of the general form shown in Figure 1.2.

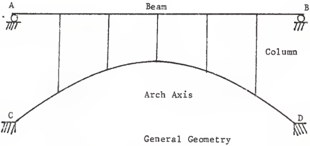


Figure 1.2

The beam supports at ends A and B are rollers. The arch supports at C and D are fixed. The user has the option of placing a column at the arch support if desired.

(8) The arch axis is assumed to be formed by straight line segments between columns.

(9) Gravity loads are assumed to be in the vertical direction and are perpendicular to the beam.

There is no theoretical limit on the size or number of spans which can be considered using this package. However, there is a practical limit which is related to the storage capacity of the system being used. This will be discussed later.

The loading of the structure, which conforms to the AASHTO code, is handled in seperate programs within the package. These programs permit consideration of both the rolling load and the equivalent loading. One of the functions of the package is to permit a comparison of the two load conditions.

The package is written to be interactive with the user. The intent of the package is to provide a design aid which eliminates the tedious calculations normally involved with an analysis of this type. At the same time, it improves on the analysis normally done by hand calculations by including the effects due to axial deformations into the analysis. It provides a tool to produce an output which consists of the forces and deformations corresponding to the dimensions selected and loading specified. This can save time, reduce error and enable the user to find ways to improve the approximate solution in a structural design. At the same time, the programs must be 'friendly' in that the user is not constantly bombarded with questions to answer which could be handled by the computer. The realization of this balance results in a program which can be integrated into the design process with confidence and a significant saving of time and materials.

CHAPTER 2

Method of Analysis

Matrix Equation for Linear Elastic Structures

The purpose of this battery of programs is to provide an analysis of a bridge using displacement method. This method is based on linear elastic theory and is well developed and universally accepted. The principal tool used is the slope-deflection equation. Because the structure is statically indeterminant to a large degree, matrix methods are applied to simplify the calculations by expressing them in a format which is most efficient for the computer. This procedure has also been well developed and is commonly used in mainframe computer solutions. The use of matrix methods with the aid of a computer also permits the inclusion of axial deformation in the analysis.

There are three principal assumptions on which this theory is based. First, the members of the structure will behave in the elastic region. Second, the members are homogeneous and isotropic. Third, displacements are small. All of these assumptions remain valid for this structure.

When applying the displacement method, the basic matrix equation which characterizes the relationship between forces and displacements is;

$$X = K\ u \tag{2.1}$$

where X is a force vector, K is the stiffness matrix, and u is the corresponding displacement vector of all the joints. The approach of this program is to calculate displacements caused by the applied loading on the structure. In partitioned form, equation 2.1 can be written as;

$$\begin{Bmatrix} F \\ \hline R \end{Bmatrix} = \left|\begin{array}{c|c} K_{ff} & K_{fr} \\ \hline K_{rf} & K_{rr} \end{array}\right| \begin{Bmatrix} U_f \\ \hline U_r \end{Bmatrix} \tag{2.2}$$

where F is a vector of given external forces, R is a vector of unknown reaction forces of grounded supports, and U_f and U_r are the displacements corresponding to F and R respectively. The displacements are then calculated as;

$$U_f = K_{ff}^{-1}\ F \tag{2.3}$$

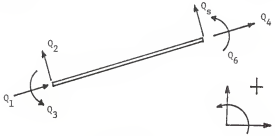
since K_{ff} is a nonsingular matrix. The displacements are then used to calculate the end-forces on each member using the slope-deflection equations.

Coordinate Transformation

The K matrix is assembled member by member. This is accomplished by considering the stiffness matrix of a single member, K_1, of the form;

$$K_1 = \frac{1}{L^3}\begin{vmatrix} AEL^2 & 0 & 0 & -AEL^2 & 0 & 0 \\ 0 & 12EI & 6EIL & 0 & -12EI & 6EIL \\ 0 & 6EIL & 4EIL^2 & 0 & -6EIL & 2EIL^2 \\ -AEL^2 & 0 & 0 & AEL^2 & 0 & 0 \\ 0 & -12EI & -6EIL & 0 & 12EI & -6EIL \\ 0 & 6EIL & 2EIL^2 & 0 & -6EIL & 4EIL^2 \end{vmatrix} \quad (2.4)$$

This matrix is a local matrix based on a coordinate system where the longitudinal axis of the member is considered to be the x-axis. The order of arrangement of forces and displacements is shown in Figure 2.1.



Sign Convention

Figure 2.1

In order to analyze the entire structure, a global coordinate system must be used. The local K_1 matrix must then be transformed to the global system. This is accomplished by;

$$K_g = T^t K_l T \tag{2.5}$$

where T is the transformation matrix and T^t is the transpose of the transformation matrix. Adopting a sign convention where upward, to the right, and counterclockwise are positive, as shown in Figure 2.1, the transformation matrix will be;

$$T = \begin{vmatrix} \text{Cos A} & -\text{Sin A} & 0 & 0 & 0 & 0 \\ \text{Sin A} & \text{Cos A} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \text{Cos A} & -\text{Sin A} & 0 \\ 0 & 0 & 0 & \text{Sin A} & \text{Cos A} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix} \tag{2.6}$$

For convenience of notation, the order number of the upper end of each column is treated the same as that of the left end of the beam spans.

Global Indexing System

The next step is to assemble the global stiffness matrix for the entire structure by superposing the member matrices. Row and column numbers of each element of the stiffness matrix of a structural member must be indexed to correspond to the correct order of the joint force and joint displacement matrices in the global system. This is simplified by standardizing the displacement vector arrangement in the following order and is represented by a row vector as follows;

$$u = \{ dx_1, dy_1, r_1, dx_2, dy_2, ...dx_n, dy_n, r_n \} \tag{2.7}$$

where the subscripts are the order numbers of the joints and dx and dy are the horizontal and vertical displacements and r is the rotational displacement. The detailed discussion of the mechanics of how the indexing is accomplished is presented in Chapter Three.

Method of Solution

Once the global structural stiffness matrix is assembled, it is inverted by use of the Gauss-Jordan elimination method with a full search for the largest pivot elements. The inverse of the stiffness matrix (which is a flexibility matrix for the structure) is stored on a data disk.

After finding the inverse of the stiffness matrix, the structural displacement vector is calculated by Eqn. 2.3 for any given loading. The displacement vector represents the displacement components of the joints of the structure in the format as shown in Eqn. 2.7 above which result from the given loading.

Internal Forces

The joint displacements are used in the slope-deflection equation to calculate the moments on either end of the member. Taking moments on the resulting free-body diagram then yields the

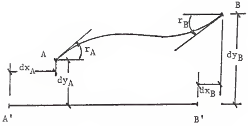


Figure 2.2a Displacement Components

Figure 2.2b End Force Components

shear and axial forces on the end of the member. These relationships are;

$$M_{AB} = EI/L(\ 4r_A + 2r_B - 6dy_{AB}/L) + FEM_{AB} \quad (2.8)$$

$$V_A = (M_{AB} + M_{BA})/L + FEV_A \quad (2.9)$$

$$N_A = AEdx_{AB}/L \quad (2.10)$$

where $dy_{AB} = dy_B - dy_A$ and $dx_{AB} = dx_B - dx_A$ as shown in Figures 2.2a and 2.2b. These forces are those which correspond to a particular load case.

AASHTO Live Loading

For live load analysis, there are two load cases which are specified by the AASHTO codes for use in the design of a bridge. The first case requires that the maximum positive and maximum negative force components (normal force, shear force and moment) be found for an HS 20 tractor and trailer crossing the bridge. This is referred to as the AASHTO rolling load. The second case requires the use of an equivalent uniform load and a concentrated load placed so as to produce the worst possible condition for the force being considered. These are illustrated in Figure 2.3. In either case, the calculations are made easier by using the appropriate influence line.

Construction of the Influence Lines

Construction of the influence line is readily accomplished by moving a unit load incrementally across the bridge and

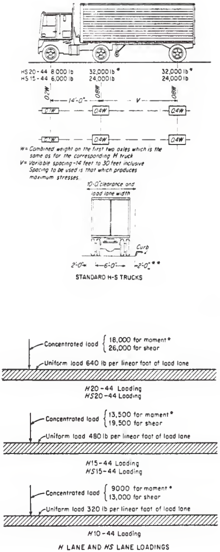


Figure 2.3 AASHTO Code Live Loading

calculating the resulting forces at each step. These force vectors are arranged column by column in a matrix of end forces. The influence lines are then the rows of the matrix and will correspond to the forces in the same order as that of the displacement components given in Eqn. 2.7 above. This method is simple and direct and was used initially. However, limited disk storage and memory made the selection of another method preferrable. Discussion of the computer requirements is dealt with in Chapter Three. The second method is accomplished in two steps. First, unit fixed-end moments and shears are applied to each joint of the beam and the corresponding end-forces are calculated using the matrix methods described above. The results form vectors of end forces which shall be called influence coefficient vectors. Then a unit load is moved incrementally across each span (or beam segment between joints). The fixed-end moments and shear values corresponding to each step are calculated and multiplied by the corresponding elements of the influence coefficient vectors to find the desired influence lines. This indirect method is somewhat more complex, but considerably reduces the amount of disk storage space required.

Finding Critical Live Load Conditions

The application of the AASHTO equivalent load is carried out by placing uniform loads and a concentrated load on the structure in such a way as to maximize, in turn, the positive and negative forces on each member, respectively. Knowing the influence line, determination of the load conditions is fairly direct. To obtain the maximum positive force, the value of the uniform load is multiplied by the total positive area under the influence line first and then the value of the product of the concentrated load and the maximum positive ordinate of the influence line is added to that result. The same procedure is used to obtain the maximum negative force using the negative areas and ordinates of the influence lines.

The AASHTO rolling load results are also obtained by use of the influence lines. The AASHTO code allows variation of the trailer length from fourteen to thirty feet. The front axle of the tractor is moved incrementally across the bridge. At each step, the trailer length is varied to obtain the maximum and minimum values of the force being calculated. The load of each axle is multiplied by the ordinate of the influence line at the location of the axel. The sum of the three products is the force corresponding to that unique location of the tractor-trailer. The front axle is moved another increment and the process is repeated. The new values obtained are compared to previous maximum and minimum values and replace them if appropriate. Once the rolling has completely crossed the structure, the maximum and minimum values have been obtained for the force corresponding to the influence line. It is recognized that this is not the most efficient method to accomplish a comprehensive analysis, but memory restrictions necessitate this procedure. This will be discussed further in Chapters Three and Six.

Internal Forces Due to Yield of Supports

In addition to the determination of the maximum force values, the matrix method can be used conveniently to analyze the effect of support settlement on a structure. When the settlement, translation, or rotation of one or more supports is known the resulting forces imposed on the structure can be determined. In order to accomplish this, it is necessary to return to the original partitioned stiffness matrix Eqn. 2.2. In this case the external loading portion of the force vector is equal to zero. The resulting matrix multiplication yields;

$$0 = K_{ff}\, U_f + K_{fr}\, U_r \tag{2.11}$$

Solving for the joint displacements;

$$U_f = -\, K_{ff}^{-1}\, K_{fr}\, U_r \tag{2.12}$$

It is noted that the K_{ff} inverse matrix has been calculated, stored and is ready for use. Only the K_{fr} protion of the matrix need be constructed to find the joint displacement vector from the known support displacements. Once the joint displacments are known, the end-forces of each member may be determined using the slope deflection formulas.

Established matrix methods are used to complete the structural analysis of this bridge. The structural stiffness matrix is constructed and used in conjunction with either the loading vector or the support displacement vector to determine the end-forces on each member of the structure. The speed and memory of the computer are used to perform the calculations which would otherwise be very difficult to accomplish by hand.

CHAPTER 3

The Internal Force Analysis Package

Application Limitations

With the system and language selected, and the method of analysis established, the main thrust of this effort was to merge the two into a usable package of software. In the accomplishment of this objective, numerous limitations of the Apple II computer were encountered. These limitations forced changes and modifications to the final product which resulted in both improvements and limitations in the utility of the package. In the process of overcoming the limitations of the system, many important questions regarding the use of micro-computers in structural engineering were answered. It is the development of the software package which is now addressed.

Initially, it was thought that the problem could be solved within the structure of one large program. However, the memory requirements of the required matrix manipulations are so great that the capacity of the micro-computer is exhausted in the solution of a four span structure of the nature considered here. If the computer were restricted to such a small structure, then its utility in the solution of complicated structural analysis problems would be inconsequential. The solution to this problem is to break the general solutioo into a series of individual programs. Each program is designed to perform a particular task. The results of each program are stored on a data disk for retrieval as needed. The result reduces memory requirements to that needed for the solution of each individual step of the analysis instead of that needed to solve the entire problem.

The problem is broken down into a series of steps. Each step is designed to accomplish one basic operatioo aod store the results on a data disk. In each successive step, only the input data needed for that step is read from the data disk or input from the keyboard. Matrix operatioos of previous steps are not retained in memory so that maximum usage of that memory is obtaioed. The matrix variables dimensioned and used io each program are the greatest memory users, so memory usage may be planned fairly closely by considering principally those variables with some allowaoce made for the needs of the program itself. An analysis of the minimum requirements for each program of the package is showo in Table 3.1. Inspection of the table reveals that the program which calculates deflections uses the most memory and therefore limits the size of the structure which can be solved. Using this table it was predicted that a twelve span bridge with a total of 35 members could be solved using this package. This was verified io trial runs when a twelve span bridge ran without memory problems while the computer was unable to even start the stiffness matrix computation of a thirteeo span trial. Memory constraints of a 48K system will then limit this program to consideration of a bridge of twelve or fewer spans.

TABLE 3.1 Memory Requirement Analysis		
PROGRAM	MATRICES	MEMORY
Structural and Geometric Properties	JN, L, CS, SN, X, Y, EI, AE	NB x 7 + N x 2
Flexibility Matrix Generation	CK, K, KG, KT, KE L, T, SN, CS, AE, EI	NB x 5 + 6 x 25 + N x N
Loading	DL, V, N, M, P, F	NB x 8 + N + 6
Internal Forces Calculation	JN, DL, D, CS, SN, V, M, N, L, EI, AE	NB x 15 + N
Displacement Calculation	D, K, P	N x (N + 2)

Where NB = number of members
N = number of joints x 3 - 8

As stated, each program represents the solution of one phase of the operations needed to complete the analysis of the structure. The first program of the package is the 'Hello' program. This unit introduces the user to the problem and presents the options available. Detailed discussion of the options is left until after the various computational programs are discussed, at which time the use of the options is more easily understood.

Structural Data Input Program

The first computational program establishes the geometric and structural properties of the structure. The user is queried from the monitor about the structure to be analyzed. The structural and geometric parameters are input from the keyboard and then the opportunity to review and change any input data desired is given. Because it is assumed that the bridge deck is horizontal, it is only necessary to input the lengths of the beam spans and columns, and the locations of the arch supports. The program itself then automatically determines the geometric properties. This includes numbering the joints, the identification of joint numbers for each member, determination of the global orientation of each member with respect to the structure, and calculation of the length and unit vectors for each member. This is accomplished by establishing a global grid system which sets the left end of the beam as the origin. The user inputs the structural information in the form of AE and EI for each member. The input data is stored in the form in which it will be used in later programs. For example, instead of storing the grid locations, the length and orientation of each member is stored. Except for loading cases, this is the last data input required of the user. Throughout the remaining programs only available options must be answered.

Flexibility Matrix Generation

The second program is technically the most complicated of all of the programs in the package. In this program, the local stiffness matrix of each member is generated and then transformed into the global system. The stiffness matrix for the structure is then assembled and inverted. As discussed in the development of the theory in Chapter 2, this is accomplished by first assembling the stiffness matrix for each member and transposing it to the global system. There are several decisions which were necessary to accomplish this. First, the order of displacements is standardized to progress joint by joint. At each joint the X displacement is first, the Y displacement is second and the rotation is third. Support displacements are eliminated so that the global matrix may be partitioned as described previously. The sign convention is set with upward, to the right and counterclockwise being selected as positive (see Fig 2.1). This sign convention is then adhered to for the remainder of the solution of the analysis. The member matrices are assembled according to the same format and sign convention using the procedure described in Chapter 2. As each member matrix is transformed to the global coordinate system, it is indexed such that the rows and columns will correspond to the appropriate global rows and columns. The member matrix elements are then superposed on the global matrix. This procedure procedes member by member until the structural K_{ff} matrix is assembled. When this is accomplished, the K_{ff} matrix is inverted using the Gauss-Jordan elimination method. The inverted K matrix is the stored on the data disk for use in later programs.

This program is automatic, requiring the user to input only the name of the input file to be used and the name to be assigned to the stiffness data file. The computation portion of the program is very long. It is here that another limitation of the micro-computer is encountered. A six span bridge will require approximately forty-five minutes to complete. This expands logrithmically so that a twelve span structure requires nearly four hours. While this is much slower than a main-frame computer, it is felt that the advantages of having this tool available on a micro-computer outweigh the time inconvenience. It was found during testing that other activities could, if planned effectively, be done during this time. Compiling the program into machine language would, of course, reduce this time factor considerably. A limitation peculiar to the Apple II system is also exposed by this program. Storage of the output data from this program takes up much more disk space than it aparently should. Real numbers require five bytes of memory to store in the computer. However, when they are stored on a data disk, the Apple stores them as string variables requiring twelve bytes for each number. This greatly reduces the flexibility of this package of programs, requiring a minimum of three data disks to complete an AASHTO load analysis for a six span structure.

The resulting disk manipulations also lengthen nearly all of the programs in the package. A system which stores real numbers as such on the data disk would greatly aleviate this problem and make the program more useful.

Loading Data Input Program

The next program of the package assembles the joint loading vectors. Loading may be input in any one of a combination of three different formats. Fixed-end moments and shears, uniform and concentrated loads in the spans, and direct joint loading are the options available. Selection of the appropriate option or combination of options permits the application of nearly any loading condition. In addition, an option is available which will automatically generate the necessary load cases for the calculation of influence lines with this package. While dead loads are not automatically included, selection of the appropriate options permits the inclusion of dead loads in the analysis. If influence lines are desired for the evaluation of AASHTO rolling loads or for other purposes, the user need only select the option and input the name of the input file to be used. The rest is automatic including the storage of the load case files. If other options are selected, loading is input member by member. All loading options are automatically converted to equivalent joint loads and fixed-end moments, shears, and axial forces for storage and use in later programs. While the user is permitted to input downward loading as positive to conform with accepted practice, it is converted to the program sign convention automatically and stored in that form. Error handling routines prevent the program from stopping due to inadequate data disk storage space. This protection is necessary due to the inefficient storage procedure of the Apple II system as noted in the previous section.

Once the load cases to be analyzed have be stored, the next program uses them in conjunction with the flexibility matrix to calculate the joint displacements. Once again, the user need only identify the flexibility matrix to be used and input whether or not influence lines are to be calculated. If the influence line option is selected, the program automatically calculates and stores the necessary displacement files. If other load cases are to be analyzed, the user need only input the name of the case to be used. As many cases as desired can be calculated using the same flexibility matrix. Each resulting displacement file is stored for later use. An error handling routine is used to prevent the loss of the program or data due to lack of disk storage space or due to other data handling errors, such as identifying a file for input which does not exist.

Calculation of Internal Forces

The next program is the final computation unit for all special load case analysis. Only the analysis of AASHTO loading (either rolling or equivalent) or the settlement of supports requires additional computation. This program calculates the end forces for each member of the structure. This program requires portions of the input file, load case file(s), and displacement file(s) for input. An internal check is made to ensure compatability of these files before any computations are made.

If the user is calculating influence lines, the coefficient files are calculated automatically. The end forces for each member are determined by using the slope-deflection equations. This requires the relative displacement of the ends of the member and the inclusion of fixed-end moments or forces from the loading. The latter are read from the load case files. The displacements are determined from the joint displacements of the structure. Since they are determined in reference to the global coordinate system, they must be transformed to the local coordinate system of the member in question. Once this is accomplished, the shear and axial forces and moments on the ends of the member are calculated directly. After all member forces have been calculated, all end forces relating to the given load case considered are stored . An error handling routine prevents any mismatch of files, such as attempting to use a displacement file which was calculated from a different load case than the one being used here. The routine also prevents loss of the data from such a cause as a full data disk. If the user wishes to construct influence lines, the next program is automatically provided. Otherwise the option to return to the master menu in the 'Hello' program or to the 'Printer' program for hard copies of data files is provided.

This completes the core of the computational programs. The programs which follow are special purpose and will be reached only in certain circumstances.

Influence Line Program

After calculating coefficient files, influence lines for every member are generated by the next program. The influence lines are generated for a unit load at any location along the continous beam of the bridge. This allows the structural engineer a great deal of flexibility in examining any form of loading on the bridge deck without repeating long calculations. As discussed in the theory in Chapter 2, this is accomplished by generating and using influence coefficients. These coefficients have been automatically calculated by applying unit fixed-end moments and shears to the joints of the beam. The resulting forces were calculated for each member and stored case by case. These files are now read automatically and placed in memory as elements of the influence coefficient matrix. A unit downward load is then applied to each span of the beam sequentially. The load is started an infinitesimal distance to the right of the left end of the span. The fixed end moments and shears are calculated for the span which result from this load. These values are multiplied by the corresponding coefficient file values for a given force on the end of a given member and the results superposed. This establishes the influence line value for the given force corresponding to the location of the unit load. The load is then moved to the right 0.1 times the length of the span and the computations repeated. This process is continued until the unit load reaches the right end of the last span of the beam. At this point the influence line is complete and is stored on a data disk. The entire process is repeated for each of the internal forces on the structure until all influence lines have been calculated and stored. The error handling routine permits changing data disks without interrupting the program and losing the data.

AASHTO Rolling Load Program

The influence lines are used in the next program to compute the AASHTO rolling load influence lines for the structure. This program uses the AASHTO HS20 tractor-trailer. The load on each axle is set as well as the spacing of the front two axles. The code allows variable spacing of the rear axle and this is accomplished automatically by the program. One influence line is read from the data disk and the front axle is moved across the structure in increments equal to 0.1 times the length of the span it is in. The locations of the other two axles are then determined for each location of the front axle. The load values corresponding to the three axles are then multiplied by the influence line values corresponding to the locations of the respective axles. The sum of these products is the value of the end force corresponding to the influence line being used. The trailer spacing is incremented one foot and the end force value is recalculated. This continues until the trailer spacing reaches thirty feet. Then the front axle is moved and the variation of the trailer spacing is repeated. At each step a check is made to see if a new maximum or minimum has been found. If so, then the appropriate variable is reset to the new value. Because the front axle leaves the structure first, an illusory span is automatically created so that the load af the rear axles may be evaluated while they are still on the bridge. The process continues until the last axle departs the last span. A file containing the maximum and minimum values calculated will be stored on a data disk. This file can be compared directly with the output of the equivalent load program described next, as it is stored in the same format. Such a comparison is shown in Table B-2.

AASHTO Equivalent Loading Program

The AASHTO code permits design by using an equivalent loading on the bridge. Since the influence lines have been stored on disk, they may be used effectively to determine the worst case values for equivalent loads. First, an influence lines is read from the data disk. Then the uniform portion of the equivalent loading is multiplied by the total positive area and the total negative area under the influence line. These values are stored separately in memory. At the same time, the concentrated portion of the equivalent loading is multiplied by the maximum positive and negative ordinates of the influence line. The uniform load and concentrated load products are summed for positive and negative values. These values represent the worst case positive and negative values of the force which the influence line represents. The worst cases for each force are stored in memory and the next case is calculated. When every force for every member has been calculated, they are stored on the data disk in one file. Completion of the last two files permits comparison of the two loading cases permitted by the AASHTO codes. This completes the load analysis of the structure.

