MAC-TR-6

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
PROJECT MAC

STRESS: A PROBLEM-ORIENTED LANGUAGE FOR STRUCTURAL ENGINEERING

by John M. Biggs
Robert D. Logcher

Civil Engineering Department

May 6, 1964

ABSTRACT:

STRESS is a general purpose programming system for the analysis of structures. Compared to most other [illegible] it has three distinguishing characteristics: (1) the input language is that of the structural engineer which [illegible] communication between the engineer and the computer, (2) the system is capable of analyzing a wide variety of structural types and loading conditions, thus permitting industrial use on a routine basis, and (3) [illegible] is exemplified by the fact that modifications of the [illegible] can be easily executed. The [illegible] is used in the time-sharing mode. [illegible] to provide a system which not only reduces the [illegible] for structural analysis but, more significantly, enhances the [illegible] ability to design an efficient structure.

INTRODUCTION

Presented in this paper is a brief description of STRESS (Structural Engineering Systems Solver) which is a system for structural analysis by digital computer. It consists of a language which describes the structural problem and a processor which produces the requested results. STRESS is a general purpose system in the sense that it is capable of analyzing a wide variety of structural types and situations. The input lanugage is problem-oriented, i.e., the only problem description required is in engineering rather than computer language.

On the assumption that the reader is not a structural engineer, a few words concerning the general nature of the structural design problem appear to be in order. For example, consider the simple building frame shown in Fig. 1. The members of this structural system may be of steel, reinforced concrete or some other material and are rigidly connected at the joints. The objective of design is to evolve a structure which will support the imposed loads without excessive stress or deformation and with maximum economy.

The analysis of the relatively simple frame in Fig. 1 requires the determination of 63 distinct force and moment components. This is accomplished by the solution of an equal number of equations. Forty-two of these are classified as equilibrium equations. The remainder express the compatability of distortions between the various elements. The total set of equations may be subdivided such that analysis requires the solution of 21 simultaneous equations. It should be apparent that rigorous analysis of a more sizable structure (e.g., a 20-story building frame) requires an enormous amount of computation and data processing.

The problem is further complicated by the fact that the deformation of the individual members and hence the compatability equations depend upon the size and elastic properties of those members. Hence design must be an iterative process each cycle of which involves a new analysis of the complete structure and a revision of the member sizes.