Yielding of Supports Program

The final computational program permits the evaluation of the settlement, translation and/or rotation of any support for the structure. Prior to using this program, only the input and stiffness matrix files need to have been calculated. These files are read from the data disk, requiring no additional input from the user. The computation of member forces resulting from support displacement requires the calculation of the joint displacements of the structure. To do this the K_{fr} portion of the flexibility matrix needs to be assembled, per the theory developed at the end of Chapter 2. The assembly of the K_{fr} portion proceeds in the same manner as the assembly of the K_{ff} partition did in the second porgram. The local flexibility matrix is assembled for each member and transposed to the global coordinate system. The difference occurs only when the local matrix rows and columns are indexed to assemble the global K_{fr} partition of the global flexibility matrix. Once this is accomplished, the user is asked to input the support displacements to be examined. The matrix multiplications are performed per Eqn. 2.12. The result is a displacement vector containing the joint displacements of the structure which result from the support displacement. This vector is stored on a data disk. The resulting forces can then be analyzed in the same manner as described earlier.

Utility Programs

The final two programs contained in this package are utility programs. The first program permits the transfer of either an input file or a flexibility file from one data disk to another.

Although each program is protected by an error handling routine, it is often desireable to have either one or both of these files on the data disk which is be used. This is true for all disks except the two which will recieve the influence lines and the AASHTO rolling load files. This program requires only the selection of an option and following the instructions as they appear.

The second utility program permits the user to obtain a hard copy of any data file stored on a data disk which is generated by this package. Again, the user need only select an option and identify the file to be printed. Placing the print routines in a separate program permits the user to complete all calculations without having to wait for printing at each stage. At the same time this program can be used at any time if a copy of intermediate results is desired and then the computations can be continued. This permits maximum flexibility with a minimum of confusion.

Summary

This package is designed to permit the user to pursue any portion of the analysis of this structure desired. Nearly any loading condition may be examined. Special cases of AASHTO loading and support displacement may be examined. Throughout all of the programs in the package, the maximum amount of automation is sought in making computations. At the same time, engineering decisions are left to the user. Breaking the package into several individual programs was originally done because of memory constraints. In the final analysis, it has given the user a very flexible tool for analysis.

CHAPTER 4

Verification of the Package

Throughout the program design process, test structures were run to ensure that the output of each individual program was correct. After completion of the design of the package, an analysis of a typical structure was run. The final product is a refinement of the lessons learned in each of these runs.

In the early work, structures needed to be devised which could be run within the constraints of the program and still be calculated by hand. Because of the complexity of the type of bridge which the programs are intended to analyze, examples of similar structures with complete solutions are not readily available. Therefore small structures which could be calculated by hand were needed. One such structure (Fig 4.1) was designed to contain the required support conditions and the minimum number of members. An exact solution could be hand calculated using the same steps and procedures as the program. This program was used extensively in refining the first five programs of the package, the core of the analysis.

Other test programs were needed to test the package for three major reasons. First, when a single structure is used to debug and refine a program, there is always a possibility that

TRIAL STRUCTURES

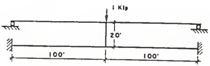
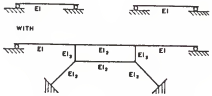


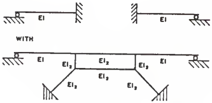
FIG. 4.1

SIMULATE



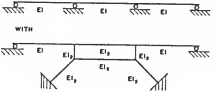
$EI = EI$
$EI_2 = EI / 100$ FIG. 4.2

SIMULATE



$EI = EI$
$EI_2 = 50\ EI$ FIG. 4.3

SIMULATE



$EI = EI$
$EI_2 = EI / 100$ FIG. 4.4

Trial Structures

the program may become oriented to the solution of that particular program. Second, the symmetry and orthogonality of the members of this structure are not typical of the type of structures for which the package is designed. Problems could remain in the package which would not surface under these conditions. Third, the test structure is the smallest for which this pack could be used. As such, problems related to increased size could pass undiscovered.

Therefore, several structures were designed (Figures 4.2, 4.3 & 4.4) which would meet the package requirements of support conditions and geometry while behaving essentially like much simpler structures. This was accomplished by assigning very large or very small relative stiffness to selected members. When the results obtained from the computer package were correct, the program was tested with larger more complex structures.

Several structures were used which varied in size from four to thirteen spans, taking the form of Figure 1.1. This phase of testing revealed several weaknesses with the program concept and the Apple II system. The memory of the computer restrained the use of this package to a twelve span structure (as discussed in Chapter 3). Systems which have been marketed since the start of this project, such as the Zenith model Z-100 and the IBM PC, have much expanded memory which would raise this limit considerably. The Apple II system stores data files as alphanumerics. This uses much more disk space than is necessary. The Apple II also limits the user to 102 separate files to be stored on one disk. These last two problems necessitated the development of error handling routines in each program which prevent the loss of data while permitting the user to take remedial steps such as changing disks. The Apple II DOS was found to be slow in storing and retrieving data, taking ten minutes to retrieve the stiffness file for a six span structure. Because of the rapid development in micro-computers, models are available which eliminate or greatly reduce these difficulties. Because the package is written in basic, it can be transferred to these systems with a minimum of change.

In addition to the constraints detailed above, limitations within the program are revealed when analyzing the large structure. The main limitation is time. The combination of the use of Basic and the Apple II DOS result in very long run times for large structures. Much of the time required does not require monitoring by the user. With experience, this permits other activities to be carried out during these periods so that little actual keyboard time is required. In spite of this, it will take approximately seventeen hours to make a complete analysis of a six span structure. Of this, the final twelve hours are used to find the design values for the AASHTO rolling load. This is completely automatic and may be run unattended. Compiling the programs would reduce this time considerably, as would use of a different system.

While there is no size limit inherent in the logic of the package, the restrictions of the system limit the practical use of this program to a six span bridge. Beyond this, the user would lose the ability to leave the program unattended during what would otherwise be automatic computations. It would be necessary to either constantly change data disks or manually input each load case. This restriction severly limits the prsctical application of the package for design use.

Because of these restrictions, the final sample problem run through the package is a six spsn bridge (fig 5.1). This example is discussed in detail in Chapter 5 and representative results are included in Appendix B. With the analysis functions of the package verified, this example was used primarily to test and refine the user friendly aspects of the programs. The final results of this last example were also examined to verify that the analysis was functioning properly. This final verification was primarily accomplished by examination of the influence lines and comparison of then with characteristic valuesfor similar structures. The lines correlate very well with what is expected. The ability to generate these influence lines is considered to be one of the strongest features of this package. Their use can save the engineer considerable time as well as provide an invaluable aid to understanding the effects of loading on the internal forces in any given member.

Another strength of the package is that the design engineer may now use the package to explore the effects on the structure caused by changing one or more members. By judicious use of the package, the user is able to spend more time understanding the structure and exploring design options instead of performing lengthy and complex analysis computations.

Chapter 5

Descriptive Example

The final example structure used in the development of this package of programs is used here to demonstrate to the reader the interactive feature of the package. The structure selected is a six span bridge composed of members whose dimensions are realistic for this type of structure. The geometry and structural description are shown on Figure 5.1. A step by step description of the analysis of this structure is described here. The questions and remarks displayed on the video monitor are shown in capitals. The responses which the user makes from the keyboard are shown in small letters between quotes. Comments on the procedure are in parenthesis. The results of this example are included in Appendix B.

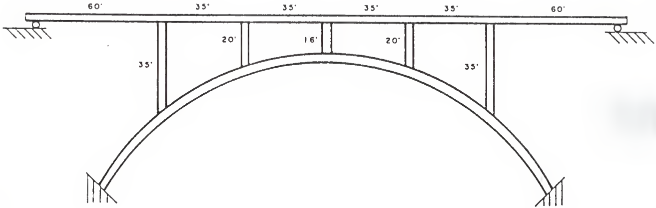
The user places the program disk in Drive 1 of the computer and turns it on. The first screen displays the title page in the same format as seen on page A-1. Striking any key gives the user a series of options. It is here that the program begins.

YOUR OPTIONS ARE:

(1) CATALOG OF DISK IN DRIVE #1

(2) CATALOG OF DISK IN DRIVE #2

TEST STRUCTURE



BEAM	W 36 x 194	A = 57.2 $In.^2$	I = 12100 $In.^4$
COLUMNS	W 14 x 90	A = 29.1 $In.^2$	I = 1110 $In.^4$
ARCH. RIBS	W 14 x 257	A = 75.6 $In.^2$	I = 3400 $In.^4$

FIG. 5-1

(3) CREATE OR MODIFY A BRIDGE

(4) CREATE NEW LOADING FOR AN EXISTING STRUCTURE

(5) RUN OR LOAD ANY PROGRAM

SELECT YOUR OPTION: "3"

Properties Input Program

(New Screen)

IN THIS PROGRAM, YOU WILL INPUT THE GEOMETRIC AND STRUCTURAL PROPERTIES OF YOUR STRUCTURE. THIS DATA WILL BE STORED ON A DISK IN DRIVE #2

HOW MANY CLEAR SPANS DOES THE CONTINOUS BEAM HAVE? "6"

HOW MANY CLEAR SPANS DOES THE ARCH HAVE? "6"

(New Screen)

INPUT GEOMETRIC PROPERTIES

WHAT ARE THE LENGTHS OF EACH CLEAR SPAN OF THE BEAM (NUMBERING FROM LEFT TO RIGHT)?

LENGTH OF SPAN #1 "60" (This continues until all lengths have been entered)

WHAT ARE THE LENGTHS OF THE COLUMNS (NUMBERING FROM LEFT TO RIGHT)?

LENGTH OF COLUMN #1 "35" (This message repeats until all lengths have been entered)

HOW FAR TO THE LEFT OF THE FIRST COLUMN IS THE LEFT ARCH SUPPORT?
"35"
HOW FAR BELOW THE BEAM IS THE LEFT ARCH SUPPORT?
"60"
HOW FAR TO THE RIGHT OF THE LAST COLUMN IS THE RIGHT ARCH SUPPORT? "35":HOW FAR BELOW THE BEAM IS THE RIGHT ARCH SUPPORT?
"60"

(New Screen)
INPUT THE STRUCTURAL PROPERTIES OF THE MEMBERS

ARE THE CROSS-SECTION PROPERTIES THE SAME FOR EACH SPAN OF THE BEAM
"Y" (If the answer is 'N' then the computer will ask for each member)

EI FOR THE BEAM = "$3.51 * 10^{11}$"
AE FOR THE BEAM = "$1.659 * 10^{9}$"

(The same procedure is repeated for the columns and archs. The values which are input are: $3.219 * 10^{10}$, $8.439 * 10^{8}$, $9.86 * 10^{10}$, $2.192 * 10^{9}$)

DO YOU WANT TO REVIEW YOUR INPUT DATA? "Y"

(The input data is printed to the screen or printer at the users option)

EXAMINE THE INPUT DATA CAREFULLY

DO YOU WANT TO CHANGE ANY OF THE INPUT DATA? "N"

(Changes may be entered)

(New Screen)

****STORAGE ROUTINE****

CHECK TO BE SURE THAT YOU HAVE A DATA DISK IN DRIVE #2

TYPE THE TITLE/NUMBER WHICH YOU WANT TO ASSIGN TO THIS INPUT FILE

INPUT FILE- "6S"

(The computer automatically adds the words 'Input file-' to file name for the output from this program)

(This completes the data input program)

Flexibility Matrix Generation

(New Screen)

THIS PROGRAM TAKES ANY DESIRED INPUT FILE AND CALCULATES THE STIFFNESS MATRIX AND ITS INVERSE. THE INVERSE IS STORED ON A DATA DISK IN DRIVE #2

DO YOU WISH TO SEE THE CATALOG FOR DRIVE #2? "N"

TYPE THE TITLE/NUMBER OF THE INPUT FILE TO BE USED:

INPUT FILE- "6S"

CALCULATING THE BEAM STIFFNESS (These messages appear to mark the progress of the program as this is a long operation)

CALCULATING THE COLUMN STIFFNESS

CALCULATING THE ARCH STIFFNESS

##WARNING## DO NO RESET!!

THIS SUBROUTINE TAKES TIME! (This is the inverse routine which is about twenty minutes for this structure)

K-INVERSE HAS BEEN CALCULATED

WHAT TITLE/NUMBER DO YOU WANT TO ASSIGN TO THIS MATRIX?

STIFFNESS MATRIX- "6S" (The prefix is again added automatically)

(This completes the automatic generation of the inverse of the stiffness matrix)

<u>Loading Input Routine</u>

##LOADING ROUTINE##

THIS PROGRAM ALLOWS LOADING TO BE INPUT IN ONE OF FOUR DIFFERENT OPTION OR A COMBINATION OF THEM

SIGN CONVENTION - DOWNWARD, TO THE RIGHT AND COUNTERCLOCKWISE ARE POSITIVE WHEN ENTERING LOADS.

#NOTE# WHEN USING FIXED-END MOMENTS AND SHEARS, POSITIVE IS - UPWARD, TO THE RIGHT AND COUNTERCLOCKWISE.

SELECT YOUR OPTION:

(1) AUTOMATIC INFLUENCE LINE CASES

(2) FIXED-END MOMENTS AND SHEARS

(3) UNIFORM OR CONCENTRATED LOADS

(4) JOINT LOADING

(5) A COMBINATION OF THE ABOVE

"1" (Influence lines are needed to examine AASHTO live loading)

(New Screen)

THIS PROGRAM REQUIRES AND INPUT FILE TO BE PRESENT ON A DATA DISK IN DRIVE #2.

DO YOU WANT TO SEE THE CATALOG OF THAT DISK? "N"

TYPE THE NAME/NUMBER OF THE INPUT FILE TO BE USED:

INPUT FILE- "6S"

MEMBER LOADING

TYPE THE TITLE/NUMBER TO BE USED TO IDENTIFY THIS SERIES OF LOAD CASES:

LOAD CASE- "6S" (The computer will automatically add the 'Load Case- ' to the front of the file name and add a sequence number as a suffix for each file. For this structure twenty-four load cases will be required to generate influence lines.)

SPAN #5, INFLUENCE LOAD CASE #3 (This an example of the information message which marks the progress of the automatic load case generation)

DO YOU HAVE ADDITIONAL LOAD CASES BASED ON THE SAME INPUT FILE? "N" (This option is available no matter which load option was selected at the beginning)

Joint Displacements

(New Screen)

** GLOBAL DISPLACEMENT CALCULATION **

THE K INVERSE FILE AND ONE OR MORE LOAD CASES ARE THE INPUT FOR THIS PROGRAM.

#NOTE# IT IS IMPORTANT THAT THESE TWO FILES ARE BASED ON THE SAME INPUT FILE (The program checks this to be sure.)

THE DISPLACEMENT VECTOR CORRESPONDING TO THE LOAD CASE IS CALCULATED AND STORED ON THE DATA DISK IN DRIVE #2

DO YOU WANT TO SEE THE CATALOG OF THE DISK IN DRIVE #2? "N"

TYPE THE NAME/NUMBER OF THE STIFFNESS FILE TO BE USED FOR THIS CASE: STIFFNESS FILE- "6S"

ARE YOU CALCULATING INFLUENCE LINES? "Y"

THE PROCESS FOR CALCULATING THE DISPLACEMENT VECTORS IS AUTOMATIC HOWEVER, THE MEMORY SPACE IS CRITICAL IN THIS PHASE. THEREFORE EACH DELTA VECTOR MUST BE STORED AS IT IS CALCULATED.

SELECT ONE;

(1) STORE ON THE DATA DISK CURRENTLY IN DRIVE #2 (MUCH PREFERRED)

(2) STORE ON A FRESH DISK IN DRIVE #1 (If this option is selected, a message will direct the user to place a disk with storage space available in drive #1)

HOW MANY BEAM SPANS IN THIS STRUCTURE? "6"
TYPE THE TITLE/NUMBER OF THE LOAD CASE SERIES TO BE USED IN THIS PROGRAM: LOAD CASE- "6S"
WHAT TITLE/NUMBER DO YOU WANT TO USE TO USE TO IDENTIFY THIS SERIES OF DISPLACEMENT VECTORS?
DISPLACEMENT FILE- "6S" (Although '6S' was used to identify all files relating to this example, it is not necessary.)
CALCULATING DISPLACEMENTS FOR LOAD CASE- 6S14 (This message marks the progress of the automatic displacement vector calculation. Storage is also automatic, permitting the user to leave the terminal as this is another long process.)

DO YOU HAVE ADDITIONAL CASES USING THE SAME STIFFNESS MATRIX? "N"

Internal Force Calculation

(New Screen)
THIS PROGRAM TAKES CALCULATED DEFLECTIONS AND CORRESPONDING INPUT FILES AND CALCULATES THE FORCES ON THE ENDS OF THE MEMBERS. THE INTERNAL FORCE MATRIX IS STORED.
THIS PROGRAM USES AND INPUT FILE AND ONE OR MORE DEFLECTION FILES TO CALCULATE THE INTERNAL FORCES FOR EACH MEMBER OF THE BRIDGE

DO YOU WISH TO SEE THE CATALOG OF THE DATA DISK IN DRIVE #2? "N"

** INTERNAL FORCE CALCULATION **

ARE YOU CALCULATING INFLUENCE LINES? "Y"

TYPE THE NAME/NUMBER OF THE INPUT FILE TO BE USED:

INPUT FILE- "6S"

TYPE THE NAME/NUMBER OF THE LOAD CASE(S) TO BE USED:

LOAD CASE- "6S"

TYPE THE NAME/NUMBER OF THE DISPLACEMENTS FILE(S) TO BE USED:

DISPLACEMENT FILE- "6S"

ARE THE LOAD CASE FILES AND THE DISPLACEMENT FILES ON THE SAME DATA DISK, DIFFERENT DISKS OR A MIXTURE? (S/D/M) "S" (For other situations, messages appear to direct the correct placement of the data disks.)

CALCULATING INTERNAL FORCES FOR LOAD CASE 6S10 (This message marks the progress of the automatic calculations)

FORCE CALCULATIONS ARE COMPLETE. ENSURE THAT THE PROGRAM DISK IS IN DRIVE #1. PRESS RETURN WHEN READY.

Influence Line Generation

(New Screen)

THIS PROGRAM ASSEMBLES THE INFLUENCE COEFFICIENT VALUES AND USES THEM TO CALCULATE THE INFLUENCE LINES. AN INPUT FILE AND A COMPLETE SET OF INTERNAL FORCE FILES ARE NEEDED TO RUN THIS PROGRAM. A CLEAN INITIALIZED DISK IS NEEDED TO STORE THE INFLUENCE LINE FILES. IT IS DESIRABLE TO HAVE A COPY OF THE

INPUT FILE ON THE DATA DISK. AT THIS TIME THE DATA DISK IN DRIVE #2 SHOULD CONTAIN AN INPUT FILE AND THE SET OF FORCE FILES.

TYPE THE TITLE/NUMBER OF THE INPUT FILE TO USED:

INPUT FILE- "6S"

TYPE THE TITLE/NUMBER OF THE FORCE FILES TO BE USED:

FORCE FILE- "6S"

RETRIEVING FORCE FILES

TYPE THE TITLE/NUMBER YOU WANT TO USE TO IDENTIFY THIS SET OF INFLUENCE LINES: INFLUENCE LINE- "6S"

A CLEAN DISK MUST BE PLACED IN DRIVE #2 TO RECIEVE THE INFLUENCE FILES. PRESS RETURN WHEN READY.

CALCULATING INFLUENCE LINE #56 (This message marks the progress of the program)

AASHTO Rolling Load

(New Screen)

THIS PROGRAM USES AN INPUT FILE AND A SETIES OF INFLUENCE LINES TO CALCULATE THE AASHTO HS20 ROLLING LIVE LOADS.

THE DATA DISK IN DRIVE #2 SHOULD CONTAIN AN INPUT FILE AND THE INFLUENCE LINE FILES TO BE USED. REPLACE THE PROGRAM DISK WITH A CLEAN DATA DISK IN DRIVE #1.

DO YOU WISH TO CALCULATE AND STORE THE AASHTO ROLLING LOAD INFLUENCE LINES? "Y"

#NOTE# THIS IPTION WILL REQUIRE AT LEAST ONE ADDITIONAL DATA DISK.

DO YOU WISH TO STORE THE INFLUENCE LINES? "Y"

TYPE THE TITLE/NUMBER OF THE INPUT FILE TO BE USED:

INPUT FILE- "6S"

WHAT TRAILOR LENGTH IS TO BE USED IN THIS RUN? "30"

TYPE THE TITLE/NUMBER OF THE INFLUENCE LINE FILES TO BE USED:

INFLUENCE LINE- "6S"

CALCULATING AASHTO FILE 75 (This message marks the progress of the program)

THE MAXIMUM AND MINIMUM VALUES WILL BE STORED ON THE DISK IN DRIVE #2. TYPE THE TITLE/NUMBER YOU WANT TO USE TO IDENTIFY THIS AASHTO FILE: AASHTO MAXIMUMS- "6S"

AASHTO Equivalent Loading

(New Screen)·

THIS PROGRAM USES AN INPUT FILE AND A SET OF INFLUENCE LINE FILES TO CALCULATE THE MAXIMUM POSITIVE AND NEGATIVE INTERNAL FORCES ON EACH MEMBER.

IN EACH CASE THE STRUCTURE IS LOADED WITH THE 'WORST CASE' AASHTO HS20 EQUIVALENT LOADING. THE DISK IN DRIVE #2 WHOULD CONTAIN THE INPUT FILE AND THE SET OF INFLUENCE LINES TO BE USED.

TYPE THE TITLE/NUMBER OF THE INPUT FILE TO BE USED:

INPUT FILE- "6S"

TYPE THE TITLE/NUMBER OF THE SET OF INFLUENCE LINES TO BE USED:

INFLUENCE LINE- "6S"

ALL OF THE OUTPUT FOR THIS PROGRAM WILL BE STORED IN ONE FILE. TYPE THE TITLE/NUMBER YOU WANT TO USE TO IDNETIFY THIS AASHTO FILE: EQUIVALENT LOAD FILE- "6S"

This completes the analysis of the structure. For this particular case the total run time will be about six hours. During this time there are very few demands on the user due to the automation of the calculations. Note that only those portions enclosed by quotations need be entered from the keyboard. In order to examine the output, the user need only run the utility program, PRINTER, and select the output to be printed.

Examples of the output from this program are included in Appendix B. The input file is shown as it is stored. The displacement vectors and influence lines are stored as numeric files. Since they are difficult to visualize, they are presented here in graphic form. AASHTO rolling load influence lines are not normally calculated in conventional analysis, so two of these lines are presented for comparison with the conventional unit load influence lines. Exagerated displacement diagrams are presented for two of the fixed-end load cases to show how these loads affect the structure. Finally, a comparison table is presented for the internal forces resulting from the AASHTO rolling loads and equivalent loads. It can be seen this package provides a powerful tool for the analysis of this type of structure. It also enables the user to examine numerous aspects of the problem which have not been looked at before because of expense or time. Finally, it provides a flexible tool for research into the stresses on this type of structure.

CHAPTER 6

Concluding Remarks

There were two major objectives in the development of this battery of programs. First, it was desired to demonstrate that a complex and lengthy analysis could be programmed on a microcomputer. Second, it was desired to make such a program usable to the practicing engineer by making it 'friendly'. Several secondary goals were also to be examined simultaneously with the accomplishment of the primary goals. One such goal was the analysis of the behavior of a continous beam bridge supported by a fixed arch. This included the use and comparison of the AASHTO codes for live loads on the structure. Another secondary area of interest was analysis of the effects on the structure of support displacement. Finally, while not planned, an examination of the limitations of the Apple II system for this type of application was made.

It is shown that programming an analysis of a complex rigid frame using an exact method, such as displacement theory, on microcomputers is possible. The size or complexity of the structure is restricted by the memory capacity of the system to be used. It is shown that the memory restrictions can be reduced by breaking the computation functions into groups or individual programs. By doing so, it was found that a structure three times as large could be analyzed. A secondary benefit of breaking the program into a package of several individual programs, is the ability to examine several facets of the problem without increasing the memory demands. This is shown in the special purpose programs which analyze the different AASHTO code loadings and the support displacements. A final benefit of the package concept was the storage of output data files at each computation step. This permits the user to exam several variations of a given problem without reworking the initial phases. It also provides a permanent record of the results of each calculation step for review or checking. Within the limitations noted here and below, it is felt that this concept works well.

The degree to which this package is user friendly must be judged by a user not familiar with the inner workings and possible pitfalls. The interactive portion of the programs are designed with this in mind. First, a balance is struck which automates as much of the analysis as possible while permitting the user flexibility. This is accomplished by permitting the user to select all of the structural and geometric properties of the structure, as well as any combination of loading desired. These are the two areas where engineering judgement is required. All of the rest of the calculations are handled automatically. During the automatic portions, messages are printed to the screen which notify the user of the progress of the computations. Questions and user input formats are standardized to minimize confusion about what is needed. Special check routines are included in every program to protect the user from destroying his own data. These routines check for anticipated user errors and print notification on the monitor of the problem so that it can be corrected without loss of the work to that point. Other potential errors are eliminated by minimizing user input to those items over which an engineer would desire to retain control. All other data is calculated within the package.

Disappointment with the Apple II system is a result of difficulties encountered during the programming and testing, and the knowledge that much superior systems are now available. Memory restrictions cannot be attributed to the Apple II as any 48K system would be subject to the same limitations. However, the Apple II DOS creates serious problems in the handling and storage of data on disks. Data storage is found to be inefficient and slow, placing limits on the practical application of the package which would otherwise not be present. While memory limits the package to a twelve span structure, the Apple II DOS limits it to a six span bridge for practical use. The DOS makes disk manipulations which are excessive and annoying, necessary for structures of more than six spans.The memory limitation can be lessened by adopting an improved method of solution. One such method would be to arrange the unknowns of the structure to yield a banded stiffness matrix with a minimum band width and solve for the displacements directly without inverting the stiffness matrix. Programming this method is a major research effort in itself. The Apple II system is not recommended for this application.

While this package has immediate practical applications, it should be considered as the starting point for development of improved and expanded models. The first improvement recommended is to transfer it to the Zenith Z100 or a similar system. Tripled memory, increased speed and improved data storage are only a few of the advantages which would accrue.

Within the existing package there are areas which could be improved or expanded. Memory use could be made more efficient by storing and manipulating only the nonzero diagonal of the flexibility matrix. This is a very complicated procedure and would require a long period of development. User friendliness could be improved by creating an 'option' file in the first program. This file would be created in the 'hello' program where the user would make procedure and option selections in detail. The file would be called in each program where the options would be read instead of being input from the keyboard. This would further increase the efficiency and automation of the analysis. The input of the structural geometry would be changed so that the bridge deck was no longer restricted to a pure horizontal orientation. This would also require changes in the program handling structural loading, but would increase the practical application. Finally, compiling the BASIC code into machine language code would increase the speed 20 to 40 times and reduce the major weakness of the package. It is also recognized that each program contains small inefficiencies which could be eliminated or improved by more research and development work. For example, the program which calculates the maximum and minimum live loading due to the AASHTO rolling load could be modified so that it uses a variable trailor spacing to find the most critical value.

This package could also be used as a framework for the development of other analysis programs of other structures. This would mainly involve changes to the boundary constraints and geometric assumptions. It would not be very practical to try to adapt this package to a different structure. It would be better to use the logic and processes developed here as a framework for such a problem. Certain of the routines developed in this package could be used with little or no change, but great care would be required. Of course the ultimate program would be general and permit the user to input any boundary constraints and solve the general case.

Hello Program:

```
1     D$ = ""
5     REM ##12/28/83##
10    PRINT "========================================"
11    PRINT "II                                   II"
13    PRINT "II     A BATTERY OF STRUCTURAL       II"
14    PRINT "II     ANALYSIS PROGRAMS FOR THE     II"
15    PRINT "II     SOLUTION OF A CONTINUOUS      II"
16    PRINT "II     BEAM BRIDGE SUPPORTED BY A    II"
17    PRINT "II            FIXED ARCH             II"
18    PRINT "II                                   II"
19    PRINT "II     CREATED BY JOHN J. KOONTZ     II"
21    PRINT "II               1983                II"
23    PRINT "========================================"
24    FOR I = 1 TO 10000: NEXT
25    HOME : VTAB (5): PRINT "YOUR OPTIONS ARE:": PRINT : PRINT
          SPC( 5)"(1) CATALOG OF DISK IN DRIVE #1"
26    PRINT  SPC( 5)"(2) CATALOG OF DISK IN DRIVE #2"
27    PRINT  SPC( 5)"(3) CREATE OR MODIFY A BRIDGE"
28    PRINT  SPC( 5)"(4) CREATE NEW LOADING FOR AN": PRINT
          SPC( 9)"EXISTING STRUCTURE"
29    PRINT  SPC( 5)"(5) RUN OR LOAD ANY PROGRAM": PRINT  SPC( 9)"ON
          THIS DISK"
30    PRINT : PRINT "SELECT YOUR OPTION: ": VTAB (15): HTAB (21): GET O
31    PRINT : IF O > 1 THEN 34
32    HOME : PRINT D$"CATALOG,D1"
33    PRINT "PRESS RETURN FOR THE MENU": GET O$: GOTO 25
34    IF O > 2 THEN 40
35    HOME : PRINT D$"CATALOG,D2"
36    PRINT "PRESS RETURN FOR THE MENU": GET O$: GOTO 25
40    IF O > 3 THEN 45
42    PRINT D$"RUN PROPERTIES,D1"
45    IF O > 4 THEN 110
50    PRINT D$"RUN LOADING,D1"
110   PRINT : PRINT  TAB( 5)"(N) NEW": PRINT  TAB( 5)"(R) RUN": PRINT
          TAB( 5)"(L) LOAD": GET O$
120   IF O$ = "N" GOTO 170
```

```
130   INPUT "PROGRAM NAME: ";PN$
140   IF O$ = "L" THEN O$ = "LOAD"
150   IF O$ = "R" THEN O$ = "RUN"
160   PRINT D$;O$;PN$
170   HOME : NEW
```

Structural Data Input Program:

```
5      REM ##11/14/83##
7      ONERR  GOTO 5000
8      HOME : VTAB (6)
9      PRINT "IN THIS PROGRAM, YOU WILL INPUT THE": PRINT "GEOMETRIC AND
            STRUCTURAL PROPERTIES OF": PRINT "YOUR STRUCTURE": PRINT :
            PRINT "THIS DATA WILL BE STORED ON A DISK IN": PRINT "IN
            DRIVE #2"
15     REM  SPECIAL PROJECT- ARCH REINFORCED WITH A CONTINUOUS BEAM
            (GENERAL CASE)
20     D$ = ""
25     REM  **DEFINE DIMENSIONS**
30     PRINT : INPUT "HOW MANY CLEAR SPANS DOES THE CONTINUOUS BEAM
            HAVE? ";NS
35     NC = NS - 1:CC = 0
40     PRINT : INPUT "HOW MANY CLEAR SPANS DOES THE ARCH HAVE?";NA
45     IF NA = NS THEN A$ = "N":CC = 1: GOTO 55
50     A$ = "Y"
55     NB = NS + NA + NC:NJ = NS + NA + 2
60     NJ = NS + NA + 2
65     N = 3 * NJ - 8
75     DIM SN(NB),CS(NB),L(NB),EI(NB),AE(NB),JN(NB,2),X(NJ),Y(NJ)
80     REM  **DEFINE COORDINATES FOR EACH JOINT**
85     X(1) = 0:Y(1) = 0
87     HOME : PRINT "INPUT GEOMETRIC PROPERTIES:": PRINT
88     PRINT "ALL LENGTHS ARE INPUT IN FT.": PRINT
90     PRINT : PRINT "WHAT ARE THE LENGTHS OF EACH CLEAR SPAN OF THE
            BEAM (NUMBERING FROM LEFT TO RT)?": PRINT
95     FOR I = 1 TO NS
100    PRINT "LENGTH OF SPAN # "I
105    INPUT L(I)
110    X(I + 1) = X(I) + L(I):Y(I + 1) = 0
115    NEXT I
120    PRINT : PRINT "WHAT ARE THE LENGTHS OF THE COLUMNS   (NUMBERING
            FROM LEFT TO RIGHT)?": PRINT
125    FOR I = 1 TO NC
```

```
130    PRINT "COLUMN # "I
135    INPUT L(I + NS)
140    X(NS + I + 1 + CC) = X(I + 1):Y(NS + CC + I + 1) = - L(I + NS)
145    NEXT I
150    IF A$ = "Y" GOTO 185
160    PRINT : INPUT "HOW FAR TO THE LEFT OF THE FIRST COLUMN IS THE
           LEFT ARCH SUPPORT?";D1
162    PRINT : INPUT "HOW FAR BELOW THE BEAM IS THE LEFT ARCH SUPPORT?";D2
164    X(NS + 2) = L(1) - D1:Y(NS + 2) = - D2
166    PRINT : INPUT "HOW FAR TO THE RIGHT OF THE LAST COLUMN IS THE RIGHT
           ARCH SUPPORT?";D1
168    PRINT : INPUT "HOW FAR BELOW THE BEAM IS THE RIGHT ARCH SUPPORT?";D2
169    REM THE SHAPE OF THE ARCH IS DETERMINED BY THE LOCATIONS OF THE
           SUPPORTS AND THE LENGTHS OF THE COLUMNS.
170    X(NJ) = X(NJ - 1) + D1:Y(NJ) = - D2
180    REM  **ESTABLISH JOINT NUMBERS FOR EACH MEMBER**
185    FOR I = 1 TO NS
190    JN(I,1) = I:JN(I,2) = I + 1
195    NEXT I
200    FOR I = 1 TO NC
205    II = I + NS
210    JN(II,1) = I + 1:JN(II,2) = II + 1 + CC
215    NEXT I
220    FOR I = 1 TO NA
225    II = NC + NS + I
230    JN(II,1) = NS + I + 1:JN(II,2) = NS + I + 2
235    NEXT I
240    REM  **CALCULATE LENGTHS AND TRIG FUNCTIONS FOR EACH MEMBER**
245    FOR I = 1 TO NB
250    DX = (X(JN(I,2)) - X(JN(I,1))) * 12
255    DY = (Y(JN(I,2)) - Y(JN(I,1))) * 12
260    L(I) = ( SQR (DX ↑ 2 + DY ↑ 2))
265    SN(I) = DY / L(I):CS(I) = DX / L(I)
270    NEXT I
275    REM  **INPUT STRUCTURAL PROPERTIES**
276    IF CG$ = "G" THEN 490
```

```
280   HOME : PRINT "INPUT THE STRUCTURAL PROPERTIES OF THE MEMBERS":
          PRINT : PRINT
282   PRINT "UNITS FOR E ARE PSI, A ARE IN↑2, I ARE IN↑4": PRINT
285   INPUT "ARE THE CROSS-SECTION PROPERTIES THE SAME FOR EACH SPAN
          OF THE BEAM?";B$
290   IF B$ = "N" THEN 320
295   PRINT : INPUT "EI FOR THE BEAM = ";EI
300   PRINT : INPUT "AE FOR THE BEAM = ";AE
305   FOR I = 1 TO NS
310   EI(I) = EI:AE(I) = AE
315   NEXT I: GOTO 350
320   FOR I = 1 TO NS
325   PRINT "EI FOR SPAN # "I" OF THE CONTINUOUS BEAM"
330   INPUT EI(I)
335   PRINT "AE FOR SPAN # "I" OF THE CONTINUOUS BEAM"
340   INPUT AE(I)
345   NEXT I
350   PRINT : INPUT "ARE THE CROSS-SECTION PROPERTIES THE SAME FOR ALL
          COLUMNS?";B$
355   IF B$ = "N" THEN 385
360   PRINT : INPUT "EI FOR THE COLUMNS = ";EI
365   PRINT " INPUT "AE FOR THE COLUMNS + ";AE
370   FOR I = NS + 1 TO NS + NC
375   EI(I) = EI:AE(I) = AE
380   NEXT I: GOTO 420
385   FOR I = 1 TO NC
390   II = NS + I
395   PRINT "EI FOR COLUMN # "I
400   INPUT EI(II)
405   PRINT "AE FOR COLUMN # "I
410   INPUT AE(II)
415   NEXT I
420   PRINT : INPUT "ARE THE CROSS-SECTION PROPERTIES THE SAME FOR EACH
          SPAN OF THE ARCH?";B$
425   IF B$ = "N" THEN 455
430   PRINT : INPUT "EI FOR THE ARCH = ";EI
```

```
435    PRINT : INPUT "AE FOR THE ARCH = ";AE
440    FOR I = NS + NC + 1 TO NS + NC + NA
445    EI(I) = EI:AE(I) = AE
450    NEXT I: GOTO 490
455    PRINT
457    FOR I = 1 TO NA
460    II = NS + NC + I
465    PRINT "EI FOR THE ARCH SPAN # "I
470    INPUT EI(II)
475    PRINT "AE FOR THE ARCH SPAN # "I
480    INPUT AE(II)
485    NEXT I
490    HOME : INPUT "DO YOU WANT TO REVIEW YOUR INPUT DATA?";B$
495    IF B$ = "N" THEN 550
496    PRINT : PRINT "DO YOU WANT THE INPUT DATA TO BE PRINTED TO THE
            SCREEN OR PRINTER? (S/P) ": GET O$
497    IF O$ = "P" THEN PR# 1
499    PRINT : PRINT : PRINT "MEMBER" SPC( 3)"LENGTH(FT)" SPC( 5)"EI"
            SPC( 9)"EA"
500    FOR I = 1 TO NB
505    IF I = 1 THEN 510
506    IF I = NS + 1 THEN 515
507    IF I = NS + NC + 1 THEN 520
508    GOTO 522
510    PRINT : PRINT "BEAM SPAN #": PRINT :J = I: GOTO 530
515    PRINT : PRINT "COLUMN #": PRINT :J = I - NS: GOTO 530
520    PRINT : PRINT "ARCH SPAN #": PRINT :J = I - (NS + NC): GOTO 530
522    IF I < = NS THEN J = I: GOTO 530
524    IF I < = NS + NC THEN J = I - NS: GOTO 530
526    J = I - (NS + NC)
530    PRINT J; SPC( 6);L(I); SPC( 4);EI(I) SPC( 4)AE(I)
535    NEXT I
540    PR# 0
545    PRINT : PRINT "EXAMINE THE INPUT DATA CAREFULLY"
550    PRINT : INPUT "DO YOU WANT TO CHANGE ANY OF THE INPUT DATA? ";B$
555    IF B$ = "N" THEN 700
```

```
556   PRINT : PRINT "DO YOU WANT TO MAKE GEOMETRIC OR STRUCTURAL CHANGES?
          (G/S) ": GET CG$
557   IF CG$ = "G" THEN 87
560   PRINT : INPUT "CHANGES FOR ANY BEAM SPANS? ";B$
562   IF B$ = "N" THEN 570
564   PRINT : INPUT "WHICH SPAN? ";J
566   C$ = "B": GOSUB 600
567   IF E$ = "Y" THEN 564
570   PRINT : INPUT "CHANGES FOR ANY COLUMNS? ";B$
572   IF B$ = "N" THEN 580
574   PRINT : INPUT "WHICH COLUMN? ";J
576   C$ = "C": GOSUB 600
577   IF E$ = "Y" THEN 574
580   PRINT : INPUT "CHANGES FOR ANY ARCH SPANS? ";B$
582   IF A$ = "N" THEN 630
584   PRINT : INPUT "WHICH SPAN? ";J
586   C$ = "A": GOSUB 600
587   IF E$ = "Y" THEN 584
588   GOTO 490
600   IF C$ = "B" THEN I = J:B$ = "BEAM SPANS"
602   IF C$ = "C" THEN I = J + NS:B$ = "COLUMNS"
604   IF C$ = "A" THEN I = J + NS + NC:B$ = "ARCH SPANS"
606   PRINT : INPUT "NEW EI = ";EI(I)
608   PRINT : INPUT "NEW AE = ";AE(I)
630   PRINT : PRINT : PRINT "CHANGES FOR ANY OTHER "B$"? ": INPUT E$
634   RETURN
700   HOME : PRINT "****STORAGE ROUTINE****": PRINT : PRINT : PRINT "CHECK
          TO BE SURE YOU HAVE A DATA DISK IN DRIVE 2!"
702   PRINT : PRINT : PRINT "TYPE THE TITLE/NUMBER WHICH YOU WANT TO
          ASSIGN TO THIS INPUT FILE:": INPUT "INPUT FILE- ";B$
703   B$ = "INPUT FILE- " + B$
704   ER = 0
705   PRINT D$;"OPEN"B$",D2"
710   PRINT D$;'DELETE"B$
712   PRINT D$;"OPEN"B$
715   PRINT D$;"WRITE"B$
```

```
720   PRINT NS: PRINT NA
725   FOR I = 1 TO NB
735   PRINT L(I): PRINT SN(I): PRINT CS(I): PRINT JN(I,1): PRINT JN(I,2):
        PRINT EI(I): PRINT AE(I)
740   IF ER > 0 THEN 704
765   NEXT I
770   PRINT D$;"CLOSE"B$
775   PRINT D$;"RUNK CALC,D1"
800   REM ##DEFINITION OF VARIABLES##
801   REM NJ- NUMBER OF JOINTS
802   REM NB- NUMBER OF MEMBERS
803   REM NS- NUMBER OF BEAM SPANS
804   REM NC- NUMBER OF COLUMNS
805   REM NA- NUMBER OF ARCH SPANS
806   REM D1,D2- DEFINE LOCATION OF THE ARCH SUPPORTS
807   REM CC,A$- SIGNALS WHETHER OR NOT A COLUMN IS LOCATED AT THE ARCH
        SUPPORTS
808   REM II- INDEX CONVERTS COLUMN OR SRCH NUMBER TO STRUCTURAL MEMBER
        NUMBER
809   REM DX,DY- HORIZONTAL AND VERTICAL DEVIATIONS FOR A MEMBER
810   REM SN(I),CS(I)- SINE AND COSINE FUNCTIONS OF MEMBER #I
811   REM L(I),EI(I),AE(I)- LENGTH AND STRUCTURAL PROPERTIES OF MEMBER #I
812   REM JN(I,1),JN(I,2)- JOINT NUMBERS FOR THE LEFT AND RIGHT ENDS OF
        MEMBER #I (TOP AND BOTTOM FOR COLUMNS)
813   REM X(I),Y(I)- GLOBAL COORDINATES FOR JOINT #I
5000  ER = PEEK (222)
5010  IF ER = 6 THEN 5100
5020  IF ER = 9 THEN 5200
5030  IF ER = 13 THEN 5300
5080  HOME : PRINT "ERROR # "ER: END
5100  PRINT : PRINT : PRINT "YOU HAVE EITHER MISTYPED OR INPUT THE NAME
        OF A FILE WHICH DOES NOT EXIST."
5110  PRINT "TRY AGAIN"
5120  PRINT D$;"DELETE"B$:RESUME
5200  PRINT : PRINT : PRINT "THE DATA DISK IS FULL.  REPLACE IT WITH AN
        INITIALIZED DISK WITH SPACE AVAILABLE"
```

```
5210  PRINT : PRINT "HIT RETURN WHEN READY TO CONTINUE": GET OE$
5220  PRINT : RESUME
5300  PRINT : PRINT : PRINT "FILE TYPE MISMATCH": PRINT "RETYPE YOUR
          INPUT.  BE SURE TO USE THE APPROPRIATE NUMBER OR STRING."
5310  RESUME
```