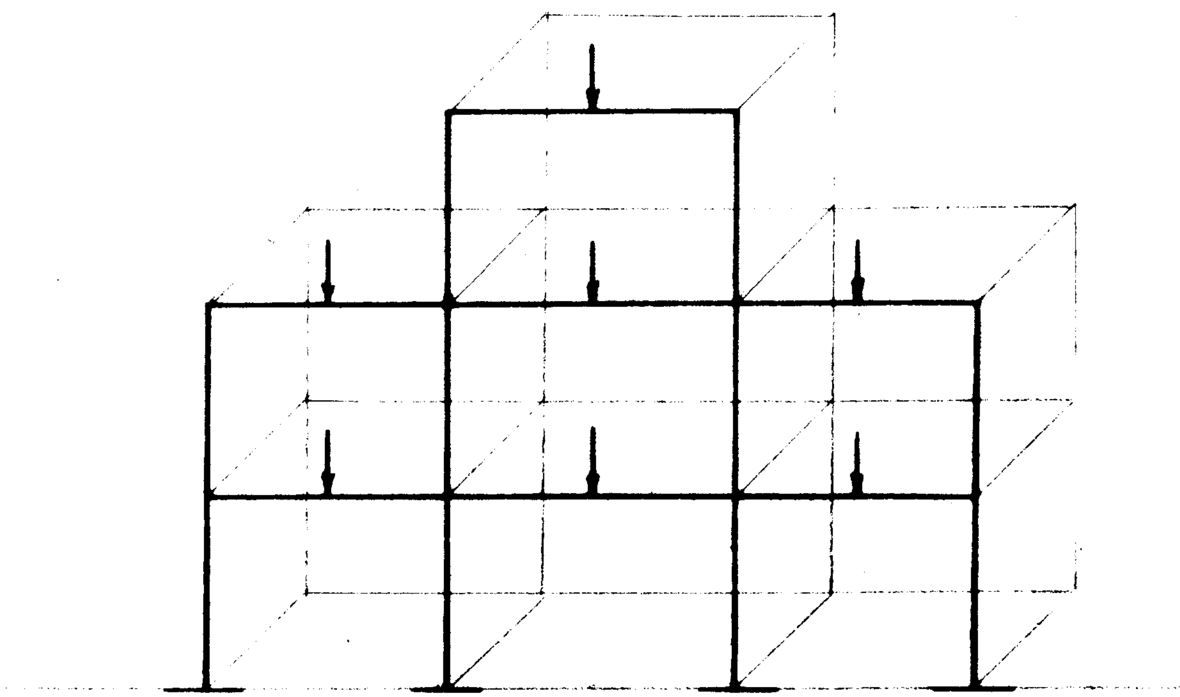


Fig. 1   Rigid Building Frame

Before the advent of computers rigorous or "exact" analysis was possible only in the most simple cases. For structures of moderate size relaxation techniques were commonly used. In the case of larger structures it was necessary to employ approximate methods. These latter methods were very ingenious and resulted in structures which were satisfactory from the viewpoint of safety and performance. However, because of the approximation in analysis a degree of conservatism was prudent and structures were seldom optimal with respect to economy. Furthermore, the very time-consuming computation involved, even with the approximate methods, made it impractical to conduct the repetitive analysis necessary to converge on an optimum design.

Because of the nature of the problem structural analysis is ideally suited to electronic computation. Structural engineers were quick to recognize this fact and extensive use has been made of computers over the past decade. This usage was primarily in the form of special purpose programs, i.e. programs written for the analysis of a particular structure. Since the time and money required to develop a new program is considerable this form of computer application is restricted to large, important structures and is not a satisfactory solution to the general structural engineering problem.

More general programs have been written but these are also restrictive in that they may be used only for a specific type of structure, e.g., continuous stringer bridges. Although a step in the right direction, this attack falls far short of the ultimate objective of making the computer readily accessible to the engineer for any purpose. Such programs are also undesirable because of the understandable tendency of the designer to make his structure fit the available program.

The most serious deficiency in the current mode of computer usage in structural engineering is the lack of direct communication between the engineer as such and the machine. The engineer has had two choices; He could become a programmer himself or he could turn the analysis over to a middleman who was a computer expert but probably did not fully understand the structural problem. The first choice is impractical because the broader aspects of design fully tax the engineer's capacity leaving little time for diversion into another field. The second choice is undesirable because the engineer loses control of his own design.

STRESS represents an attempt to automate structural analysis and, to some extent, design in such a way that some of the deficiencies in current practice are alleviated. The system has three characteristics of basic importance which distinguish it from previous efforts: (1) The only input required is in engineering rather than machine language thus making possible full use of the system by an engineer not trained in computer programming; (2) It is a general purpose program capable of handling structures of many types which include the majority of analysis problems encountered in structural engineering; and (3) Modifications of the original structure may be easily made thus expediting the iterative design process. The latter capability is most effective when STRESS is used in the time-sharing mode which permits an engineer to actually design a structure while sitting at a console. Design is a decision-making process and the role of STRESS in the time-sharing mode is to immediately process the data on which decisions are based.

The types of structures which can be analyzed by STRESS include those shown in Fig. 2. The structure may be either two-or-three-dimensional and the joints may be considered pinned as in a riveted truss or rigid as in a welded steel frame. The members may be prismatic or may vary in cross-section along the length. The loading may be in the form of concentrated or distributed forces and moments or may be the result of temperature changes or support movement. This generality of application is unique since the structures shown in Fig. 2 are usually considered to be of distinct types requiring different methods of analysis.

A typical STRESS problem description is shown in Fig. 3. Although the example is trivial it serves to demonstrate the simplicity of the STRESS input language. The input, which is completely shown in the figure, can be written in a matter of minutes. An analysis of this structure by hand computation would require approximately one hour. Thus, even for this very simple case, the use of a computer becomes economical. For [illegible] structures the input program is expanded only by the additional numerical data required and [illegible] more advantageous.