Flexibility Matrix Generation:

```
5     REM          **6/13/1983**
6     D$ = ""
7     ONERR  GOTO 6000
10    REM THIS PROGRAM TAKES ANY DESIRED INPUT FILE AND CALCULATES THE
          STIFFNESS MATRIX AND ITS INVERSE.  THE INVERSE IS THEN STORED
          ON THE DISK
11    HOME ; VTAB (5)
12    PRINT "THIS PROGRAM TAKES ANY DESIRED INPUT FILE AND CALCULATES
          THE STIFFNESS MATRIX AND ITS INVERSE.": PRINT "THE INVERSE
          IS STORED ON A DATA DISK IN DRIVE #2"
13    PRINT : PRINT "DO YOU WISH TO SEE THE CATALOG FOR YOUR DATA DISK IN
          DRIVE #2?": GET O$
14    IF O$ > < "Y" THEN 30
15    D$ = ""
20    HOME : PRINT "THESE ARE THE FILES CURRENTLY ON YOUR DATA DISK:"
25    PRINT : PRINT D$"CATALOG,D2"
30    PRINT : PRINT : PRINT "TYPE THE NAME/NUMBER OF THE INPUT FILE": INPUT
          "TO BE USED: INPUT FILE- ";B$
35    B$ = "INPUT FILE- " + B$
40    PRINT D$;"OPEN";B$;",D2"
45    PRINT D$;"READ";B$
50    INPUT NS
54    INPUT NA
55    NC = NS - 1:NB = NS + NC + NA:NJ = NS + NA + 2:N = 3 * NJ - 8:
          IF NA = NS THEN A$ = "N":CC = 1:  GOTO 65
60    A$ = "Y":CC = 0
65    DIM L(NB),SN(NB),CS(NB),EI(NB),AE(NB),JN(NB,2),K(N,N),ID(N,2)
70    FOR I = 1 TO NB
75    INPUT L(I): INPUT SN(I): INPUT CS(I): INPUT JN(I,1): INPUT JN(I,2):
          INPUT EI(I): INPUT AE(I)
80    NEXT I
90    PRINT D$"CLOSE"B$
95    REM **CONVERT KE TO K FOR THE BEAM**
100   HOME : PRINT "CALCULATING THE BEAM STIFFNESS"
105   GOSUB 3000
```

```
110    CK(1) = 1:CK(2) = 0:CK(3) = 2:CK(4) = 3:CK(5) = 4:CK(6) = 5
115    L = 1: GOSUB 5000
120    GOSUB 3200
125    IF NC = 1 THEN 170
130    FOR L = 2 TO NC
135    GOSUB 3000
140    GOSUB 5000
145    RN = (L - 2) * 3 + 2
150    FOR I = 1 TO 6
155    CK(I) = RN + I: NEXT
160    GOSUB 3200
165    NEXT L
170    IJ = (NS - 1) * 3
175    GOSUB 3000
180    CK(1) = IJ:CK(2) = IJ + 1:CK(3) = IJ + 2:CK(4) = IJ + 3:CK(5) =
           0:CK(6) = IJ + 4
185    L = NS: GOSUB 5000
190    GOSUB 3200
195    REM  **CONVERT KE TO K FOR THE COLUMNS**
200    PRINT : PRINT "CALCULATING THE COLUMN STIFFNESS"
205    A = 2 - CC
210    N2 = NC - A + 1
215    FOR M = A TO N2
220    L = NS + M
225    GOSUB 3000
230    RN = (M - 1) * 3 + 2
235    RL = (NS + M - A) * 3 + 1
240    GOSUB 5000
245    CK(1) = RN + 1:CK(2) = RN + 2:CK(3) = RN + 3
250    CK(4) = RL + 1:CK(5) = RL + 2:CK(6) = RL + 3
255    GOSUB 3200
260    NEXT M
265    REM  **THIS BRANCH ACCOUNTS FOR COLUMNS ATTACHED TO FIXED SUPPORTS**
270    IF A$ = "N" GOTO 320
275    L = NS + 1: GOSUB 3000
280    GOSUB 5000
```

```
285    CK(1) = 3:CK(2) = 4:CK(3) = 5
290    GOSUB 3200
295    RN = (NC - 1) * 3 + 2
300    GOSUB 3000
305    L = NS + NC: GOSUB 5000
310    CK(1) = RN + 1:CK(2) = RN + 2:CK(3) = RN + 3
315    GOSUB 3200
320    REM  **CONVERT KE TO K FOR THE ARCH MEMBERS**
325    PRINT : PRINT "CALCULATING THE ARCH STIFFNESS:
330    N4 = NA - 1
335    IF N4 < 2 THEN 390
340    FOR M = 2 TO N4
345    L = NS + NC + M
350    GOSUB 3000
355    RL = (NS + M - 2) * 3 + 1
360    GOSUB 5000
365    FOR I = 1 TO 6
370    CK(I) = RL + I: NEXT
375    GOSUB 3200
380    NEXT M
385    REM  **ACCOUNT FOR ARCHES ATTACHED TO SUPPORTS**
390    RJ = NS * 3 + 1
395    L = NS + NC + 1
400    GOSUB 3000
405    GOSUB 5000
410    CK(4) = RJ + 1:CK(5) = RJ + 2:CK(6) = RJ + 3
415    GOSUB 3200
420    RJ = (NS + N4 - 1) * 3 + 1
425    L = NB
430    GOSUB 3000
435    GOSUB 5000
440    CK(1) = RJ + 1:CK(2) = RJ + 2:CK(3) = RJ + 3
445    GOSUB 3200
450    HOME : PRINT "##WARNING## DO NOT RESET!!        THIS SUBROUTINE
            TAKES TIME!"
```

```
455   REM  **INVERT THE K MATRIX**
460   GOSUB 4000
465   REM **STORE THE K MATRIX**
470   HOME : PRINT "K-INVERSE HAS BEEN CALCULATED.": PRINT : PRINT :
          INPUT "WHAT NUMBER/TITLE DO YOU WANT TO ASSIGN TO THIS MATRIX?";C$
475   C$ = "STIFFNESS FILE- " + C$
479   ER = 0
480   PRINT D$;"OPEN"C$",D2"
485   PRINT D$;"DELETE"C$
490   PRINT D$;"OPEN"C$
495   PRINT D$;"WRITE"C$
500   PRINT B$: PRINT N
505   FOR I = 1 TO N
510   FOR J = 1 TO N
515   PRINT K(I,J)
519   IF ER > THEN 479
520   NEXT J,I
525   PRINT D$;"CLOSE"C$
530   PRINT D$;"RUN LOADING,D1"
3000  REM  **ZERO CE AND CK**
3005  FOR I = 1 TO 6
3010  CK(I) = 0
3015  FOR J = 1 TO 6
3018  KT(I,J) = 0:KG(I,J) = 0
3020  NEXT J,I
3030  RETURN
3200  REM  **CONVERT KE TO K**
3210  FOR J = 1 TO 6
3220  RK = CK(J)
3230  FOR K = 1 TO 6
3240  CK = CK(K)
3250  K(RK,CK) = K(RK,CK) + KG(J,K)
3260  NEXT K,J
3270  RETURN
4000  REM   **GAUSS JORDAN INVERSE SUBROUTINE**
4130  FOR I = 1 TO N
```

```
4140  ID(I,1) = 0
4150  NEXT I
4155  II = 0
4158  REM LOCATE THE ABSOLUTE MAXIMUM VALUE TO USE AS THE NEXT PIVOT
           (EXCLUDE COLUMNS AND ROWS OF PREVIOUS PIVOTS)
4160  AM = - 1
4165  FOR I = 1 TO N
4170  IF ID(I,1) < > 0 THEN 4210
4175  FOR J = 1 TO N
4177  TP = ABS (K(I,J))
4180  IF ID(J,1) > < 0 THEN 4200
4183  IF TP < AM THEN 4200
4184  REM RECORD THE COLUMN AND ROW OF THE PIVOT ELEMENT
4185  IR = I:IC = J:AM = TP
4200  NEXT J
4210  NEXT I
4215  REM CHECK FOR SINGULARITY
4220  IF AM < 0 THEN 4540
4225  IF AM = 0 THEN 4745
4228  REM MOVE THE PIVOT TO THE DIAGONAL
4230  ID(IC,1) = IR
4240  IF IC = IR THEN 4310
4250  FOR J = 1 TO N
4255  TP = K(IR,J)
4260  K(IR,J) = K(IC,J)
4270  K(IC,J) = TP
4280  NEXT J
4285  REM PERFORM THE GAUSS-JORDAN ELIMINATION
4290  II = II + 1
4300  ID(II,2) = IC
4310  PV = K(IC,IC)
4320  K(IC,IC) = 1
4330  PV = 1 / PV
4340  FOR J = 1 TO N
4350  K(IC,J) = K(IC,J) * PV
4360  NEXT J
```

```
4370   FOR I = 1 TO N
4380   IF I = IC THEN 4440
4390   TP = K(I,IC)
4400   K(I,IC) = 0
4410   FOR J = 1 TO N
4420   K(I,J) = K(I,J) - K(IC,J) * TP
4430   NEXT J
4440   NEXT I
4450   GOTO 4160
4455   REM RESTORE THE ROWS AND COLUMNS TO THEIR ORIGINAL LOCATIONS
4460   IC = ID(II,2)
4470   IR = ID(II,1)
4480   FOR I = 1 TO N
4490   TP = K(I,IR)
4500   K(I,IR) = K(I,IC)
4510   K(I,IC) = TP
4520   NEXT I
4530   II = II - 1
4540   IF II < > 0 THEN 4460
4550   GOTO 4760
4746   PRINT "K MATRIX IS SINGULAR!"
4760   RETURN
5000   REM  **KE-GLOBAL SUBROUTINE**
5010   REM  **INPUT KE-LOCAL**
5020   A1 = AE(L):B = 6 * EI(L) / L(L):C = 12 * EI(L) / (L(L)) ↑ 2
5040   KE(1,1) = A1:KE(1,4) = - A1:KE(4,1) = - A1:KE(4,4) = A1
5050   KE(2,2) = C:KE(2,3) = + B:KE(2,5) = - C:KE(2,6) = + B
5060   KE(3,2) = + B:KE(3,3) = 4 * EI(L):KE(3,5) = -B:KE(3,6) = 2 * EI(L)
5070   KE(5,2) = - C:KE(5,3) = -B:KE(5,5) = C:KE(5,6) = - B
5080   KE(6,2) = +B:KE(6,3) = 2 * EI(L):KE(6,5) = - B:KE(6,6) = 4 * EI(L)
5082   FOR I = 1 TO 6
5084   FOR J = 1 TO 6
5086   KE(I,J) = KI(I,J) / L(L)
5088   NEXT J,I
5100   REM   **INPUT TRANSFORMATION MATRIX (TT)**
5120   T(1,1) = CS(L):T(1,2) = - SN(L)
```

```
5130  T(2,1) = SN(L):T(2,2) = CS(L)
5140  T(3,3) = 1:T(6,6) = 1
5150  T(4,4) = CS(L):T(4,5) = - SN(L)
5160  T(5,4) = SN(L):T(5,5) = CS(L)
5200  REM  **COMPUTE KE-GLOBAL = TT*KE(L)*T **
5220  FOR J = 1 TO 6
5230  FOR K = 1 TO 6
5240  FOR I = 1 TO 6
5250  KT(J,K) = KT(J,K) + KE(J,I) * T(K,I)
5260  NEXT I,K,J
5280  FOR J = 1 TO 6
5290  FOR K = 1 TO 6
5300  FOR I = 1 TO 6
5310  KG(J,K) = KG(J,K) + T(J,I) * KT(I,K)
5320  NEXT I,K,J
5330  RETURN
5999  REM ERROR HANDLING ROUTINE
6000  ER = PEEK (222)
6010  IF ER = 5 THEN 6050: REM END OF DATA
6020  IF ER = 9 THEN 6080: REM DISK FULL
6030  IF ER = 13 THEN 6110: REM FILE TYPE MISMATCH
6040  HOME : PRINT "ERROR # "ER: END
6050  PRINT : PRINT : PRINT "YOU HAVE EITHER MISTYPED OR INPUT THE NAME
          OF A FILE WHICH DOES NOT EXIST."
6060  PRINT "TRY AGAIN"
6070  PRINT D$;"DELETE"B$: GOTO 30
6080  PRINT : PRINT : PRINT "THE DATA DISK IS FULL.  REPLACE IT WITH AN
          INITIALIZED DISK WITH SPACE AVAILABLE"
6090  PRINT : PRINT "HIT RETURN WHEN READY TO CONTINUE": GET OE$
6100  PRINT : RESUME
6110  PRINT : PRINT : PRINT "FILE TYPE MISMATCH": PRINT "RETYPE YOUR
          INPUT.  BE SURE TO USE THE APPROPRIATE NUMBER OR STRING."
6120  RESUME
7000  REM ##DEFINITION OF VARIABLES##
7002  REM NB- NUMBER OF MEMBERS
7003  REM NS- NUMBER OF BEAM SPANS
```

```
7004  REM NC- NUMBER OF COLUMNS
7005  REM NA- NUMBER OF ARCH SPANS
7006  REM CC,A$- SIGNALS WHETHER OR NOT A COLUMN IS LOCATED AT THE ARCH
          SUPPORTS
7007  REM SN(I),CS(I)- SINE AND COSINE FUNCTIONS OF MEMBER #I
7008  REM L(I),EI(I),AE(I)- LENGTH AND STRUCTURAL PROPERTIES OF MEMBER #I
7009  REM JN(I,1),JN(I,2)- JOINT NUMBERS FOR THE LEFT AND RIGHT ENDS OF
          MEMBER #I (TOP AND BOTTOM FOR COLUMNS)
7010  REM (K(I,J) IS USED FOR BOTH THE STIFFNESS MATRIX AND THE INVERSE.
          (THE ORIGINAL STIFFNESS MATRIX IS LOST IN THE INVERSE PROCESS)
7011  REM CK(6) IS USED TO INDEX THE ROWS AND COLUMNS OF THE STRUCTURE
          STIFFNESS MATRIX FOR SUPERPOSITION OF THE INDIVIDUAL MEMBER
          MATRICES
7012  REM T(6,6) IS THE TRANSFORMATION MATRIX CALCULATED FOR EACH MEMBER
7013  REM KE(6,6) IS THE LOCAL MEMBER STIFFNESS MATRIX
7014  REM KT(6,6) IS THE KE MATRIX TRANSFORMED
7015  REM KG(6,6) IS THE KT MATRIX TIMES THE TRANSPOSE OF THE TRANSFORMA-
          TION MATRIX.  THIS IS THE K GLOBAL FOR THE MEMBER
7016  REM L IS THE MEMBER COUNTER
7017  REM IJ,A,N2,N4,RL & RN ARE COUNTERS USED TO INDEX MATRIX LOCATIONS
7018  REM THE FOLLOWING VARIABLES ARE USED IN THE INVERSE ROUTINE
7019  REM ID(I,2) STORES THE ORIGINAL LOCATION AND ORDER OF CHANGE OF THE
          PIVOTS
7020  REM TP,IR,IC & PV ARE TEMPORARY VARIABLES FOR THE PIVOT, LOCATIONS
          OF THE PIVOT AND INVERSE OF THE PIVOT
7021  REM AM IS A SIGNAL TO INDICATE THE END OF THE ELIMINATION
```

Loading Data Input Program:

```
5      REM **11/23/83**
7      ONERR GOTO 3000
10     REM THIS PROGRAM TAKES ANY INPUT FILE AND CALCULATES THE CORRESPONDING
            FORCE VECTOR.  THE FORCE VECTOR IS THEN STORED ON DISK.
15     D$ = ""
16     HOME : VTAB (6): PRINT  SPC( 8);"##LOADING ROUTINE##"
17     PRINT : PRINT : PRINT "THIS PROGRAM ALLOWS LOADING TO BE INPUT IN
            ONE OF FOUR DIFFERENT OPTIONS OR A COMBINATION OF THEM."
18     PRINT : PRINT "*SIGN CONVENTION* - DOWNWARD, TO THE RIGHT AND
            COUNTERCLOCKWISE ARE POSITIVE WHEN ENTERING LOADS, #NOTE#
            WHEN USING FIXED-END MOMENTS AND SHEARS, POSITIVE IS- UPWARD,
            TO THE RIGHT AND COUNTERCLOCKWISE."
20     PRINT : PRINT "SELECT YOUR OPTION:"
21     PRINT : PRINT  SPC( 5)"(1) AUTOMATIC INFLUENCE LINE CASES"
22     PRINT  SPC( 5)"(2) FIXED-END MOMENTS AND SHEARS"
23     PRINT  SPC( 5)"(3) UNIFORM OR CONCENTRATED LOADS"
24     PRINT  SPC( 5)"(4) JOINT LOADING"
25     PRINT  SPC( 5)"(5) A COMBINATION OF THE ABOVE"
30     PRINT : PRINT : GET LO
32     HOME : VTAB (6): PRINT "THIS PROGRAM REQUIRES AN INPUT FILE TO BE
            PRESENT ON A DATA DISK IN DRIVE #2": PRINT : PRINT "DO YOU WANT
            TO SEE THE CATALOG OF THAT DISK? ": GET O$
33     IF O$ < > "Y" THEN 50
40     HOME : PRINT "THESE ARE THE FILES CURRENTLY ON YOUR DATA DISK:"
45     PRINT : PRINT D$"CATALOG,D2"
50     PRINT : PRINT : PRINT "TYPE THE NAME/NUMBER OF THE INPUT FILE":
            INPUT "TO BE USED:  INPUT FILE- ";B$
52     B$ = "INPUT FILE- " + B$
55     PRINT D$;"OPEN";B$;",D2"
60     PRINT D$;"READ";B$
65     INPUT NS
70     INPUT NA
75     NC = NS - 1:NB = NS + NC + NA:NJ = NS + NA + 2:N = 3 * NJ - 8:
            IF NA = NS THEN A$ = "N":CC = 1: GOTO 85
```

```
80    A$ = "Y":CC = 0
85    DIM SN(NB),CS(NB),JN(NB,2),V(NB,2),N(NB,2),M(NB,2),P(N),F(6),L(NB)
90    FOR I = 1 TO NB
95    INPUT L(I): INPUT SN(I): INPUT CS(I): INPUT JN(I,1): INPUT JN(I,2):
          INPUT DUM: INPUT DUM
100   NEXT I
105   PRINT D$"CLOSE"B$
110   IF LO = 4 THEN 265
135   REM ## INPUT THE MEMBER LOADING ##
140   HOME : PRINT "   ** MEMBER LOADING **": PRINT : PRINT
145   IF LO > 1 THEN 168
147   PRINT : PRINT "TYPE THE TITLE/NUMBER TO BE USED TO": PRINT "IDENTIFY
          THIS SERIES OF LOAD CASES:": INPUT "LOAD CASE- ";CI$
148   REM AUTOMATIC INFLUENCE LINE LOAD CASE GENERATION
149   FOR MN = 1 TO NS
151   FOR IL = 1 TO 4
160   REM REZERO FOR EACH LOAD CASE
168   FOR I = 1 TO NB
169   FOR J = 1 TO 2
170   V(I,J) = 0:N(I,J) = 0:M(I,J) = 0
171   NEXT J,I
172   FOR I = 1 TO N:P(I) = 0: NEXT I
173   FOR I = 1 TO 6:F(I) = 0: NEXT
174   IF LO > 1 THEN 178
175   GOSUB 2000
176   IC = IC + 1: GOTO 495
177   REM DETERMINE THE MEMBER # WHICH IS BEING LOADED
178   PRINT : INPUT "FROM LEFT TO RIGHT, WHICH BEAM IS LOADED(ENTER 0 TO
          CONTINUE TO COLUMNS)";M1
179   IF M1 = 0 THEN 200
180   IF M1 < = NS THEN 190
185   PRINT : PRINT : PRINT "YOUR STRUCTURE DOES NOT HAVE THAT MANY SPANS!":
          GOTO 178
190   MN = M1: GOSUB 2000
195   GOTO 178
200   HOME : INPUT "STARTING WITH THE LEFTMOST COLUMN, WHICH COLUMN IS
          LOADED (ENTER 0 TO CONTINUE TO THE ARCH MEMBERS)? ";M2
```

```
205   IF M2 = 0 THEN 235
210   IF M2 < = NC THEN 225
215   PRINT : PRINT "YOUR STRUCTURE DOES NOT HAVE THAT MANY COLUMNS!":
          GOTO 200
220   IF M2 = 0 THEN 235
225   MN = M2 + NS: GOSUB 2000
230   GOTO 200
235   HOME : PRINT : INPUT "FROM LEFT TO RIGHT, WHICH ARCH MEMBER IS
          LOADED (ENTER 0 WHEN LOADING IS COMPLETE";M3
240   IF M3 = 0 THEN 270
245   MN = NS + NC + M3
250   IF MN < = NB THEN 260
255   PRINT : PRINT "YOUR STRUCTURE DOES NOT HAVE THAT MANY ARCH MEMBERS!":
          GOTO 235
260   GOSUB 2000
265   REM ## INPUT THE JOINT LOADING ##
266   HOME : PRINT "     *** JOINT LOADING ***"
270   IF LO < 4 THEN 500
280   PRINT : INPUT "NUMBERING FROM LEFT TO RIGHT ON THE BEAM WHICH JOINT
          IS LOADED? (0 FOR NO JOINT LOADS ON THE BEAM)";JN
285   IF NJ = 0 THEN 500
290   K1 = 0: IF JN > 1 THEN K1 = 1
295   IF JN < = NS + 1 THEN 305
300   PRINT : PRINT "THERE AREN'T THAT MANY JOINTS ON THE BEAM!": GOTO 280
305   RN = 3 * (JN - 1) - K1
310   PRINT : INPUT "THE X-COMPONENT OF THE LOAD = ";XL
315   P(RN + 1) = P(RN + 1) + XL
320   IF JN = 1 THEN 340
325   IF JN = NS + 1 THEN 340
330   PRINT : INPUT "THE Y-COMPONENT OF THE LAOD = ";YL
335   P(RN + 2) = P(RN + 2) + YL
340   PRINT : INPUT "THE APPLIED MOMENT = ";ML
345   IF JN = 1 THEN 365
350   IF JN = NS + 1 THEN 365
355   P(RN + 3) = P(RN + 3) + ML
360   GOTO 280
```

```
365   P(RN + 2) = P(RN + 2) + ML
370   GOTO 280
375   PRINT : INPUT "FROM LEFT TO RIGHT ON THE ARCH, WHICH JOINT IS LOADED?
           (0 FOR NO ADDITIONAL LOADS) ";JN
380   IF JN = 0 THEN 500
385   IF JN < = NJ THEN 395
390   PRINT : PRINT "TOO MANY JOINTS!": GOTO 375
395   RN = 3 * (JN + NS - 1) - 2
400   IF JN = 1 THEN 415
405   IF JN = NJ THEN 415
410   GOTO 420
415   PRINT : PRINT "THIS JOINT IS FIXED AND WILL NOT AFFECT THE
           STRUCTURE!": GOTO 375
420   PRINT : INPUT "THE X-COMPONENT OF THE LOAD = ";XL
425   P(RN + 1) = P(RN + 1) + XL
430   INPUT "THE Y-COMPONENT OF THE LOAD = ";YL
435   P(RN + 2) = P(RN + 2) + YL
440   INPUT "THE APPLIED MOMENT = ";ML
445   P(RN + 3) = P(RN + 3) + ML
450   GOTO 375
495   C$ = CI$ + STR$ (IC)
496   GOTO 505
500   HOME : PRINT "THE FORCE VECTOR CORRESPONDING TO THIS LOADING
           CONDITION IS CALCULATED": PRINT : INPUT "WHAT NUMBER/TITLE
           DO YOU WANT TO ASSIGN TO THIS VECTOR? ";C$
505   C$ = "LOAD CASE- " + C$
509   ER = 0
510   PRINT D$;"OPEN"C$",D2"
515   PRINT D$;"DELETE"C$
520   PRINT D$;"OPEN"C$
525   PRINT D$;"WRITE"C$
530   PRINT B$: PRINT N: PRINT NB
535   FOR I = 1 TO N
540   PRINT P(I): NEXT I
545   FOR I = 1 TO NB
550   FOR J = 1 TO 2
```

```
555   PRINT V(I,J): PRINT N(I,J): PRINT M(I,J)
559   IF ER > 0 THEN 509
560   NEXT J,I
565   PRINT D$;"CLOSE"C$
570   IF LO > 1 THEN 600
572   PRINT "SPAN #"MN", INFLUENCE LOAD CASE #"IC
575   NEXT IL,MN
600   HOME : INPUT "DO YOU HAVE ADDITIONAL LOAD CASES BASED ON THE SAME INPUT
          FILE?";O$
605   IF O$ = "Y" THEN 135
610   PRINT
620   PRINT D$;"RUND CALC,D1"
2000  REM THIS SUBROUTINE INPUTS LOADING FOR INDIVIDUAL MEMBERS AND CONVERTS
          THE DESIRED LOADING TO EQUIVALENT JOINT LOADS ON THE STRUCTURE
          AND FIXED-END MOMENTS AND SHEARS
2003  IF LO < 3 THEN 2100
2004  REM INPUT CONCENTRATED OR UNIFORM LOADS (NOTE UNIFORM LOADING IS
          ASSUMED TO BE OVER THE ENTIRE SPAN)
2010  HOME : VTAB (6): PRINT "IS THE LOADING CONCENTRATED, UNIFORM OR A
          COMBINATION (P,W,C)? ": GET O1$
2015  IF O1$ = "P" THEN 2050
2020  IF O1$ = "W" OR O1$ = "C" THEN 2030
2025  PRINT : PRINT "WRONG RESPONSE!": FOR I = 1 TO 300: NEXT : GOTO 2010
2030  PRINT : INPUT "WHAT IS THE UNIFORM LOAD RATE(LBS/FT)? ";W
2035  V(MN,1) = + W * L(MN) / 24:V(MN,2) = + V(MN,1): REM W IS CONVERTED
          TO LBS/INCH
2040  M(MN,1) = W * (L(MN) ↑ 2) / 144:M(MN,2) = - M(MN,1)
2045  IF O1$ = "U" THEN 2085
2050  PRINT : INPUT "WHAT IS THE CONCENTRATED LOAD (LBS)?  ";P
2055  PRINT : INPUT "HOW FAR FROM THE LEFT END OF THE SPAN IS THE LOAD
          LOCATED (FT)?";X
2060  X = 12 * X
2065  FS = + P * (L(MN) + 2 * X) * ((L(MN) - X) ↑ 2) / L(MN) ↑ 3
2067  F(1) = - SN(MN) * FS:F(2) = + CS(MN) * FS:V(MN,1) = V(MN,1) + FS
2070  FS = + P * (X) ↑ 2 * (3 * L(MN) - 2 * X) / L(MN) ↑ 3
2072  F(4) = - SN(MN) * FS:F(5) = + CS(MN) * FS:V(MN,2) = V(MN,2) + FS
```

```
2075  FS = + P * X * (L(MN) - X) ↑ 2 / L(MN) ↑ 2
2077  F(3) = + FS:M(MN,1) = M(MN,1) + FS
2080  FS = - P * (L(MN) - X) * (X) ↑ 2 / L(MN) ↑ 2
2082  F(6) = + FS:M(MN,2) = M(MN,2) + FS
2085  PRINT : PRINT "DO YOU HAVE ADDITIONAL LOADS FOR THIS SPAN? (Y/N)":
          GET O2$
2088  IF O2$ = "Y" THEN 2010
2090  IF LO = 3 THEN 2185
2099  REM INPUT FIXED-END EQUIVALENTS FOR ANY GENERAL LOADING CASE
2100  IF MN = NS + NC + 1 THEN 2160
2101  IF IL = 1 THEN FS = 1; GOTO 2110
2102  IF IL = 2 THEN FS = 1: GOTO 2125
2103  IF IL = 3 THEN FS = 1: GOTO 2165
2104  IF IL = 4 THEN F(6) = 1:M(MN,2) = M(MN,2) + F(6): GOTO 2185
2105  PRINT : INPUT "FIXED-END SHEAR FOR THE LEFT END: ";FS
2110  V(MN,1) = V(MN,1) + FS
2115  F(1) = - SN(MN) * FS:F(2) = + CS(MN) * FS
2116  IF IL = 1 THEN 2185
2120  INPUT "FIXED-END MOMENT FOR THE LEFT END: ";FS
2125  F(3) = + FS
2130  M(MN,1) = M(MN,1) + FS
2131  IF IL = 2 THEN 2185
2135  IF MN = NB THEN 2190
2140  IF A$ = "Y" THEN 2150
2145  GOTO 2160
2150  IF MN = NS + 1 GOTO 2175
2155  IF MN = NS + NC THEN 2175
2160  PRINT : INPUT "FIXED-END SHEAR FOR THE RIGHT END: ";FS
2165  F(4) = - SN(MN) * FS:F(5) = + CS(MN) * FS
2170  V(MN,2) = V(MN,2) + FS
2171  IF IL = 3 THEN 2185
2175  INPUT "FIXED-END MOMENT FOR THE RIGHT END: ";F(6)
2180  M(MN,2) = M(MN,2) + F(6)
2185  REM ## CONVERT SHEARS AND MOMENTS TO GLOBAL SYSTEM LOADS ##
2190  FOR EM = 1 TO 2
2195  EN = JN(MN,EM)
```

```
2200  KS = 1:K1 = 1: IF EN > NS + 1 THEN K1 = 2
2205  IF EN = 1 THEN K1 = 0
2210  RN = 3 * (EN - 1) - K1
2215  IF EN = 1 THEN KS = 2
2220  IF EN = NS + 1 THEN KS = 2
2225  JS = 0:JJ = 3 * (EM - 1)
2230  FOR J = 1 TO 3 STEP KS
2235  JS = JS + 1
2240  P(RN + JS) = P(RN + JS) - F(JJ + J)
2245  NEXT J,EM
2250  RETURN
3000  ER = PEEK (222)
3010  IF ER = 5 THEN 3050: REM END OF DATA
3020  IF ER = 9 THEN 3080: REM DISK FULL
3030  IF ER = 13 THEN 3110: REM FILE TYPE MISMATCH
3040  HOME : PRINT "ERROR # "ER: END
3050  PRINT : PRINT : PRINT "YOU HAVE EITHER MISTYPED OR INPUT THE NAME
        OF A FILE WHICH DOES NOT EXIST."
3060  PRINT "TRY AGAIN"
3070  PRINT D$;"DELETE"B$: GOTO 40
3080  PRINT : PRINT : PRINT "THE DATA DISK IS FULL.  REPLACE IT WITH AN
        INITIALIZED DISK WITH SPACE AVAILABLE"
3090  PRINT : PRINT "HIT RETURN WHEN READY TO CONTINUE": GET OE$
3100  PRINT : RESUME
3110  PRINT : PRINT : PRINT "FILE TYPE MISMATCH": PRINT "RETYPE YOUR
        INPUT.  BE SURE TO USE THE APPROPRIATE NUMBER OR STRING."
3120  RESUME
```

Displacement Calculation Program:

```
5     REM  **11/28/83**
10    ONERR  GOTO 5000
15    REM THIS PROGRAM CALCULATES THE GLOBAL DISPLACEMENTS FOR THE
           STRUCTURE.  A K FILE AND A LOAD CASE ARE INPUT.  THE DISPLACE-
           MENTS ARE OUTPUT AND SAVED.
20    D$ = ""
25    HOME : PRINT "** GLOBAL DISPLACEMENT CALCULATION **"
26    PRINT : PRINT "THE K INVERSE FILE AND ONE OR MORE LOAD CASES ARE THE
           INPUT FOR THIS PROGRAM.": PRINT : PRINT "#NOTE# IT IS IMPORTANT
           THAT THESE TWO FILES ARE BASED ON THE SAME INPUT FILE."
27    PRINT : PRINT "THE DISPLACEMENT VECTOR CORRESPONDING TO THE LOAD
           CASE IS CALCULATED AND": PRINT "STORED ON THE DATA DISK IN
           DRIVE #2"
28    PRINT : PRINT "DO YOU WANT TO SEE THE CATALOG OF THE DATA DISK IN
           DRIVE #2?": GET O$
29    IF O$ < > "Y" THEN 40
30    PRINT : PRINT "THESE ARE THE FILES CURRENTLY ON YOUR DATA DISK:"
35    PRINT D$;"CATALOG,D2"
40    PRINT : PRINT : PRINT "TYPE THE NAME/NUMBER OF THE STIFFNESS FILE TO
           BE USED FOR THIS CASE:": INPUT "STIFFNESS FILE- ";B$
45    B$ = "STIFFNESS FILE- " + B$
50    REM READ THE STIFFNESS FILE FROM THE DATA DISC
55    PRINT D$;"OPEN"B$",D2"
60    EC = 1
65    PRINT D$;"READ"B$
70    INPUT E$: INPUT N
75    DIM K(N,N),D(N),P(N)
80    FOR I = 1 TO N
85    FOR J = 1 TO N
90    INPUT K(I,J)
95    NEXT J,I
100   PRINT D$;"CLOSE"B$
105   IF G$ = "K" THEN 265
110   C$ = B$
115   PRINT "ARE YOU CALCULATING INFLUENCE LINES? ": GET IL$
```

```
120   PRINT : IF IL$ < > "Y" THEN 180
125   HOME : VTAB (3): PRINT "THE PROCESS FOR CALCULATING THE DISPLACEMENT
          VECTORS IS AUTOMATIC.  HOWEVER, THE MEMORY SPACE IS CRITICAL
          IN THIS PHASE.  THEREFORE EACH DELTA VECTOR MUST BE STORED AS IT
          IS CALCULATED."
130   PRINT : PRINT "SELECT ONE:": PRINT : PRINT  SPC( 3)"(1) STORE ON THE
          DATA DISK CURRENTLY": PRINT  SPC( 3)"IN DRIVE 2 (MUCH
          PREFERRED!)"
135   PRINT  SPC( 3)"(2) STORE ON A SEPARATE DISC IN": PRINT SPC( 3)"DRIVE
          1": GET O
140   PRINT
145   IF O = 1 THEN DO$ = "D2"
150   IF O = 2 THEN DO$ = "D1": PRINT "PLACE A DISC WITH SPACE AVAILABLE IN
          DRIVE 1"
155   PRINT : INPUT "HOW MANY BEAM SPANS IN THIS STRUCTURE? ";NS
160   PRINT : PRINT "TYPE THE TITLE/NUMBER OF THE LOAD CASE SERIES TO BE
          USED IN THIS PROGRAM:": INPUT "LOAD CASE- ";BI$
165   PRINT : PRINT "WHAT TITLE/NUMBER DO YOU WANT TO USE TO IDENTIFY THIS
          SERIES OF DISPLACEMENT": PRINT "VECTORS?": INPUT "DISPLACEMENT
          FILE- ";GI$
170   FOR IL = 1 TO (4 * NS)
175   B$ = "LOAD CASE- " + BI$ + STR$ (IL): GOTO 190
180   PRINT : PRINT : PRINT "TYPE THE TITLE OF THE LOAD CASE USED FOR THE
          CALCULATION OF THIS DISPLACEMENT:": INPUT "LOAD CASE- ";B$
185   B$ = "LOAD CASE- " + B$
190   PRINT D$;"OPEN"B$",D2"
195   EC = 2
200   PRINT D$;"READ"B$
205   INPUT F$: INPUT N1: INPUT NB
210   REM CHECK COMPATIBILITY OF STIFFNESS FILE AND LOAD CASE
215   IF F$ < > E$ THEN 235
220   FOR I = 1 TO N: INPUT P(I): NEXT I
225   PRINT D$;"CLOSE"B$
230   GOTO 260
235   HOME : PRINT "THE K MATRIX AND LOAD CASE ARE FROM TWO DIFFERENT
          STRUCTURES!"
240   PRINT D$;"CLOSE"C$
```

```
245  PRINT : INPUT "DO YOU WANT TO SELECT A DIFFERENT K MATRIX OR LOAD
         CASE OR BOTH? (K/L/B) ";G$
250  IF G$ = "L" THEN HOME : GOTO 180
255  HOME : GOTO 40
260  REM ## SOLVE FOR DISPLACEMENTS BY MULTIPLICATION OF THE STIFFNESS
         MATRIX AND THE LOAD VECTOR##
265  HOME : PRINT "CALCULATING DISPLACEMENTS FOR": PRINT B$
270  FOR I = 1 TO N
275  D(I) = 0
280  FOR J = 1 TO N
285  D(I) = D(I) + K(I,J) * P(J)
290  NEXT J
295  NEXT I
300  REM STORE THE DISPLACEMENT VECTOR
305  IF IL$ = "Y" THEN G$ = "DISPLACEMENT FILE- " + GI$ + STR$ (IL):
         GOTO 320
310  PRINT : INPUT "WHAT NUMBER/TITLE DO YOU WANT TO ASSIGN TO THIS
         DISPLACEMENT FILE? ";G$
315  G$ = "DISPLACEMENT FILE- " + G$
320  PRINT D$;"OPEN"G$",")DO$
325  ER = 0
330  PRINT D$;"WRITE"G$
335  PRINT C$: PRINT B$: PRINT E$: PRINT N
340  FOR I = 1 TO N
345  PRINT D(I)
350  IF ER > 0 THEN 320
355  NEXT I
360  PRINT D$;"CLOSE"C$
365  IF IL$ > < "Y" THEN 375
370  NEXT IL
375  HOME : INPUT "DO YOU HAVE ADDITIONAL CASES USING THE SAME STIFFNESS
         MATRIX?";O$
380  IF O$ = "Y" THEN 110
385  PRINT : PRINT "BE SURE TO REPLACE THE PROGRAM DISC IN DRIVE 1.
         PRESS RETURN WHEN READY. ": GET O$
390  PRINT
```

```
395    PRINT D$;"RUNSTRESS,D1"
5000   ER = PEEK (222): REM USER ERROR HANDLING ROUTINE
5010   IF ER = 5 THEN 5100: REM END OF DATA
5020   IF ER = 9 THEN 5200: REM DISK FULL
5030   IF ER = 13 THEN 5300: REM FILE TYPE MIS-MATCH
5040   IF ER = 6 THEN 5400: REM FILE NOT FOUND
5080   HOME : PRINT "ERROR # "ER: END
5100   PRINT : PRINT : PRINT "YOU HAVE EITHER MISTYPED OR INPUT THE NAME
           OF A FILE WHICH DOES NOT EXIST."
5110   PRINT "TRY AGAIN"
5120   PRINT D$;"DELETE"B$: IF EC = 1 THEN 30
5130   GOTO 180
5200   PRINT : PRINT : PRINT "THE DATA DISK IN DIRVE 2 IS FULL.": PRINT
           "PLACE A DISK WITH SPACE AVAILABLE IN DRIVE 1."
5210   PRINT : PRINT "HIT RETURN WHEN READY TO CONTINUE": GET OE$
5220   PRINT : RESUME
5300   PRINT : PRINT : PRINT "FILE TYPE MISMATCH": PRINT "RETYPE YOUR INPUT.
           BE SURE TO USE THE APPROPRIATE NUMBER OR STRING."
5310   RESUME
5400   GOTO 385
```

Internal Force Calculation Program:

```
5     REM **11/28/83**
9     D$ = "":DO$ = ",D2"
10    REM THIS PROGRAM TAKES CALCULATED DEFLECTIONS AND CORRESPONDING
          INPUT FILES AND CALCULATES THE STRESS ON THE ENDS OF THE
          MEMBERS.  THE STRESS MATRIX IS STORED.
11    HOME : VTAB (6): PRINT "THIS PROGRAM USES AN INPUT FILE AND ONE OR
          MORE DEFLECTION FILES TO CALCULATE THE END-FORCES ON EACH MEMBER
          OF THE": PRINT "BRIDGE"
12    PRINT : PRINT "THE END-FORCE FILES ARE THEN STORED ON THE DATA DISK
          IN DRIVE #2"
13    PRINT : PRINT "DO YOU WISH TO SEE THE CATALOG OF THE DATA DISK IN
          DRIVE #2?": GET O$
14    IF O$ <  > "Y" THEN 30
18    REM **READ INPUT, LOADING AND DISPLACEMENT FILES FROM THE DATA DISKS**
20    HOME : PRINT "** STRESS CALCULATION **"
25    PRINT : PRINT "THESE ARE YOUR DATA FILES:": PRINT D$;"CATALOG,D2"
30    PRINT : PRINT "TYPE THE NAME/TITLE OF THE INPUT FILE:": INPUT
          "INPUT FILE- ";B$
35    B$ = "INPUT FILE- " + B$
40    PRINT D$;"OPEN"B$
45    EC = 1
50    PRINT D$;"READ"B$
55    INPUT NS: INPUT NA
60    NB = 2 * NS + NA - 1:N = (NS + NA) * 3 - 2:NL = 4 * NS
65    DIM L(NB),SN(NB),CS(NB),EI(NB),AE(NB),JN(NB,2),V(NB,2,NL),N(NB,2,NL),
          M(NB,2,NL),DL(NB,6),D(N)
70    DIM FV(NB,2),FEN(NB,2),FM(NB,2)
75    FOR I = 1 TO NB
80    INPUT L(I): INPUT SN(I): INPUT CS(I): INPUT JN(I,1): INPUT JN(I,2):
          INPUT EI(I): INPUT AE(I)
85    NEXT I
90    PRINT D$;"CLOSE"B$
95    E$ = B$
100   PRINT : PRINT "ARE YOU CALCULATING INFLUENCE LINES?": GET IL$
105   PRINT : PRINT "TYPE THE NAME/NUMBER OF THE LOAD CASE(S) TO BE USED: ":
          INPUT "LOAD CASE- ";F$
```

```
110   PRINT : PRINT "TYPE THE NAME/NUMBER OF THE DISPLACEMENT FILE(S) TO
          BE USED:": INPUT "DISPLACEMENT FILE- ";G$
115   IF IL$ > < "Y" THEN 140
120   GOSUB 1000
125   FOR IL = 1 TO NL
130   B$ = "LOAD CASE- " + F$ + STR$ (IL)
135   GOTO 145
140   B$ = "LOAD CASE- " + F$
145   PRINT D$;"OPEN "B$",D2"
150   EC = 2
155   PRINT D$;"READ"B$
160   FOR I = 1 TO (N + 3)
165   INPUT DUM$: NEXT I
170   FOR I = 1 TO NB
175   FOR J = 1 TO 2
180   INPUT FV(I,J): INPUT FEN(I,J): INPUT FM(I,J)
185   NEXT J,I
190   PRINT D$;"CLOSE"B$
195   H$ = B$:B$ = "DISPLACEMENT FILE- " + G$ + STR$ (IL)
200   IF IL$ = "Y" THEN 210
205   B$ = "DISPLACEMENT FILE- " + G$
210   PRINT D$:"OPEN"B$ + DO$
215   EC = 3
220   PRINT D$;"READ"B$
225   FOR I = 1 TO 4
230   INPUT DUM$: NEXT I
235   FOR I = 1 TO N
240   INPUT D(I): NEXT I
245   PRINT D$;"CLOSE"B$
250   I$ = B$
255   NJ = NA + 2 + NS
260   HOME : PRINT "CALCULATING THE END-FORCES FOR ": PRINT H$
265   IF NA = NS THEN A$ = "N": GOTO 275
270   A$ = "Y"
275   REM ## TRANSFORM GLOBAL DEFLECTIONS TO LOCAL ##
280   FOR MN = 1 TO NB
285   FOR EM = 1 TO 2
```

```
290   EN = JN(MN,EM): REM ##INDEX DISPLACEMENTS TO EXCLUDE SUPPORT
           CONDITIONS##
295   K1 = 1
300   IF EN = 1 THEN K1 = 0
305   IF EN > NS + 1 THEN K1 = 5
310   DN = 3 * (EN - 1) - K1
315   EK = 3 * (EM - 1) + 1
320   IF EN = 1 THEN 360: REM ##CHECK FOR BEAM SUPPORTS##
325   IF EN = NS + 1 THEN 360
330   IF EN = NS + 2 THEN 370: REM ##CHECK FOR ARCH SUPPORTS##
335   IF EN = NJ THEN 370
340   DL(MN,EK) = D(DN + 1) * CS(MN) + D(DN + 2) * SN(MN)
345   DL(MN,EK + 1) = - D(DN + 1) * SN(MN) + D(DN + 2) * CS(MN)
350   DL(MN,EK + 2) = D(DN + 3)
355   GOTO 370
360   DL(MN,EK) + D(DN + 1) * CS(MN)
365   DL(MN,EK + 2) = D(DN + 2)
370   NEXT EM
375   REM ## CALCULATE THE MOMENT AND SHEAR FOR EACH MEMBER ##
380   FOR K = 1 TO 2
385   KZ = 6:KX = 3: IF K = 1 THEN 395
390   KZ = 3:KX = 6
395   V(MN,K,IL) = + 12 * EI(MN) * (DL(MN,2) - DL(MN,5)) / L(MN) ↑ 3 +
           6 * EI(MN) * (DL(MN,3) + DL(MN,6)) / L(MN) ↑ 2
400   N(MN,K,IL) = AE(MN) * (DL(MN,4) - DL(MN,1)) / L(MN)
405   N(MN,K,IL) = ( - 1 ↑ K) * N(MN,K,IL) + FEN(MN,K)
410   M(MN,K,IL) = 6 * EI(MN) * (DL(MN,2) - DL(MN,5)) / L(MN) ↑ 2 +
           2 * EI(MN) * DL(MN,KZ) / L(MN) + 4 * EI(MN) * DL(MN,KX) / L(MN) +
           FM(MN,K)
415   V(MN,K,IL) = ( - 1 ↑ (K + 1) * V(MN,K,IL) + FV(MN,K)
420   NEXT K,MN,IL
422   PRINT : PRINT : PRINT "## STORAGE ROUTINE ##": PRINT : PRINT "PLACE
           THE DISK ON WHICH YOU WANT TO STORE THE FORCE FILE(S) IN DRIVE
           2.": PRINT "REPLACE THE PROGRAM DISK IN DRIVE 1."
425   PRINT : PRINT : PRINT "WHAT NUMBER/TITLE DO YOU WANT TO ASSIGN TO THE
           END-FORCE FILE(S)?": INPUT "STRESS FILE- ";A1$
430   IF IL$ < > "Y" THEN 440
```

```
432   FOR IL = 1 TO NL: REM ##STORE THE MEMBER END-FORCE FILES##
435   A$ = "STRESS FILE- " + A1$ + STR$ (IL)
440   A$ = "STRESS FILE- " + A$
445   PRINT D$;"OPEN"A$",D2"
450   PRINT D$;"DELETE"A$
455   PRINT D$;"OPEN"A$
460   ER = 0
465   PRINT D$;"WRITE"A$
470   PRINT E$: PRINT H$: PRINT I$: PRINT NB
475   FOR I = 1 TO NB
480   FOR J = 1 TO 2
485   PRINT V(I,J,IL): PRINT N(I,J,IL): PRINT M(I,J,IL)
490   IF ER > 0 THEN 445
495   NEXT J,I
500   PRINT D$;"CLOSE"A$
501   NEXT IL
505   PRINT : PRINT "END-FORCE CALCULATIONS ARE COMPLETE": PRINT : PRINT
          "ENSURE THAT THE PROGRAM DISK IS IN": PRINT "DRIVE 1.  PRESS
          RETURN WHEN READY.": GET O$
510   IF IL$ > < "Y" THEN 520
515   PRINT D$;"RUN INFL COEF,D1"
520   PRINT : PRINT : PRINT "DO YOU WANT TO RETURN TO THE MENU OR GO TO
          THE PRINT ROUTINE? (M/P)": GET O$
525   IF O$ = "P" THEN 535
530   PRINT D$;"RUN HELLO,D1"
535   PRINT D$;"RUN PRINTER,D1"
1000  PRINT : PRINT : PRINT "ARE THE LOAD CASE FILES AND THE": PRINT
          "DISPLACEMENT FILES ON THE SAME DATA DISKDIFFERENT DISKS OR
          A MIXTURE? (S/D/M)": GET O$
1001  REM ##ESTABLISH DISK CONTROLS FOR AUTOMATIC INFLUENCE LINE GENERATION##
1010  IF O$ = "S" THEN RETURN
1020  PRINT : PRINT : PRINT "INSERT THE DATA DISK WITH THE DISPLACEMENT
          FILES IN DRIVE 1": PRINT : PRINT "PRESS RETURN WHEN READY":
          GET O1$
1025  IF O$ = "M" THEN 1040
1030  DO$ = ",D1"
1040  RETURN
```

```
5000  ER = PEEK (222): REM ##USER ERROR HANDLING ROUTINE##
5010  IF ER = 5 THEN 5100: REM END OF DATA
5020  IF ER = 9 THEN 5200: REM DISK FULL
5030  IF ER + 13 THEN 5300: REM FILE TYPE MISMATCH
5040  IF ER = 6 THEN 5400: REM FILE NOT FOUND
5080  HOME : PRINT "ERROR # "ER: END
5100  PRINT : PRINT : PRINT "YOU HAVE EITHER MISTYPED OR INPUT THE NAME
         OF A FILE WHICH DOES NOT EXIST ON THE DATA DISK."
5110  IF EC = 1 THEN 5180
5115  IF EC = 2 THEN 5160
5120  IF IL = 1 THEN 5160
5130  IF O$ = "M" THEN DO$ = ",D1": GOTO 130
5160  PRINT "TRY AGAIN": PRINT D$;"DELETE"B$: GOTO 105
5180  PRINT "TRY AGAIN"
5190  PRINT D$;"DELETE "B$: GOTO 25
5200  PRINT : PRINT : PRINT "THE DATA DISK IS FULL.  REPLACE IT WITH AN
         INITIALIZED DISK WITH SPACE AVAILABLE"
5210  PRINT : PRINT "HIT RETURN WHEN READY TO CONTINUE": GET OE$
5220  PRINT : RESUME
5300  PRINT : PRINT : PRINT "FILE TYPE MISMATCH": PRINT "RETYPE YOUR INPUT.
         BE SURE TO USE THE APPROPRIATE NUMBER OR STRING."
5310  RESUME
5400  PRINT : PRINT : PRINT "REPLACE THE PROGRAM DISK IN DRIVE 1": PRINT :
         PRINT "HIT RETURN WHEN READY": GOTO 520
```