The important point to be made in connection with Fig. 3 is that the program consists of engineering terms such as [illegible] "member", etc. An engineer trained in structural analysis can learn the STRESS language in a few hours. He is then in a position to analyze by computer the majority of structures which he encounters in practice. In other words, the engineer who will make the design decisions is in direct communication with the machine on his own terms. By this means the use of computers in structural engineering becomes economical, not only for the large, complex problems as at present, but for the routine, day-to-day analysis which comprises the bulk of professional practice.

[illegible] of Structural Analysis

[illegible] interest has recently been shown in the application of [illegible] the framed structures problem. Kron[1] showed the analogies between the electrical network and the structural network. Fenves joined Branin[2] in formulating and solving the linear structural analysis problem using network theory. The method of analysis used in STRESS is based on this work, although the method has more recently been derived more explicitly by Connor[3].

The electrical network consists of branches and nodes, with scalar quantities being associated with the properties and variables. The structural analogy to the branch is the member, the joint to the node. The unknown at a point in a member is not a scalar, but a vector. The

A typical STRESS problem description is shown in Fig. 3. Although the example is trivial it serves to demonstrate the simplicity of the STRESS input language. The input, which is completely shown in the figure, can be written in a matter of minutes. An analysis of this structure by hand computation would require approximately one hour. Thus, even for this very simple case, the use of a computer becomes economical. For larger structures the input program is expanded only by the additional numeric data required and the use of STRESS becomes more advantageous.

The important point to be made in connection with Fig. 3 is that the program consists of engineering terms such as "frame", "joint", "member", etc. An engineer trained in structural analysis can learn the STRESS language in a few hours. He is then in a position to analyze by computer the majority of structures which he encounters in practice. In other words, the engineer who will make the design decisions is in direct communication with the machine on his own terms. By this means the use of computers in structural engineering becomes economical, not only for the large, complex problems as at present, but for the routine, day-to-day analysis which comprises the bulk of professional practice.

Formulation of Structural Analysis

A great deal of interest has recently been shown in the application of network theory to the framed structures problem. Branin[1] showed the analogies between the electrical network and the structural network. Fenves joined Branin[2] in formulating and solving the linear structural analysis problem using network theory. The method of analysis used in STRESS is based on this work, although the method has more recently been derived more explicitly by Connor[3].

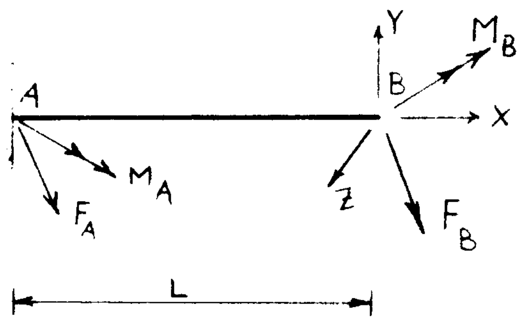
The electrical network consists of branches and nodes, with scalar quantities being associated with the properties and variables. The structural analogy to the branch is the member, the joint to the node. The unknown at a point in a member is not a scalar, but a vector. The

```
STRUCTURE [illegible]
TYPE PLANE FRAME
NUMBER OF JOINTS 5
NUMBER OF MEMBERS 4
NUMBER OF SUPPORTS 2
NUMBER OF LOADINGS 1
METHOD STIFFNESS
JOINT COORDINATES
1 0. 0. S
2 0. 150.
3 450. 225.
4 500. 150.
5 500. 0. S
MEMBER INCIDENCES
1 1 2
2 2 3
3 3 4
4 4 5
MEMBER PROPERTIES PRISMATIC
1 AX 10. IZ 1000.
2 AX 10. IZ 1500.
3 AX 10. IZ [illegible]
4 AX 10. IZ 1000.
LOADING WIND
JOINT 2 LOAD X [illegible]
MEMBER 2 LOAD FORCE Y UNIFORM [illegible]
TABULATE FORCES REACTIONS DISPLACEMENTS
SOLVE
```

joint variables also are vectors. The member unknown at another point in the member is related, not by a linear transformation, but by a matrix transformation. Figure 4 illustrates the force transformation from one end of a straight member to the other, with no forces applied in between. The general force vector for the three dimensional structure consists of three linear force components in an orthogonal system and three moment components acting about the axes. The general transformation matrix then has the size 6 x 6. Each column corresponds to the effect of a particular component on the right side of the equation, each row is used for the computation of one of the components on the left side. It can be shown that the displacement transformation is similar and equal to the transpose of the inverse of the force transformation.