Influence Line Program:

```
5     REM THIS ROUTINE ASSEMBLES THE INFLUENCE COEFFICIENT VALUES AND USES
          THEM TO CALCULATE THE INFLUENCE LINES.
10    REM   **12/27/83**
20    DIM IC(101,23),L(6),IL(66)
25    D$ = ""
29    HOME : VTAB (5)
30    PRINT "THIS PROGRAM ASSEMBLES THE INFLUENCE": PRINT "COEFFICIENT
          VALUES AND USES THEM TO": PRINT "CALCULATE THE INFLUENCE LINES."
31    PRINT : PRINT "AN INPUT FILE AND A COMPLETE SET OF": PRINT "END-
          FORCE ('STRESS') FILES ARE NEEDED TORUN THIS PROGRAM."
32    PRINT : PRINT "A CLEAN INITIALIZED DISK IS NEEDED TO": PRINT "STORE
          THE INFLUENCE LINE FILES": PRINT "IT IS DESIRABLE TO HAVE A COPY
          OF THE APPROPRIATE INPUT FILE ON THIS DISK."
33    PRINT " PRINT "AT THIS TIME THE DATA DISK IN DRIVE #2 SHOULD CONTAIN
          AN INPUT FILE AND THE SETOF END-FORCE FILES."
35    PRINT : PRINT : PRINT "TYPE THE TITLE/NUMBER OF THE INPUT FILE TO BE
          USED FOR THIS PROGRAM":" INPUT "INPUT FILE- ";B$
40    PRINT : REM READ THE BEAM SPAN LENGTHS
45    REM  READ THE LENGTHS OF THE BEAM SPANS
48    B$ = "INPUT FILE- " + B$
50    PRINT D$;"OPEN"B$",D2"
55    PRINT D$;"READ"B$
60    INPUT NS,NA
65    FOR I = 1 TO NS: INPUT L(I),DUM,DUM,DUM,DUM,DUM,DUM: NEXT
70    PRINT D$;"CLOSE"B$
79    NB = 2 * NS - 1 + NA
80    NL = 4 * NS:NI = 6 * NB:EC = 2
90    REM  READ FIXED-END MOMENT AND SHEAR COEFFICIENT FILES
95    PRINT : PRINT : PRINT "TYPE THE TITLE/NUMBER OF THE END-FORCE FILES
          TO BE USED:": INPUT "STRESS FILE- ";A1$
96    HOME : VTAB (6): PRINT "RETRIEVING END-FORCE FILES"
97    FOR I = 1 TO NL
100   A$ = "STRESS FILE- " + A1$ + STR$ (I)
105   PRINT D$;"OPEN"A$",D2"
110   PRINT D$;"READ"A$
```

```
115    INPUT B$,C$,E$,NB
120    FOR J = 1 TO NI
125    INPUT IC(J - 1,I - 1)
130    NEXT J
135    PRINT D$;"CLOSE"A$
140    NEXT I
145    PRINT : PRINT "TYPE THE TITLE/NUMBER YOU WANT TO USE TO IDENTIFY THIS
           SET OF INFLUENCE LINES:": INPUT "INFLUENCE LINE- "I1$
148    HOME : VTAB (8): PRINT "A CLEAN DISC MUST BE PLACED IN DRIVE 2 TO
           RECEIVE THE INFLUENCE FILES": PRINT "PRESS RETURN WHEN READY
           TO CONTINUE": GET O$
150    FOR IL = 1 TO NI: REM  FOR EACH END FORCE OR MOMENT (6 PER MEMBER)
152    HOME : VTAB (6): PRINT "CALCULATING INFLUENCE LINE- "I1$;IL
155    IK = 0
160    PRINT : REM CLEARS GET FOR DOS
165    FOR MN = 1 TO NS: REM  LOAD EACH SPAN SEQUENTIALLY FROM LEFT TO RIGHT
170    X = 0:MC = (MN - 1) * 4
175    FOR J = 1 TO 11: REM  MOVE A UNIT LOAD 0.1L ALONG SPAN #MN
180    P = 1:X = X + .1 * L(MN):IK = IK + 1
185    IF J = 11 THEN X = L(MN) - .0001
190    IF J = 1 THEN X = .0001
195    REM  CALCULATE FIXED-END MOMENTS AND SHEARS
200    V1 =+ P *(L(MN) + 2 * X) * ((L(MN) - X) ↑ 2) / L(MN) ↑ 3
205    V2 = P * (X) ↑ 2 * (3 * L(MN) - 2 * X) / L(MN) ↑ 3
210    M1 = P * X * (L(MN) - X) ↑ 2 / L(MN) ↑ 2
215    M2 = - P * (L(MN) - X) * (X) ↑ 2 / L(MN) ↑ 2
220    REM  CALCULATE INFLUENCE LINE VALUES AS THE SUM OF FIXED-END VALUES
           TIMES THE INFLUENCE COEFFICIENT VALUES
225    IL(IK) = V1 * IC(IL - 1,MC) + M1 * IC(IL - 1,MC + 1) + V2 * IC(IL - 1,
           MC + 2) + M2 * IC(IL - 1,MC + 3)
230    NEXT J,MN
235    REM  STORE THE INFLUENCE LINES
240    IL$ = "INFLUENCE LINE- " + I1$ + STR$ (IL)
245    HOME : VTAB (10):PRINT "STORING "IL$
250    PRINT D$;"OPEN"IL$
255    PRINT D$;"DELETE"IL$
260    PRINT D$;"OPEN"IL$
```

```
265   PRINT D$;"WRITE"IL$
270   PRINT NS * 11
275   FOR I = 1 TO (NS * 11): PRINT IL(I): NEXT
280   PRINT D$;"CLOSE"IL$
285   NEXT IL
290   RUN "AASHTO,D1"
295   END
5000  ER = PEEK (222)
5010  IF ER = 5 THEN 5100: REM END OF DATA
5020  IF ER = 9 THEN 5200: REM  DISK FULL
5030  IF ER = 13 THEN 5300: REM FILE TYPE MISMATCH
5080  HOME : PRINT "ERROR # "ER: END
5100  PRINT : PRINT : PRINT "YOU HAVE EITHER MISTYPED OR INPUT THE NAME
          OF A FILE WHICH DOES NOT EXIST."
5110  PRINT "TRY AGAIN"
5115  IF EC = 2 THEN 5130
5120  PRINT D$;"DELETE"B$: GOTO 35
5130  PRINT D$;"DELETE"A$: GOTO 95
5200  PRINT : PRINT : PRINT "THE DATA DISK IS FULL.  REPLACE IT WITH AN
          INITIALIZED DISK WITH SPACE AVAILABLE"
5210  PRINT : PRINT "HIT RETURN WHEN READY TO CONTINUE": GET OE$
5220  PRINT : GOTO 245
5300  PRINT : PRINT : PRINT "FILE TYPE" MISMATCH": PRINT "RETYPE YOUR
          INPUT.  BE SURE TO USE THE APPROPRIATE NUMBER OF STRING."
5310  RESUME
```

AASHTO Rolling Load Program:

```
10    REM  ##9/8/83##
15    REM THIS ROUTINE LOADS THE STRUCTURE WITH AASHO HS20 ROLLING LOAD.
16    HOME : VTAB (6)
17    PRINT "THIS PROGRAM USES AN INPUT FILE AND A": PRINT "SERIES OF
          INFLUENCE LINES TO CALCULATE THE AASHTO HS20 ROLLING LOAD MAXIMUMS
          AND MINIMUMS."
18    PRINT : PRINT "THE DATA DISK IN DRIVE #2 SHOULD CONTAIN AN INPUT FILE
          AND THE INFLUENCE LINE": PRINT "FILES TO BE USED.": PRINT
          "REPLACE THE PROGRAM DISK WITH A CLEAN, INITIALIZED DATA DISK IN
          DRIVE #1"
25    PRINT : PRINT "TYPE THE TITLE/NUMBER OF THE INPUT FILE TO BE USED:":
          INPUT "INPUT FILE- ";A$
30    A$ = "INPUT FILE- " + A$:D$ = ""
35    PRINT D$;"OPEN"A$",D2"
40    PRINT D$;"READ"A$
45    INPUT NS,NA
46    REM DEFINE PARAMETERS
47    NB = 2 * NS + NA - 1
50    DIM L(NS),IL(NS,11),SL(NS + 1),S((NS + 1) * 11),EQ(6 * NB2)
52    SL(0) = 0
55    FOR I = 0 TO NS - 1
60    INPUT L(I),DUM,DUM,DUM,DUM,DUM,DUM
65    L(I) = L(I) / 12
66    REM ACCUMULATIVE COUNTER GIVES THE TOTAL LENGTH ALONG THE STRUCTURE
          TO LEFT END OF THE CURRENT SPAN
70    SL(I + 1) = SL(I) + L(I): NEXT
75    NB = 2 * NS + NA - 1
85    PRINT D$;"CLOSE"A$
96    SL(NS + 1) = SL(NS) + 28
97    PRINT : PRINT "TYPE THE TITLE/NUMBER OF THE INFLUENCE LINE FILES TO
          BE USED:": INPUT "INFLUENCE LINE- ";C1$
99    REM U DEFINES END-FORCES FOR EACH MEMBER
100   FOR U = 1 TO (6 * NB)
105   HOME : VTAB (6): PRINT "CALCULATING END FORCE #"U
110   C$ = "INFLUENCE LINE- " + C1$ + STR$ (U)
```

```
115   PRINT D$;"OPEN"C$",D2"
120   PRINT D$;"READ"C$
122   INPUT DUM
125   FOR I = 0 TO NS - 1
127   FOR J = 1 TO 11
130   INPUT IL(I,J): NEXT J,I
150   PRINT D$;"CLOSE"C$
155   L1 = + 8E3:L2 = + 32E3:L3 = +32E3:T = 0
159   REM ESTABLISH AN ILLUSORY SPAN ON THE FAR END OF THE STRUCTURE SO
         THAT THE VEHICLE CAN LEAVE THE STRUCTURE BY INCREMENTS
160   L(NS) = V:EQ(U,1) = 0:EQ(U,2) = 1E12
165   FOR Q = 0 TO NS
170   FOR R = 1 TO 11
171   FOR S = 14 TO 30
172   V = S + 14
174   REM DISTANCE FROM THE LEFT END OF THE STRUCTURE TO THE FRONT AXLE
175   X1 = SL(Q) + (R - 1) * L(Q) / 10
176   IF Q = NS THEN X1 = SL(Q) + R * L(Q - 1) / 10:L1 = 0
177   X2 = X1 - 14:X3 = X1 - V:T = T + 1
180   REM :FIND THE SPAN NUMBERS IN WHICH THE REAR TRACTOR AXLE AND THE
         TRAILER AXLE ARE LOCATED.
185   Q2 = 0
187   FOR RAX = 0 TO Q + 1
189   IF X2 > = SL(RAX) THEN 193
191   Q2 = RAX - 1: GOTO 195
193   NEXT RAX
195   Q3 = 0
197   FOR RAX = 0 TO Q + 1
199   IF X3 > = SL(RAX) THEN 203
201   Q3 = RAX - 1: GOTO 206
203   NEXT RAX
205   REM CHECK IF THE SECOND AXLE IS ON THE BRIDGE
206   IF Q2 < 0 THEN I2 = 0:I3 = 0: GOTO 240
208   IF Q2 = NS THEN Q2 = 0:I2 = 0: GOTO 220
209   Y2 = X2 - SL(Q2)
210   J2 = INT (Y2 / (L(Q2) / 10)) + 1
214   REM INTERPOLATE THE INFLUENCE VALUE
```

```
215   I2 = IL(Q2,J2) + (IL(Q2,J2 + 1) - IL(Q2,J2)) * (Y2 - (J2 - 1) *
          L(Q2) / 10) / (L(Q2) / 10)
219   REM CHECK IF THE THIRD AXLE IS ON THE BRIDGE
220   IF Q3 < 0 THEN I3 = 0: GOTO 240
222   IF Q3 = NS THEN I3 = 0:T = T - 1: GOTO 325
224   Y3 = X3 - SL(Q3)
225   J3 = INT (Y3 / (L(Q3) / 10)) + 1
229   REM INTERPOLATE THE INFLUENCE VALUE
230   I3 = IL(Q3,J3) + (IL(Q3,J3 + 1) - IL(Q3,J3)) * (Y3 - (J3 - 1) *
          L(Q3) / 10) / (L(Q3) / 10)
239   REM  OBTAIN THE FORCE VALUE BY MULTIPLYING THE AXLE LOADS BY THE
          CORRESPONDING INFLUENCE VALUES
240   ST = L1 * IL(Q,R) + L2 * I2 + L3 * I3
241   IF ST > EQ(U,1) THEN EQ(U,1) = ST
242   IF ST < EQ(U,2) THEN EQ(U,2) = ST
245   NEXT S,R,Q
325   NEXT U
395   HOME : VTAB (5): PRINT "THE MAXIMUM AND MINIMUM VALUES OBTAINED WILL
          BE STORED ON THE DATA DISK IN": PRINT "DRIVE #2"
400   PRINT : PRINT "TYPE THE TITLE/NUMBER THAT YOU WISH TO USE TO IDENTIFY
          THIS AASHTO FILE:": INPUT "AASHTO MAXIMUMS- "B$
410   B$ = "AASHTO MAXIMUMS- " + B$
420   PRINT D$;"OPEN"B$",D1"
425   PRINT D$;"DELETE"B$
430   PRINT D$;"OPEN"B$
435   PRINT D$;"WRITE"B$
440   PRINT 12 * NB,A$
445   FOR U = 1 TO 6 * NB
450   FOR V = 1 TO 2
455   PRINT EQ(U,V): NEXT V,U
460   PRINT D$"CLOSE"B$
475   END
```

AASHTO Equivalent Load Program:

```
10    REM     ##9/13/83##
15    REM  THIS ROUTINE LOADS THE STRUCTURE WITH AASHO HS20 EQUIVALENT
           LOADING
25    HOME : VTAB (6): PRINT "THIS PROGRAM USES AN INPUT FILE AND A": PRINT
           "SET OF INFLUENCE LINE FILES TO CALCULATE THE MAXIMUM POSITIVE
           AND NEGATIVE END-": PRINT "FORCES ON EACH MEMBER."
26    PRINT : PRINT "IN EACH CASE THE STRUCTURE IS LOADED": PRINT "WITH
           THE WORST CASE' AASHTO HS20": PRINT "EQUIVALENT LOADING."
27    PRINT : PRINT "THE DISC IN DRIVE #2 SHOULD CONTAIN THE INPUT FILE AND
           SET OF INFLUENCE LINES"
28    PRINT : PRINT "TYPE THE TITLE/NUMBER OF THE INPUT FILE TO BE USED:":
           INPUT "INPUT FILE- ";A$
30    A$ = "INPUT FILE- " + A$:D$ = ""
35    PRINT D$;"OPEN"A$",D2"
40    PRINT D$;"READ"A$
45    INPUT NS,NA
48    NB = 2 * NS + NA - 1
50    DIM L(NS),IL(NS + 1,11),EQ(6 * NB2)
52    SL(0) = 0
55    FOR I = 0 TO NS - 1
60    INPUT L(I),DUM,DUM,DUM,DUM,DUM,DUM
65    L(I) = L(I) / 12
70    NEXT I
75    NB = 2 * NS + NA - 1
85    PRINT D$;"CLOSE"A$
95    LI + 1:LI$ = "MOMENT":EI$ = "RIGHT"
96    PRINT : PRINT "TYPE THE TITLE/NUMBER OF THE SET OF": PRINT "INFLUENCE
           LINE FILES TO BE USED:": INPUT "INFLUENCE LINE- ";C1$
97    PRINT : PRINT "DO YOU WANT TO SEE THE OUTPUT BEFORE IT IS STORED?":
           GET P$
98    IF P$ < > "Y" THEN 100
99    PRINT : PRINT "PRINTER OR SCREEN? (P/S)": GET P1$
100   FOR U = 1 TO (6 * NB)
105   K = 0
110   C$ = "INFLUENCE LINE- " + C1$ + STR$ (U)
```

```
115   PRINT D$;"OPEN"C$",D2"
120   PRINT D$;"READ"C$
122   INPUT DUM
125   FOR I = 1 TO NS
127   FOR J = 1 TO 11
130   INPUT IL(I,J)
140   NEXT J,I
150   PRINT D$;"CLOSE"C$
155   YX = 0:YN = 10E12:PA = 0:NA = 0:IX = 1:IN = 1
160   FOR N = 1 TO NS
165   SP = L(N) / 120
170   FOR I = 2 TO 11
175   I1 = I - 1:I2 = I:I3 = I + 1
180   IF I < 9 GOTO 190
185   I1 = I:I2 + I - 1:I3 = I - 2
190   IF IL(N,I - 1) * IL(N,I) < 0 THEN 225
195   IF IL(N,I - 1) > 0 THEN 215
200   IF IL(N,I) > 0 THEN 215
205   NA = NA + SP * (5 * IL(N,I1) + 8 * IL(N,I2) - IL(N,I3))
210   GOTO 270
215   PA = PA + SP * (5 * IL(N,I1) + 8 * IL(N,I2) - IL(N,I3))
220   GOTO 270
225   D1 = 12 * SP * IL(N,I - 1) / (IL(N,I - 1) - IL(N,I))
230   D2 = 12 * SP - D1
235   A1 = D1 * IL(N,I - 1) / 2
240   A2 = D2 * IL(N,I) / 2
245   IF IL(N,I - 1) > 0 THEN 265
250   NA = NA + A1
255   PA = PA + A2
260   GOTO 270
265   PA = PA + A1:NA = NA + A2
270   NEXT I
285   FOR I = 1 TO 11
290   IF IL(N,I) > YX THEN 300
295   GOTO 305
300   IX = I:NX = N:YX = IL(N,I)
```

```
305   IF IL(N,I) > YN THEN 320
310   IN = I:NN = N:YN = IL(N,I)
320   NEXT I,N
323   EL = 18000
325   IF U >  < LI THEN 340
330   EL = 26000:LI = LI + 3
340   EQ(U,1) = 640 * PA + EL * YX
345   EQ(U,2) = 640 * NA + EL * YN
346   IF P$ >  < "Y" THEN 390
347   IF P1$ >  <"P" THEN 350
348   PR# 1
350   IF LI$ = "AXIAL" THEN LI$ = "MOMENT": GOTO 375
355   IF LI$ = "SHEAR" THEN LI$ = "AXIAL": GOTO 375
360   IF LI$ = "MOMENT" THEN LI$ = "SHEAR"
365   IF EI$ = "RIGHT" THEN EI$ = "LEFT": GOTO 375
370   IF EI$ = "LEFT" THEN EI$ = "RIGHT"
375   PRINT EI$" END "LI$": MEMBER #" INT ((U - 1) / 6) + 1
380   PRINT EQ(U,1) SPC ( 7)EQ(U,2)
390   NEXT U
391   PR# 0
395   HOME : VTAB (5): PRINT "ALL OF THE OUTPUT FOR THIS PROGRAM WILL BE
          STORED ON THE DATA DISK IN DRIVE #2  IN ONE FILE."
400   PRINT : PRINT "TYPE THE TITLE/NUMBER THAT YOU WISH TO USE TO IDENTIFY
          THIS AASHTO FILE:": INPUT "EQUIVALENT LOAD FILE- ";B$
410   B$ = "EQUIVALENT LOAD FILE - " + B$
420   PRINT D$;"OPEN"B$",D1"
425   PRINT D$;"DELETE"B$
430   PRINT D$;"OPEN"B$
435   PRINT D$;"WRITE"B$
440   PRINT 12 * NB,A$
445   FOR U = 1 TO 6 * NB
450   FOR V = 1 TO 2
455   PRINT EQ(U,V): NEXT V,U
460   PRINT D$"CLOSE"B$
475   PRINT D$;"RUN HELLO,D1"
```