The network concept allows us to readily deal conceptually with vectors of different sizes for different type of structures. The number of unknowns is then a function of the type, while the method of solution is not. When stated so simply this result may seem obvious, but the fact is that for hand computation different methods have been used for different structure types. As a result, many computer programs have been written for the linear analysis of framed structures, each treating only one of the types listed in Table 1.

Considering a displacement or stiffness method of analysis, Table 1 also shows the minimum number of unknown vector components per joint (JF). For structural types other than the space frame the equations for the analysis can easily be formed by reducing the six equations per joint required for the space frame. By taking consistent axes the force and displacement components not shown in Table 1 are always zero and need not be considered. These zero values may be omitted from the vectors and the corresponding rows and columns deleted from the transformations. Figure 5 shows schematically the deletions for a plane frame. Not only is the number of simultaneous equations necessary to solve held to a minimum, but almost all of the program is independent of structural type, related only by JF, the joint displacement vector size.



$$\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ M_1 \\ M_2 \\ M_3 \end{Bmatrix}_A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -L & 0 & 1 & 0 \\ 0 & L & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ M_1 \\ M_2 \\ M_3 \end{Bmatrix}_B$$

Fig. 4. Force Transformation for a Straight Member

$$\begin{aligned} F_3 &= 0 \\ M_1 &= 0 \\ M_2 &= 0 \end{aligned} \qquad \begin{Bmatrix} F_1 \\ F_2 \\ \cancel{0} \\ \cancel{0} \\ \cancel{0} \\ M_3 \end{Bmatrix}_A = \begin{bmatrix} 1 & 0 & \cancel{0} & \cancel{0} & \cancel{0} & 0 \\ 0 & 1 & \cancel{0} & \cancel{0} & \cancel{0} & 0 \\ \cancel{0} & \cancel{0} & \cancel{1} & \cancel{0} & \cancel{0} & \cancel{0} \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{1} & \cancel{0} & \cancel{0} \\ \cancel{0} & \cancel{0} & \cancel{-L} & \cancel{0} & \cancel{1} & \cancel{0} \\ 0 & L & \cancel{0} & \cancel{0} & \cancel{0} & 1 \end{bmatrix} \begin{Bmatrix} F_1 \\ F_2 \\ \cancel{0} \\ \cancel{0} \\ \cancel{0} \\ M_3 \end{Bmatrix}_B \qquad \begin{Bmatrix} F_1 \\ F_2 \\ M_3 \end{Bmatrix}_A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & L & 1 \end{bmatrix} \begin{Bmatrix} F_1 \\ F_2 \\ M_3 \end{Bmatrix}_B$$

Fig. 5. Reduction of Transformation for a Plane Frame

Table 1 Member and Joint Unknowns

Type	Member Unknowns	Joint Unknowns	JF Number of Joint Unknowns
Plane Truss	F_1	U_1 U_2	2
Plane Frame	F_1 F_2 M_3	U_1 U_2 U_6	3
Grid	F_3 M_1 M_2	U_3 U_4 U_5	3
Space Truss	F_1	U_1 U_2 U_3	3
Space Frame	F_1 F_2 F_3 M_1 M_2 M_3	U_1 U_2 U_3 U_4 U_5 U_6	6

STRESS is intended to be an informative and easily usable structural design tool. The designer then must be able to specify his problem to the machine easily, rapidly and concisely. He should be able to specify the problem as he thinks of it, not in terms of how the machine solves it. He should be able to specify a problem without performing any computations during data preparation. This implies that the processor will deal with much more information than merely the generation and solution of the analysis equations. For example, the equations relate imbalanced joint forces which can be computed from a variety of load types considered by the designer. The machine will operate on joint coordinates while the designer might relate geometry to bays and stories, or spans. In the process of generating and solving the equations, and in this pre-and post-processing the machine must deal with a great amount of data, mostly in array form. The number of arrays and their sizes are variable functions of the input data, the structural type and size. The form and features of the STRESS processor are related to these problems and a desire that the processor be a dynamic entity expandable by engineers.

The STRESS Processor

The STRESS processor may be [illegible] phases, an input and compilation phase, [illegible] execution phase. Figure 6 gives a macro flow chart of [illegible]. PHASE1A controls the input translation and data storage. [illegible] performs consistency checks to determine if the problem has been completely specified and can be solved. It also defines [illegible]. The execution phase is divided into eight [illegible] which are not all needed for a particular problem. Table [illegible] summarizes the functions of these major blocks.

STRESS contains two programming features [illegible] from the requirement that the processor basically be written in a well known compiler language, i.e., FORTRAN, [illegible] be further developed and altered by engineers for [illegible]. These features attempt to circumvent the rigid restrictions [illegible] by the FORTRAN DIMENSION and input FORMAT statements in [illegible] the flexibility needed to have a processor [illegible] the wide range of interest.

The FORTRAN dimension statement [illegible] that the maximum size of arrays permissible in a program [illegible] at the time the program is developed or compiled. These arrays [illegible] have a fixed location for reference. If all the arrays do not [illegible] simultaneously, secondary storage [illegible] organization is then a complex chore for [illegible]. In addition, for small problems, all arrays could fit [illegible] if only needed portions of arrays were [illegible] secondary storage is ideally a function of the problem [illegible] data. In order to develop a processor which is in [illegible] over a wide range of problem size the [illegible] a function of that size.

In order to make memory usage [illegible] of the problem size, an automatic memory control process [illegible] for FORTRAN usage. This permits the programmer to define [illegible] function of data, reserve