Support Settlement Program:

```
5     REM ##8/15/83##
10    HOME : VTAB (6)
11    PRINT "THIS PROGRAM CALCULATES THE DISPLACEMENT OF EACH MEMBER
           CAUSED BY THE SETTLEMENT OR ROTATION OF ANY SUPPORT(S)"
12    PRINT : PRINT "THE K INVERSE FILE IS REQUIRED FOR INPUT AND MUST BE
           ON THE DATA DISK IN DRIVE #2"
13    PRINT : PRINT "THE RESULTING DISPLACEMENT VECTOR IS": PRINT "STORED
           ON THE DATA DISK.  THIS VECTOR IS THEN USED IN THE STRESS
           PROGRAM TO": PRINT "CALCULATE THE INDUCED END-FORCES"
15    D$ = ""
16    PRINT : PRINT "DO YOU WANT TO EXAMINE THE CONTENTS OF THE DISK IN
           DRIVE #2?": GET O$
17    IF O$ < > "Y" THEN 30
20    HOME : PRINT "THESE ARE THE FILES CURRENTLY ON YOUR DATA DISK:"
25    PRINT : PRINT D$"CATALOG,D2"
30    PRINT : PRINT : PRINT "TYPE THE TITLE/NUMBER OF THE STIFFNESS FILE
           THAT YOU WANT TO USE: ": INPUT "STIFFNESS FILE- ";C$
35    C$ = "STIFFNESS FILE- " - C$
40    PRINT D$;"OPEN"C$",D2"
45    PRINT D$;"READ"C$
50    INPUT B$
55    PRINT D$"CLOSE"C$
60    PRINT D$;"OPEN";B$
65    PRINT D$;"READ";B$
70    INPUT NS
75    INPUT NA
80    NC = NS - 1:NB = NS + NC + NA:NJ = NS + NA + 2:N = 3 * NJ - 8: IF
           NA = NS THEN A$ = "N":CC = 1: GOTO 90
85    A$ = "Y":CC = 0
90    DIM L(NB),SN(NB),CS(NB),EI(NB),AE(NB),JN(NB,2),KI(N,N),ID(N,2),
           KR(N,8),D(N)
95    FOR I = 1 TO NB
100   INPUT L(I): INPUT SN(I): INPUT CS(I): INPUT JN(I,1): INPUT JN(I,2):
           INPUT EI(I): INPUT AE(I)
105   NEXT I
```

```
110   PRINT D$"CLOSE"B$
140   REM ** CONVERT KE TO K FOR THE LEFT BEAM SUPPORT **
145   HOME : PRINT "CALCULATING THE K-FR PORTION OF THE STIFFNESS MATRIX"
150   GOSUB 3000
155   CK(1) = 1:CK(2) = N + 1:CK(3) = 2:CK(4) = 3:CK(5) = 4:CK(6) = 5
160   L = 1: GOSUB 5000
165   GOSUB 3200
170   IJ = (NS - 1) * 3
175   GOSUB 3000
180   CK(1) = IJ:CK(2) = IJ + 1:CK(3) = IJ + 2:CK(4) = IJ + 3:CK(5) =
          N + 2:CK(6) = IJ + 4
185   L = NS: GOSUB 5000
190   GOSUB 3200
195   REM  **THIS BRANCH ACCOUNTS FOR COLUMNS ATTACHED TO FIXED SUPPORTS**
200   IF A$ = "N" GOTO 260
205   L = NS + 1: GOSUB 3000
210   GOSUB 5000
215   CK(1) = 3:CK(2) = 4:CK(3) = 5
220   CK(4) = N + 3:CK(5) = N + 4:CK(6) = N + 5
225   GOSUB 3200
230   RN = (NC - 1) * 3 + 2
235   GOSUB 3000
240   L = NS + NC: GOSUB 5000
245   CK (1) = RN + 1:CK(2) = RN + 2:CK(3) = RN - 3
250   CK(4) = N + 6:CK(5) = N + 7:CK(6) = N + 8
255   GOSUB 3200
260   REM  **ACCOUNT FOR ARCHES ATTACHED TO SUPPORTS**
265   RJ = NS * 3 + 1
270   L = NS + NC + 1
275   GOSUB 3000
280   GOSUB 5000
285   CK(1) = N + 3:CK(2) = N + 4:CK(3) = N + 5
290   CK(4) = RJ + 1:CK(5) = RJ + 2:CK(6) = RJ + 3
295   GOSUB 3200
300   RJ = (NS + N4 - 1) * 3 + 1
305   L = NB
```

```
310   GOSUB 3000
315   GOSUB 5000
320   CK(1) = RJ + 1:CK(2) = RJ + 2:CK(3) = RJ + 3
325   CK(4) = N + 6:CK(5) = N + 7:CK(6) = N + 8
330   GOSUB 3200
335   PRINT D$;"OPEN"C$
340   PRINT D$;"READ"C$
345   INPUT B$,N
350   FOR I = 1 TO N
355   FOR J + 1 TO N
360   INPUT KI(I,J)
365   NEXT, J,I
370   PRINT D$;"CLOSE"C$
400   REM ## INPUT SETTLEMENT ##
405   HOME : VTAB (6): PRINT "## SETTLEMENT DISPLACEMENT INPUT ROUTINE"
410   PRINT : PRINT : PRINT "IS THERE ANY SETTLEMENT AT EITHER OF THE BEAM
          SUPPORTS? ": GET O$
415   IF O$ = "N" THEN 430
420   PRINT : INPUT "WHAT IS THE SETTLEMENT AT THE LEFT BEAM SUPPORT? (IN)";
          S(1)
425   PRINT : INPUT "WHAT IS THE SETTLEMENT AT THE RIGHT BEAM SUPPORT? (IN)";
          S(2)
430   PRINT : PRINT : PRINT "ARE THERE ANY DISPLACEMENTS AT THE ARCH
          SUPPORTS": GET O$
435   IF O$ = "N" THEN 470
440   PRINT : PRINT "DISPLACEMENTS AT THE LEFT ARCH SUPPORT;"
445   INPUT "     THE X-COMPONENT (IN): ";S(3)
447   INPUT "     THE Y-COMPONENT (IN): ";S(4)
449   INPUT "     THE ROTATION (RAD): ";S(5)
450   PRINT : PRINT : PRINT "DISPLACEMENTS AT THE RIGHT ARCH SUPPORT:"
453   INPUT "     THE X-COMPONENT (IN): ";S(6)
456   INPUT "     THE Y-COMPONENT (IN): ";S(7)
459   INPUT "     THE ROTATION (RAD): ";S(8)
470   FOR I = 1 TO N
475   KT = 0
480   FOR J = 1 TO 8
490   FOR K = 1 TO N
```

```
500   KT = KT + KI(I,K) * KR(K,J)
505   NEXT K
510   D(I) = D(I) + KT * S(J)
520   NEXT J,I
540   HOME : VTAB (4): PRINT ##S TO RAGEROUTINE##"
541   PRINT : PRINT "TYPE THE TITLE/NUMBER YOU WANT TO  USE TO IDENTIFY
          THIS DISPLACEMENT FILE:": INPUT "DISPLACEMENT FILE- ";E$
542   E$ = "DISPLACEMENT FILE- " + E$
543   F$ = "SETTLEMENT"
545   PRINT D$;"OPEN"E$",D2"
546   PRINT D$;"WRITE"E$
550   PRINT C$: PRINT F$: PRINT B$: PRINT N
555   FOR I = 1 TO N: PRINT D(I): NEXT
560   PRINT D$;"CLOSE"E$
600   HOME : PRINT "DO YOU HAVE ADDITIONAL SETTLEMENT CASES USING THE SAME
          STRUCTURE?": GET O$
610   IF O$ = "Y" THEN 400
620   PRINT : PRINT D$;"RUN STRESS,D1"
3000  REM  **ZERO CE AND CK**
3005  FOR I = 1 TO 6
3010  CK(I) = 0
3015  FOR J = 1 TO 6
3018  KT(I,J) = 0:KG(I,J) = 0
3020  NEXT J,I
3030  RETURN
3200  REM  **CONVERT KE TO K**
3210  FOR J = 1 TO 6
3220  RK = CK(J)
3225  IF RK > = N THEN RK = 0
3230  FOR K = 1 TO 6
3240  CK = CK(K) - N
3241  IF CK < = 0 THEN CK = 0
3250  KR(RK,CK) = KR(RK,CK) + KG(J,K)
3260  NEXT K,J
3270  RETURN
5000  REM  **KE-GLOBAL SUBROUTINE**
5010  REM  **INPUT KE-LOCAL**
```

```
5020  A1 = AE(L):B = 6 * EI(L) / L(L):C = 12 * EI(L) / (L(L)) ↑ 2
5040  KE(1,1) = A1:KE(1,4) = - A1:KE(4,1) = -A1:KE(4,4) = A1
5050  KE(2,2) = C:KE(2,3) = + B:KE(2,5) = - C:KE(2,6) = + B
5060  KE(3,2) = + B:KE(3,3) = 4 * EI(L):KE(3,5) = - B:KE(3,6) = 2 * EI(L)
5070  KE(5,2) = - C:KE(5,3) = - B:KE(5,5) = C:KE(5,6) = - B
5080  KE(6,2) = + B:KE(6,3) = 2 * EI(L):KE(6,5) = - B:KE(6,6) = 4 * EI(L)
5082  FOR I = 1 TO 6
5084  FOR J = 1 TO 6
5086  KE(I,J) = KE(I,J) / L(L)
5088  NEXT J,I
5100  REM  **INPUT TRANSFORMATION MATRIX (TT)**
5120  T(1,1) = CS(L):T(1,2) = - SN(L)
5130  T(2,1) = SN(L):T(2,2) = CS(L)
5140  T(3,3) = 1:T(6,6) = 1
5150  T(4,4) = CS(L):T(4,5) = - SN(L)
5160  T(5,4) = SN(L):T(5,5) = CS(L)
5200  REM  **COMPUTE KE-GLOBAL = TT*KE(L)*T **
5220  FOR J = 1 TO 6
5230  FOR K = 1 TO 6
5240  FOR I = 1 TO 6
5250  KT(J,K) = KT(J,K) + KE(J,I) * T(K,I)
5260  NEXT I,K,J
5280  FOR J = 1 TO 6
5290  FOR K = 1 TO 6
5300  FOR I = 1 TO 6
5310  KG(J,K) = KG(J,K) + T(J,I) * KT(I,K)
5320  NEXT I,K,J
5330  RETURN
```

Data Printer Utility:

```
5     REM **7/19/83**
10    REM PRINT ROUTINE FOR FILES
12    HOME : VTAB (4): PRINT "##PRINTER ROUTINE##"
15    PRINT : PRINT : PRINT SPC( 5)"(1)  INPUT FILE": PRINT  SPC( 5)"(2)
          STIFFNESS FILE": PRINT  SPC( 5)"(3)  LOAD CASE": PRINT
          SPC( 5)"(4) DISPLACEMENT FILE": PRINT  SPC( 5)"(5) STRESS FILE"
16    PRINT  SPC( 5)"(6)  INFLUENCE LINES": PRINT  SPC( 5)"(7)  AASHTO
          FILES": PRINT  SPC( 5)"(8)  EQUIVALENT LOAD FILE": PRINT
          SPC( 5)"(9)  AASHTO MAXIMUMS FILE"
17    D$ = ""
20    PRINT : PRINT "WHICH TYPE OF FILE?": VTAB (17): HTAB (21): GET O$
25    IF O$ = "1" THEN 100
26    IF O$ = "2" THEN 200
27    IF O$ = "3" THEN 300
28    IF O$ = "4" THEN 400
29    IF O$ = "5" THEN 500
30    IF O$ = "6" THEN 600
31    IF O$ = "7" THEN 700
32    IF O$ = "8" THEN 800
33    IF O$ = "9" THEN 900
35    PRINT : PRINT "TRY AGAIN!": GOTO 20
100   HOME : PRINT : PRINT : PRINT : INPUT "INPUT FILE NAME: ";A$
102   A$ = "INPUT FILE- " + A$
104   PRINT D$;"OPEN";A$;",D2
106   PRINT D$;"READ";A$
108   INPUT NS,NA
110   NB = 2 * NS + NA - 1
112   DIM L(NB),SN(NB),CS(NB),JN(NB,2),EI(NB),AE(NB)
114   FOR I = 1 TO NB
116   INPUT L(I),SN(I),CS(I),JN(I,1),JN(I,2),EI(I),AE(I)
118   NEXT I
120   PRINT D$;"CLOSE";A$
122   PR# 1
124   PRINT A$: PRINT : PRINT "NS = "NS,"NA = "NA
```

```
126  PRINT : PRINT "LENGTH","SINE","COSINE","LEFT END","RIGHT END",
         "EI","AE": PRINT
128  FOR I = 1 TO NB
130  PRINT L(I),SN(I),CS(I),JN(I,1),JN(I,2),EI(I),AE(I)
132  NEXT I
134  PR# 0
136  GOTO 1000
200  HOME : PRINT : PRINT : PRINT : INPUT "STIFFNESS FILE NAME: ";A$
203  A$ = "STIFFNESS FILE- " + A$
206  PRINT D$;"OPEN"A$;",D2"
209  PRINT D$;"READ"A$
212  INPUT B$,N
215  DIM K(N,N)
218  FOR I = 1 TO N
221  FOR J = 1 TO N
224  INPUT K(I,J): NEXT J,I
227  PRINT D$;"CLOSE"A$
230  PR# 1
233  PRINT A$: PRINT : PRINT B$,"N = "N: PRINT
236  FOR I = 1 TO N
239  PRINT "ROW #"I
242  FOR J = 1 TO N
245  PRINT K(I,J),
248  NEXT J,I
251  PR# 0
254  GOTO 1000
300  HOME : PRINT : PRINT : PRINT : INPUT "LOAD CASE NAME?";A$
302  A$ = "LOAD CASE- " + A$
304  PRINT D$"OPEN"A$",D2"
306  PRINT D$"READ"A$
308  INPUT B$,N,NB
310  DIM P(N),V(NB,2),N(NB,2),M(NB,2)
312  FOR I = 1 TO N
314  INPUT P(I): NEXT I
316  FOR I + 1 TO NB
318  FOR J = 1 TO 2
320  INPUT V(I,J),N(I,J),M(I,J)
```

```
322    NEXT J,I
324    PRINT D$;"CLOSE"A$
326    PR# 1
328    PRINT A$: PRINT : PRINT B$, "N = "N,"NB = "NB: PRINT
330    PRINT "EQUIVALENT JOINT LOADING:"
332    FOR I = 1 TO N
334    PRINT P(I),
336    NEXT I
338    PRINT : PRINT "APPLIED FEM AND FEV:"
340    PRINT "SHEAR","AXIAL","MOMENT"
342    FOR I = 1 TO NB
344    PRINT "MEMBER #"I
346    FOR J = 1 TO 2
348    PRINT V(I,J),N(I,J),M(I,J)
350    NEXT J,I
352    PR# 0
354    GOTO 1000
400    HOME : PRINT : PRINT : PRINT : PRINT : INPUT "DISPLACEMENT FILE
           NAME?";A$
401    A$ = "DISPLACEMENT FILE- " + A$
410    PRINT D$;"OPEN"A$",D2"
411    PRINT D$;"READ"A$
412    INPUT B$,C$,E$,N
415    DIM D(N)
420    FOR I = 1 TO N
425    INPUT D(I): NEXT I
430    PRINT D$;"CLOSE";A$
432    PR# 1
435    PRINT A$: PRINT : PRINT B$,C$,E$,"N = "N
440    PRINT : PRINT "JOINT #","DISPLACEMENTS (IN)"
445    FOR I = 1 TO N
450    PRINT I,D(I): NEXT I
455    PR# 0
460    GOTO 1000
500    HOME : PRINT : PRINT : PRINT : PRINT : INPUT "STRESS FILE NAME: ";A$
501    A$ = "STRESS FILE- " + A$
505    PRINT D$;"OPEN"A$",D2"
510    PRINT D$;"READ"A$
```

```
515   INPUT B$,C$,E$,NB
520   DIM V(NB,2),N(NB,2),M(NB,2)
525   FOR I = 1 TO NB
530   FOR J = 1 TO 2
535   INPUT V(I,J),N(I,J),M(I,J)
540   NEXT J,I
545   PRINT D$;"CLOSE"A$
550   PR# 1
555   PRINT A$: PRINT : PRINT B$,C$,E$,"NB = "NB
560   PRINT "SHEAR","AXIAL","MOMENT": PRINT
565   FOR I = 1 TO NB
570   PRINT "MEMBER #"I
575   FOR J = 1 TO 2
580   PRINT V(I,J),N(I,J),M(I,J)
585   NEXT J,I
590   PR# 0
595   GOTO 1000
600   D$ = ""
602   HOME : PRINT "## INFLUENCE LINES ##"
603   PRINT : INPUT "HOW MANY MEMBERS ARE THERE IN THE BRIDGEFOR WHICH YOU
         WANT THE INFLUENCE LINES? ";NB
604   T = 6 * NB: DIM AF(T)
605   PRINT : INPUT "HOW MANY INFLUENCE LINES DO YOU WANT PRINTED? ";N
606   FOR I = 1 TO N
607   PRINT : PRINT "TYPE THE FULL TITLE/NUMBER OF THE ": PRINT "INFLUENCE
         LINE FILE INCLUDING THE ": PRINT "SEQUENCE NUMBER ASSIGNED BY
         THE COMPUTER": INPUT "INFLUENCE LINE- ";A$
608   A$ = "INFLUENCE LINE- " + A$
612   PRINT D$;"OPEN"A$",D2"
614   PRINT D$;"READ"A$
622   FOR J = 1 TO T: INPUT AF(J): NEXT
624   PRINT D$;"CLOSE"A$
628   PRINT D$;"PR#1"
630   FOR J = 1 TO T: PRINT AF(J): NEXT
640   PR# 0
650   NEXT I
652   PRINT : PRINT
```

```
654  PRINT D$;"PR#0"
658  END
700  D$ = ""
703  HOME : PRINT "## INFLUENCE LINES FOR AASHTO SPECS ##"
705  PRINT : INPUT "HOW MANY LINES DO YOU WANT TO PRINT?   ";N
706  FOR I = 1 TO N
709  PRINT : PRINT "TYPE THE FULL TITLE/NUMBER OF THE AASHTOFILE INCLUDING
       THE SEQUENCE NUMBER WHICH WAS ASSIGNED BY THE COMPUTER:": INPUT
       "AASHTO FILE- ";A$
712  A$ = "AASHO FILE- " + A$
718  PRINT D$;"OPEN"A$",D2"
721  PRINT D$;"READ"A$
724  INPUT T
727  IF U > 1 THEN 733
730  DIM AF(5,T)
733  FOR J = 1 TO T: INPUT AF(J): NEXT
736  PRINT D$;"CLOSE"A$
742  PR# 1
745  FOR J = 1 TO T
748  IF U = 103 THEN 757
751  FOR J = 1 TO 5
754  GOTO 760
757  PRINT AF(1,I),AF(2,I): GOTO 775
772  NEXT J
773  PR# 0
775  NEXT I
787  END
800  D$ = ""
804  INPUT "WHAT FILE NAME/NUMBER DO YOU WANT PRINTED? ";A$
808  A$ = "EQUIVALENT LOAD FILE - " + A$
812  PRINT D$;"OPEN"A$",D2"
816  PRINT D$;"READ"A$
820  INPUT A,B$
824  DIM EL(A)
828  FOR I + 1 TO A: INPUT EL(I): NEXT
832  PRINT D$;"CLOSE"A$
836  PR# 1
```

```
840   C = 1
844   FOR I = 1 TO (A / 12)
848   PRINT "MEMBER #"I
852   FOR J = 1 TO 6
856   PRINT EL(C); SPC( 7);EL(C + 1)
860   C = C + 2
864   NEXT J,I
868   HOME : PRINT "MORE CASES?": GET O$
872   IF O$ = "Y" THEN 1010
876   END
900   D$ = ""
904   INPUT "WHAT FILE NAME/NUMBER DO YOU WANT PRINTED? ";A$
908   A$ = "AASHTO MAXIMUMS- " + A$
912   PRINT D$;"OPEN"A$",D2"
916   PRINT D$;"READ"A$
920   INPUT A,B$
924   DIM EL(A)
928   FOR I = 1 TO A: INPUT EL(I): NEXT
932   PRINT D$;"CLOSE"A$
936   PR# 1
940   C = 1
944   FOR I = 1 TO (A / 12)
948   PRINT "MEMBER #"I
952   FOR J = 1 TO 6
956   PRINT EL(C); SPC( 7);EL(C + 1)
960   C = C + 2
964   NEXT J,I
968   HOME : PRINT "MORE CASES?": GET O$
972   IF O$ = "Y" THEN 1000
976   END
1000  HOME : PRINT : PRINT : PRINT : PRINT "MORE?": GET O2$
1004  D$ = ""
1005  IF O2$ = "N" THEN 1020
1006  PRINT
1010  PRINT D$;"RUNPRINTER,D1"
1020  END
```

File Transfer Utility:

```
5     REM **7/19/83**
10    HOME : VTAB (6): PRINT "## FILE TRANSFER ROUTINE ##"
15    PRINT : PRINT  SPC( 5)"(1)  INPUT FILE": PRINT  SPC( 5)"(2) STIFFNESS
          FILE": PRINT  SPC( 5)"(3) CATALOG OF DRIVE #2"
17    D$ = ""
20    PRINT : PRINT "WHICH TYPE OF FILE?": GET O$
25    IF O$ = "1" THEN 50
26    IF O$ = "2" THEN 150
27    IF O$ = "3" THEN 250
35    PRINT : PRINT "TRY AGAIN!": GOTO 20
50    HOME : PRINT : PRINT : PRINT : INPUT "INPUT FILE NAME: ";A$
55    A$ = "INPUT FILE- " + A$
60    PRINT D$;"OPEN";A$;",D2"
61    PRINT D$;"READ";A$
65    INPUT NS,NA
70    NB = 2 * NS + NA - 1
75    DIM L(NB),SN(NB),CS(NB),JN(NB,2),EI(NB),AE(NB)
80    FOR I = 1 TO NB
85    INPUT L(I),SN(I),CS(I),JN(I,1),JN(I,2),EI(I),AE(I)
90    NEXT I
95    PRINT D$;"CLOSE";A$
100   HOME : PRINT "INSERT DISC TO RECEIVE "A$" IN DRIVE 2.  PRESS RETURN":
          GET O$
102   PRINT
103   D$ = ""
105   PRINT D$;"OPEN"A$
110   PRINT D$;"WRITE"A$
115   PRINT NS: PRINT NA
120   FOR I = 1 TO NB
125   PRINT L(I): PRINT SN(I): PRINT CS(I): PRINT JN(I,1): PRINT JN(I,2):
          PRINT EI(I): PRINT AE(I)
130   NEXT I
135   PRINT D$;"CLOSE"A$
140   GOTO 1000
150   HOME : PRINT : PRINT : PRINT : INPUT STIFFNESS FILE NAME: ";A$
```

```
151   A$ = "STIFFNESS FILE- " + A$
155   PRINT D$;"OPEN"A$;",D2"
156   PRINT D$;"READ"A$
160   INPUT B$,N
165   DIM K(N,N)
170   FOR I = 1 TO N
175   FOR J = 1 TO N
180   INPUT K(I,J): NEXT J,I
185   PRINT D$;"CLOSE"A$
190   HOME : PRINT "INSERT NEW DISC INTO DRIVE 2": PRINT "PRESS RETURN":
          GET O$
191   PRINT :D$ = ""
195   PRINT D$:"OPEN"A$
200   PRINT D$;"WRITE"A$
205   PRINT B$: PRINT N
207   FOR I = 1 TO N
208   FOR J = 1 TO N
210   PRINT K(I,J): NEXT J,I
215   PRINT D$;"CLOSE"A$
220   GOTO 1000
250   D$ = ""
260   HOME : VTAB (7): PRINT "PRESS RETURN WHEN YOU ARE READY": GET O1$
270   PRINT
280   PRINT D$:"CATALOG,D2"
290   PRINT : PRINT "PRESS RETURN WHEN YOU ARE READY TO RETURN TO THE
          MENU": GET O1$
300   GOTO 10
1000  HOME : PRINT : PRINT : PRINT : PRINT "MORE?": GET O2$
1004  D$ = ""
1005  IF O2$ = "N" THEN 1020
1006  PRINT
1010  PRINT D$;"RUNTRANSFER,D1"
1020  END
```

TABLE B-1

PROPERTIES INPUT FILE
for Test Structure 6S

Member Number	Length (ft)	EI $lb\text{-}in^2 \times 10^9$	AE $lb \times 10^9$	SINE	COSINE	JOINT NUMBERS Left	Right
(Beams)							
1	60.0	351	1.659	0.0	1.0	1	2
2	35.0	351	1.659	0.0	1.0	2	3
3	35.0	351	1.659	0.0	1.0	3	4
4	35.0	351	1.659	0.0	1.0	4	5
5	35.0	351	1.659	0.0	1.0	5	6
6	60.0	351	1.659	0.0	1.0	6	7
(Columns)							
7	35.0	32.2	.844	-1.0	0.0	2	9
8	20.0	32.2	.844	-1.0	0.0	3	10
9	16.0	32.2	.844	-1.0	0.0	4	11
10	20.0	32.2	.844	-1.0	0.0	5	12
11	35.0	32.2	.844	-1.0	0.0	6	13
(Arch)							
12	47.2	98.6	2.192	.530	.848	8	9
13	38.1	98.6	2.192	.394	.919	9	10
14	35.2	98.6	2.192	.114	.994	10	11
15	35.2	98.6	2.192	- .144	.994	11	12
16	38.1	98.6	2.192	- .394	.919	12	13
17	47.2	98.6	2.192	- .530	.848	13	14

TABLE B-2

COMPARISON OF MAXIMUM INTERNAL FORCES DUE TO AASHTO ROLLING LOADS OR EQUIVALENT LOADING

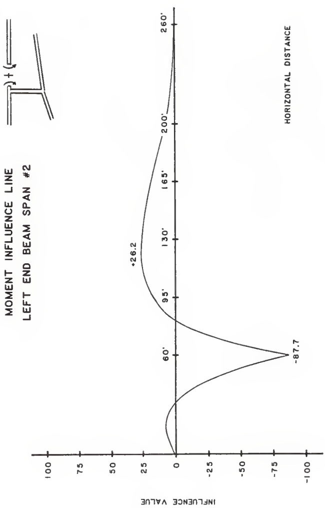
*	LEFT END						RIGHT END					
	Shear (Kips)		Axial (Kips)		Moment (K-ft)		Shear (Kips)		Axial (Kips)		Moment (K-ft)	
	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.
1-R	59.25	- 3.67	0.0	0.0	0.0	0.0	46.87	- 7.35	0.0	0.0	441.0	-220.5
1-E	39.30	- 4.62	0.0	0.0	0.0	0.0	35.90	- 5.45	0.0	0.0	293.4	-255.8
2-R	48.52	- 8.74	9.74	- 7.76	152.5	-325.7	34.17	-14.46	7.76	- 9.74	427.4	-378.0
2-E	36.56	-11.38	7.81	- 7.53	173.6	-203.5	34.98	-10.63	7.53	- 7.81	305.3	-379.3
3-R	43.01	-14.56	15.38	-14.01	344.5	-410.4	35.30	-17.62	14.01	- 15.38	351.3	-232.2
3-E	37.83	-14.26	11.33	-12.09	372.6	-303.8	35.79	-16.15	12.10	- 11.33	242.3	-226.2
4-R	42.58	-14.25	15.35	-14.03	232.3	-356.2	35.73	-17.66	14.03	- 15.35	419.9	-344.1
4-E	37.25	-14.96	12.39	-10.96	244.2	-242.3	36.64	-15.72	10.96	- 12.39	350.2	-344.3
5-R	41.16	- 9.02	9.67	- 7.74	376.9	-419.8	42.96	- 8.71	7.74	- 9.67	294.3	-152.1
5-E	40.98	-10.56	8.86	- 7.15	353.2	-359.5	45.32	-10.21	7.15	- 8.86	204.7	-161.9
6-R	53.08	- 4.31	0.0	0.0	220.5	-420.3	54.12	- 3.67	0.0	0.0	0.0	0.0
6-E	25.02	- 6.49	0.0	0.0	235.6	-316.3	28.53	- 4.34	0.0	0.0	0.0	0.0
7-R	7.76	- 9.74	45.88	- 5.65	127.9	-164.5	9.74	- 7.76	5.65	- 45.88	143.7	-176.7
7-E	8.41	- 8.96	36.55	- 4.52	123.6	-126.5	8.96	- 8.41	4.53	- 36.55	139.9	-147.1
8-R	10.90	-12.50	28.58	- 5.21	104.5	-137.2	12.50	-10.90	5.21	- 28.58	114.3	-114.1
8-E	10.75	-10.51	35.15	- 2.65	96.7	- 97.0	10.51	-10.75	2.64	- 35.15	93.2	- 83.3
9-R	18.99	-19.14	26.40	0.0	165.1	-166.5	19.14	-18.99	0.0	- 26.40	139.0	-140.0
9-E	18.27	-20.45	28.43	0.0	141.2	-159.2	20.45	-18.27	0.0	- 28.43	115.6	-132.6

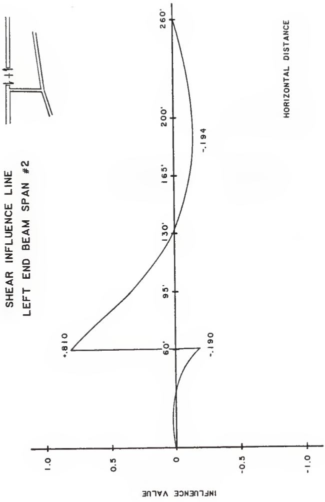
*Where the number represents the member number and the letter indicates rolling load (R) or equivalent loading (E).