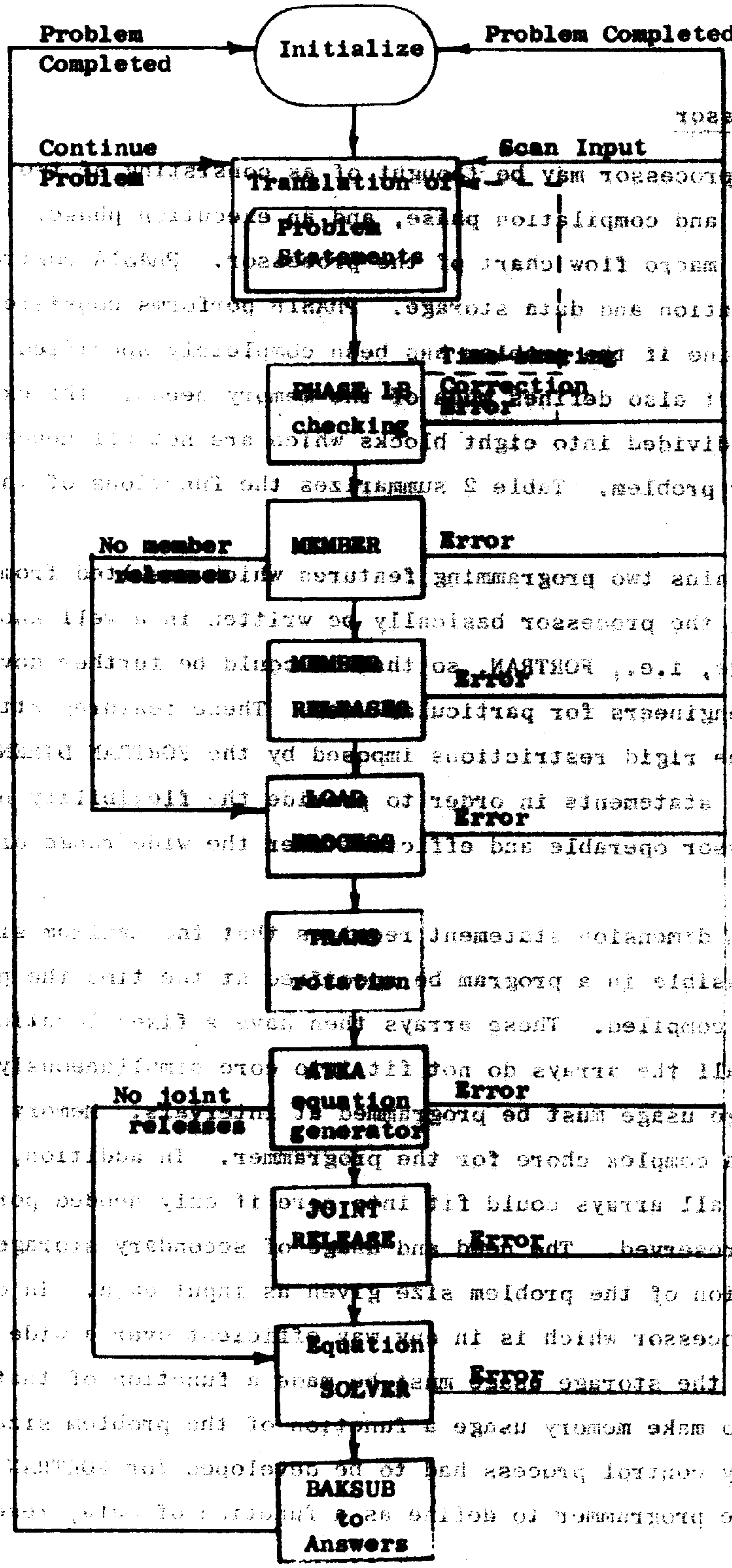


Figure 6. System Block Flow Chart

Table 2

Program Blocks

NAME	PROCESS
PHASE 1A	Translation
PHASE 1B	Consistency check, Internal representation
MEMBER	Compute member stiffness matrices
MRELES	Modify stiffness matrices for member end releases
LOAD PROCESSOR	Process all types of raw load data into equivalent joint loads
TRANS	Rotate member stiffness matrices into global coordinates
ATKA	Generate symbolically structural stiffness matrix
JRELES	Modify stiffness matrix and joint loads for joint releases
SOLVER	Solve, matrix equation for joint displacements
BAKSUB	Backsubstitute for other results and print.

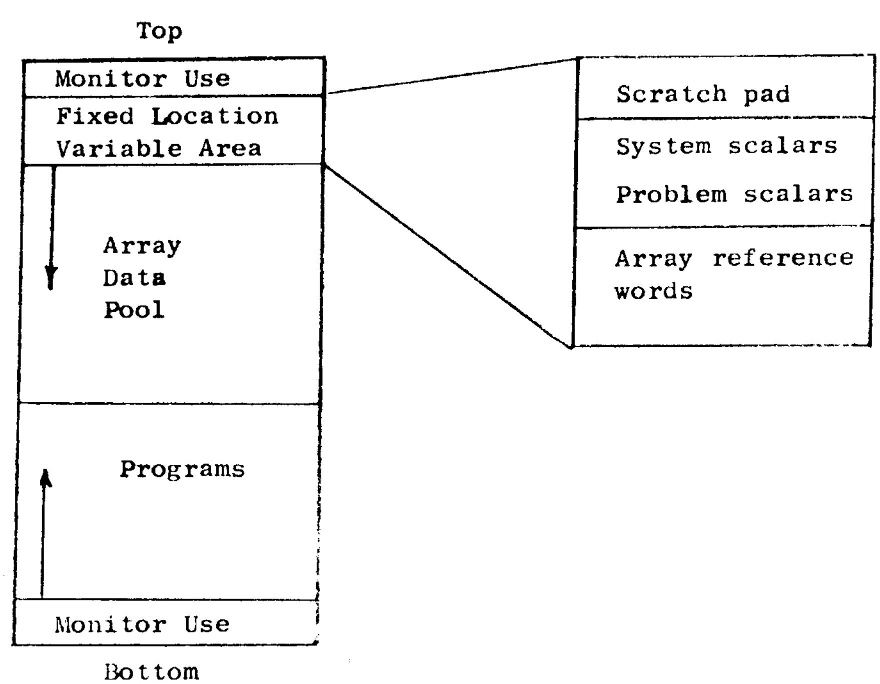


Figure 7. Core Memory Layout

space for, reference and use the arrays without requiring the location of the array to be fixed or even constant during a part of the solutions process. Figure 7 shows the normal core arrangement for STRESS operation. A very small area (about 300 words) of fixed location variables is included in upper core. Part of this area contains codewords which are used to reference the arrays. These codewords contain such information as the array size and location, in or out of core. All of the pertinent information can be changed by program control. The remaining memory down to the top of programs consists of a pool for arrays. When the pool is full, it is reorganized, using secondary storage.