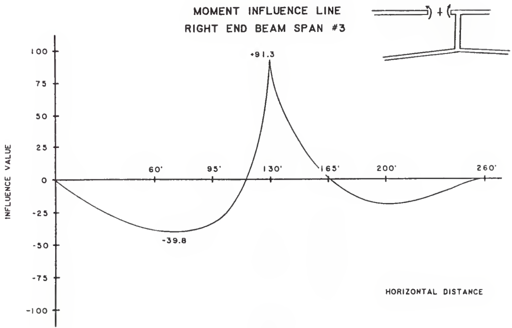
TABLE B-2 (continued)

*	LEFT END						RIGHT END					
	Shear (Kips)		Axial (Kips)		Moment (K-ft)		Shear (Kips)		Axial (Kips)		Moment (K-ft)	
	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.
10-R	12.40	-10.93	28.39	- 5.29	135.5	-105.0	10.93	-12.39	5.29	- 28.39	113.1	-114.6
10-E	10.60	-10.09	37.44	- 1.50	98.0	- 92.2	10.09	-10.60	1.50	- 37.44	84.1	- 84.4
11-R	9.67	- 7.74	45.90	- 5.65	163.4	-127.6	7.74	- 9.67	5.65	- 45.90	176.7	-143.3
11-E	10.01	- 8.03	34.49	- 4.46	150.2	-117.4	8.03	-10.01	4.46	- 34.49	160.2	-132.8
12-R	20.64	-11.02	81.11	0.0	509.1	-292.8	11.02	-20.64	0.0	- 81.11	464.4	-227.4
12-E	17.75	-12.76	130.48	0.0	387.1	-300.9	12.76	-17.75	0.0	-130.48	337.3	-242.2
13-R	5.27	-11.32	79.83	0.0	93.9	-326.8	11.32	- 5.27	0.0	- 79.83	166.8	-109.5
13-E	4.37	- 8.37	120.55	0.0	114.7	-211.6	8.37	- 4.37	0.0	-120.55	113.7	-129.4
14-R	7.48	- 7.14	70.11	0.0	112.6	-139.7	7.14	- 7.48	0.0	- 70.11	157.5	-120.3
14-E	6.90	- 6.02	112.51	0.0	107.8	- 95.5	6.02	- 6.90	0.0	-112.51	111.8	- 94.0
15-R	7.26	- 7.38	70.20	0.0	121.2	-156.1	7.38	- 7.26	0.0	- 70.20	141.0	-111.3
15-E	7.14	- 6.48	110.01	0.0	108.5	-109.2	6.48	- 7.14	0.0	-110.14	121.1	- 96.3
16-R	11.35	- 5.16	79.83	0.0	110.8	-166.4	5.16	-11.35	0.0	- 79.83	326.8	- 94.0
16-E	6.52	- 4.59	119.88	0.0	101.0	-131.8	4.59	- 6.52	0.0	-119.88	171.6	-106.7
17-R	11.00	-20.65	81.14	0.0	226.9	-464.9	20.65	-11.00	0.0	- 81.14	291.9	-508.9
17-E	12.06	-16.75	130.29	0.0	227.2	310.3	16.75	-12.06	0.0	-130.29	283.2	-366.8

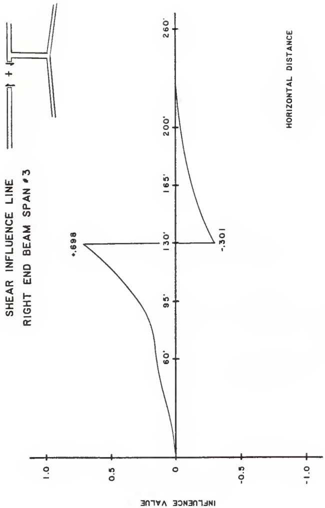
*Where the number represents the member number and the letter indicates rolling load (R) or equivalent loading (E).


MOMENT INFLUENCE LINE
LEFT END BEAM SPAN #2
+26.2
-87.7
60'
95'
130'
165'
200'
260'
HORIZONTAL DISTANCE
100
75
50
25
0
-25
-50
-75
-100
INFLUENCE VALUE

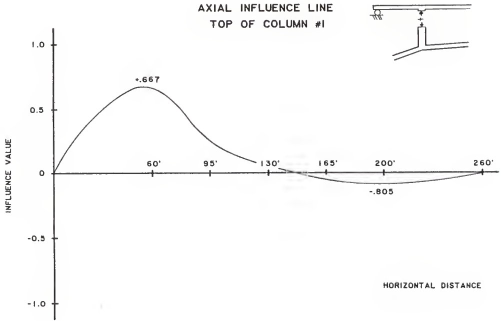

SHEAR INFLUENCE LINE
LEFT END BEAM SPAN #2
+.810
-.190
-.194
60'
95'
130'
165'
200'
260'
1.0
0.5
0
-0.5
-1.0
INFLUENCE VALUE
HORIZONTAL DISTANCE

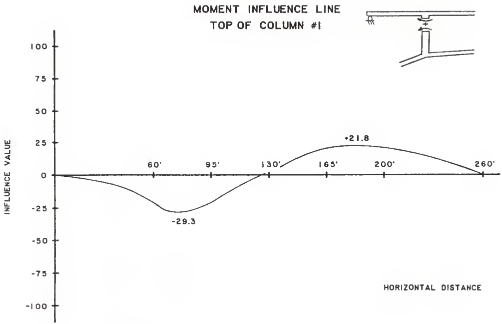


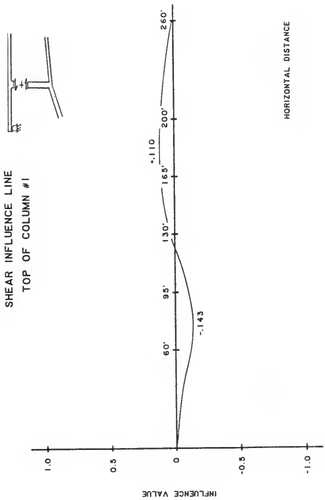
MOMENT INFLUENCE LINE
RIGHT END BEAM SPAN #3
+91.3
-39.8
100
75
50
25
0
-25
-50
-75
-100
INFLUENCE VALUE
60'
95'
130'
165'
200'
260'
HORIZONTAL DISTANCE



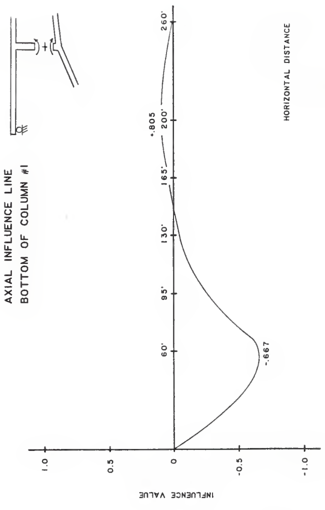
SHEAR INFLUENCE LINE
RIGHT END BEAM SPAN #3
+.698
-.301
60'
95'
130'
165'
200'
260'
HORIZONTAL DISTANCE
1.0
0.5
0
-0.5
-1.0
INFLUENCE VALUE

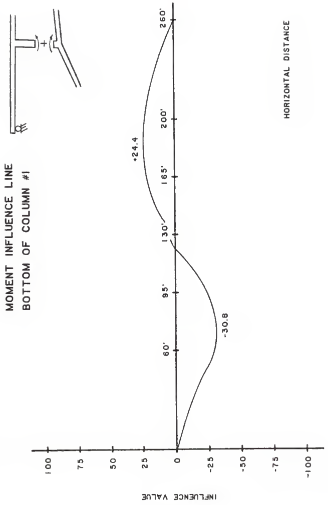

AXIAL INFLUENCE LINE
TOP OF COLUMN #1
+.667
-.805
1.0
0.5
0
-0.5
-1.0
INFLUENCE VALUE
60'
95'
130'
165'
200'
260'
HORIZONTAL DISTANCE

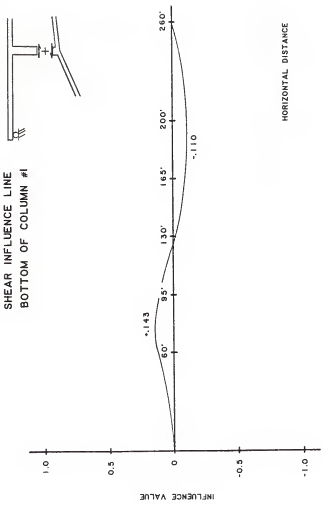

MOMENT INFLUENCE LINE
TOP OF COLUMN #1
INFLUENCE VALUE
100
75
50
25
0
-25
-50
-75
-100
60'
95'
130'
165'
200'
260'
+21.8
-29.3
HORIZONTAL DISTANCE

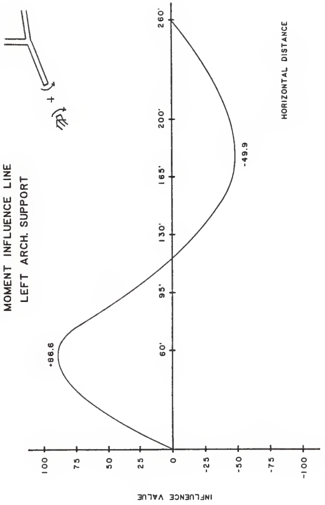


SHEAR INFLUENCE LINE
TOP OF COLUMN #1
INFLUENCE VALUE
1.0
0.5
0
-0.5
-1.0
60'
95'
130'
165'
200'
260'
-.143
+.110
HORIZONTAL DISTANCE

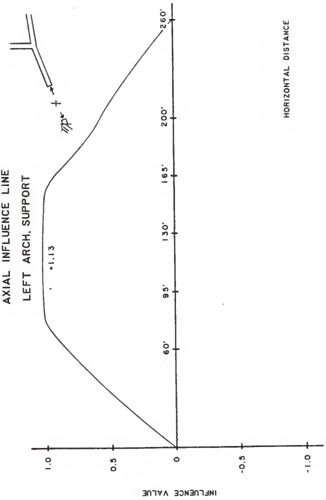

AXIAL INFLUENCE LINE
BOTTOM OF COLUMN #1
HORIZONTAL DISTANCE
INFLUENCE VALUE
1.0
0.5
0
-0.5
-1.0
60'
95'
130'
165'
200'
260'
-.667
+.805

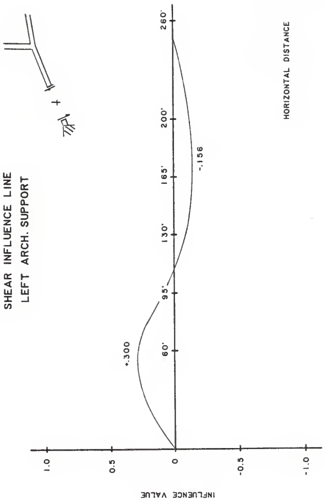

MOMENT INFLUENCE LINE
BOTTOM OF COLUMN #1
+24.4
-30.8
60'
95'
130'
165'
200'
260'
HORIZONTAL DISTANCE
100
75
50
25
0
-25
-50
-75
-100
INFLUENCE VALUE

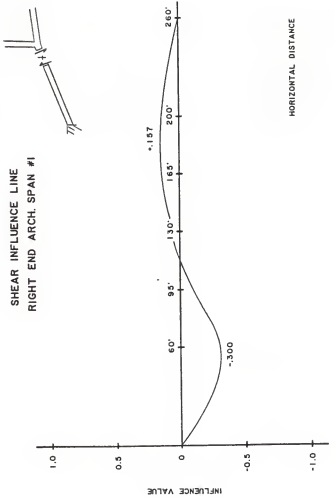

SHEAR INFLUENCE LINE
BOTTOM OF COLUMN #1
-.143
-.110
60'
95'
130'
165'
200'
260'
HORIZONTAL DISTANCE
1.0
0.5
0
-0.5
-1.0
INFLUENCE VALUE

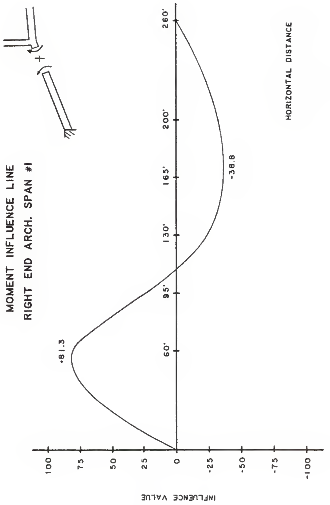


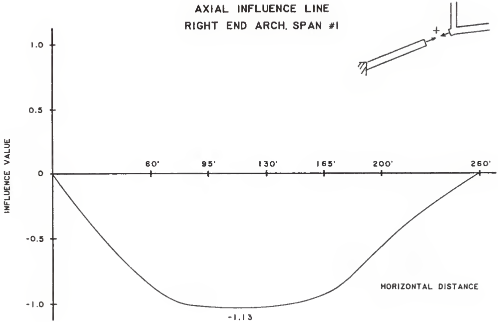
MOMENT INFLUENCE LINE
LEFT ARCH. SUPPORT
+
+86.6
-49.9
60'
95'
130'
165'
200'
260'
HORIZONTAL DISTANCE
100
75
50
25
0
-25
-50
-75
-100
INFLUENCE VALUE

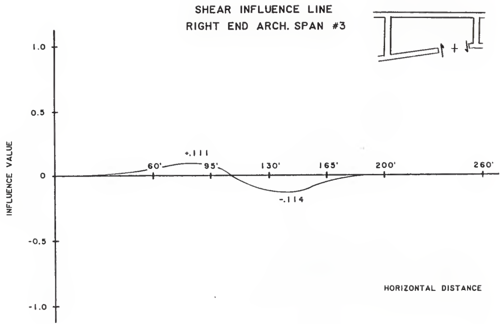

AXIAL INFLUENCE LINE
LEFT ARCH. SUPPORT
1.13
+
1.0
0.5
0
-0.5
-1.0
INFLUENCE VALUE
60'
95'
130'
165'
200'
260'
HORIZONTAL DISTANCE

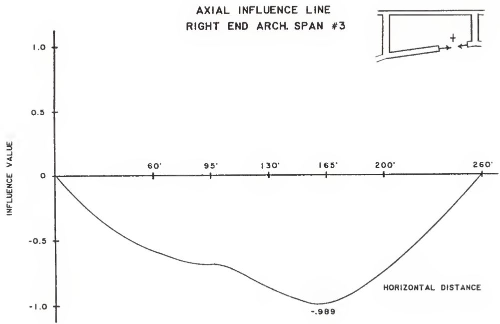

SHEAR INFLUENCE LINE
LEFT ARCH. SUPPORT
+.300
-.156
1.0
0.5
0
-0.5
-1.0
INFLUENCE VALUE
60'
95'
130'
165'
200'
260'
HORIZONTAL DISTANCE

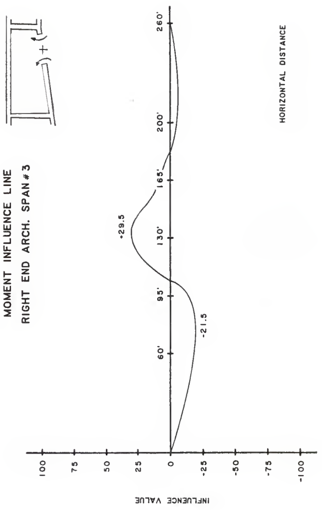

SHEAR INFLUENCE LINE
RIGHT END ARCH. SPAN #1
HORIZONTAL DISTANCE
INFLUENCE VALUE
1.0
0.5
0
-0.5
-1.0
60'
95'
130'
165'
200'
260'
-.300
+.157


MOMENT INFLUENCE LINE
RIGHT END ARCH. SPAN #1
+81.3
-38.8
60'
95'
130'
165'
200'
260'
HORIZONTAL DISTANCE
INFLUENCE VALUE
100
75
50
25
0
-25
-50
-75
-100
+


AXIAL INFLUENCE LINE
RIGHT END ARCH, SPAN #1
+
1.0
0.5
0
-0.5
-1.0
INFLUENCE VALUE
60'
95'
130'
165'
200'
260'
-1.13
HORIZONTAL DISTANCE


SHEAR INFLUENCE LINE
RIGHT END ARCH. SPAN #3
1.0
0.5
0
-0.5
-1.0
INFLUENCE VALUE
+.111
-.114
60'
95'
130'
165'
200'
260'
HORIZONTAL DISTANCE

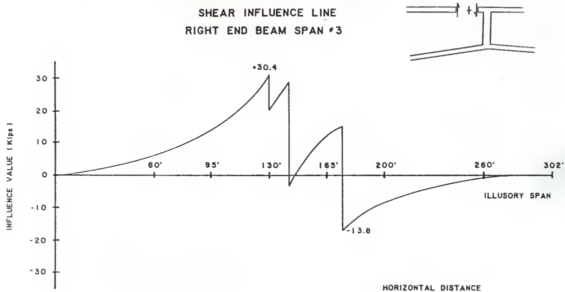

AXIAL INFLUENCE LINE
RIGHT END ARCH. SPAN #3
INFLUENCE VALUE
1.0
0.5
0
-0.5
-1.0
60'
95'
130'
165'
200'
260'
HORIZONTAL DISTANCE
-.989
+

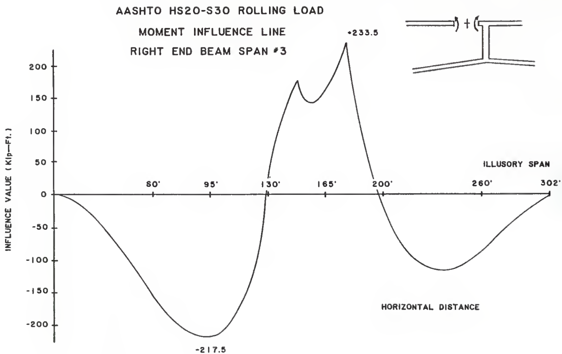

MOMENT INFLUENCE LINE
RIGHT END ARCH. SPAN # 3
+29.5
-21.5
60'
95'
130'
165'
200'
260'
HORIZONTAL DISTANCE
100
75
50
25
0
-25
-50
-75
-100
INFLUENCE VALUE

AASHTO HS20-S30 ROLLING LOAD

SHEAR INFLUENCE LINE

RIGHT END BEAM SPAN #3




AASHTO HS20-S30 ROLLING LOAD
MOMENT INFLUENCE LINE
RIGHT END BEAM SPAN #3
+233.5
-217.5
INFLUENCE VALUE (Kip—Ft.)
200
150
100
50
0
-50
-100
-150
-200
80'
95'
130'
165'
200'
260'
302'
ILLUSORY SPAN
HORIZONTAL DISTANCE

BIBLIOGRAPHY

Agnew, Jeanne and Knapp, Robert C. Linear Algebra With Applications. Monterey, CA: Brooks/Cole Publishing Company, 1978.

Apple II: the DOS Manual. Cupertino, CA: Apple Computer, Inc., 1981.

Apple II: Reference Manual. Cupertino, CA: Apple Computer, Inc., 1981.

Bosseman, Bayard E. and Ford, Maurice E. "Development of Computerized Specifications," presented at the ASCE Houston Convention, October 21, 1983.

Blaszkowiak, S. and Kaczkowski, Z. Iterative Methods of Structural Analysis. London: Permagon Press, 1966.

Cook, Steven, McNiff, Martin and Pode, Lon. Apple II Users' Guide. Berkeley, CA: Osborne/McGraw-Hill, 1981.

Fisher, James M. and West, Michael A. "Productivity Changes Using Micro/Mini Computer Systems in Consulting Engineering," presented at the ASCE Houston Convention, October 19, 1983.

Gallagher, Richard H. and McGuire, William. Matrix Structural Analysis. New York: John Wiley & Sons, 1979.

Gutkowski, Richard M. Structures: Fundamental Theory and Behavior. New York: Van Nostrand Reinhold Co., 1981.

Marshall, W. T. Solution of Problems in Structures. London: Pitman Publishing Limited, 1978.

Martin, Harold C. Introduction to Matrix Methods of Structural Analysis. New York: McGraw-Hill Book Company, 1966.

McCormac, Jack C. Structural Steel Design. Scranton, PA: International Textbook Company, 1969.

Nilson, Arthur H. and Winter, George. Design of Concrete Structures, 9th ed. New York: McGraw-Hill Book Company, 1979.

Norris, John Benson and Wilbur, Charles H. Elementary Structural Analysis, 2nd ed. New York: McGraw-Hill Book Company, 1960.

Wang, Chu-Kia. Matrix Methods of Structural Analysis, Scranton, PA: International Textbook Co., 1966.

Wright, Whitman. "The Use of Conversational Computer Programs in the Structural Engineering Office," Canadian Journal of Civil Engineering, 4, 417 (1977) 417-435.

DEVELOPMENT OF MICRO-COMPUTER
PROGRAMS FOR THE ANALYSIS OF
AN OPEN SPANDREL ARCH

by

JOHN JAY KOONTZ

B.S., United States Military Academy, 1971

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Civil Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1984

ABSTRACT

The objective of this research effort is to produce a computer analysis of an open-spandrel arch which is "friendly" to the user. The displacement method of analysis is used to analyze the structure. Matrix methods are used to simplify the calculations. The analysis includes, but is not limited to, the analysis of any given loading, the generation of influence lines for deck loading, automatic AASHTO code live load analysis, and the analysis of support yielding. Originally intended to be one single program, the final product is a package of several single purpose programs which is necessitated by the memory and data storage limitations of the micro-computer. Each program stores its output on a data disk and uses previously stored files for input. Automation is maximized to reduce the user-computer interaction to the minimum. A listing of the code for each of the programs and a set of sample output are included as appendices.