The amount of program comprising the system has exceeded core capacity. Program blocks must then be swapped during processing. The FORTRAN Chain feature, with modifications, is used for this purpose. With each program block there is a different top of programs, or bottom of the pool. This results in a constantly varying data memory capacity, easily accounted for by the memory controller. A slightly different form of the memory is used with time-sharing, but this is conceptually no different.

The use of explicit FORMAT statements requires that a programmer know the form of an input card or line before recognizing the first character. In addition very rigid restrictions are placed on character position. This is inconsistent both with the rest of FORTRAN, which allows great freedom in source program format and elegant output, and the engineers scope of concern. It is necessary to provide the engineer with a free and easy form of input, free field format and great freedom in statement ordering.

A single small subroutine was written to do operations on logical (rather than physical) input fields, performing dictionary look-up, binary conversion, etc. This routine is called for every logical data field during translation of input data by translation programs written in FORTRAN. The programmer then has the input capabilities usually found only in a compiler or other extensive assembly language programs.

The input programs are then [illegible] to a compiler language compiler.

These [illegible] are [illegible] to eliminate some compiler restrictions and [illegible] [illegible]. The value of the compiler languages for development of dynamic engineering systems is unquestionably great. We only [illegible] [illegible] [illegible] [illegible] their elimination in the next generation compilers.

__Sample Usage__

In order to illustrate in some small way the capabilities of STRESS, the specification and solution of a simple problem is given. Figure 8 shows a small building [illegible] to be analyzed for a vertical load of 1.5 kips/ft. [illegible] on all horizontal members and a 20 kips per floor wind loading. Figure 9 shows the computer oriented representation of the structure with location numbering of joints and members, and defining the member orientations. Units are also made consistent. Figure 10 shows the STRESS input, which is an attempt to [illegible] [illegible] language for easy understanding.

The STRUCTURE statement serves to identify the problem and title the output. The [illegible] statement [illegible] the problem size and together with the [illegible] [illegible] [illegible]. The TYPE statement [illegible] [illegible] of interest. The requested results are [illegible] in the [illegible] statement, while the geometry is described [illegible] [illegible], shown here in tabular form. The [illegible] are [illegible] by [illegible] [illegible], and the [illegible] [illegible]. Section properties of [illegible] [illegible] in a wide variety of ways. Only the [illegible] [illegible] [illegible] here. Material properties [illegible] [illegible]. The loadings are then specified as [illegible] [illegible] with a [illegible] statement used to separate loading conditions [illegible] [illegible] the output. Solution proceeds upon [illegible] [illegible] if the

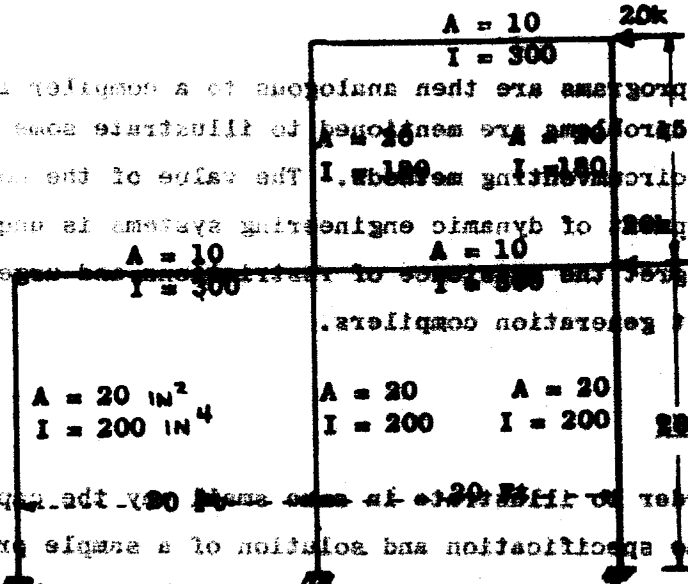


Fig. 8
Sample Structure

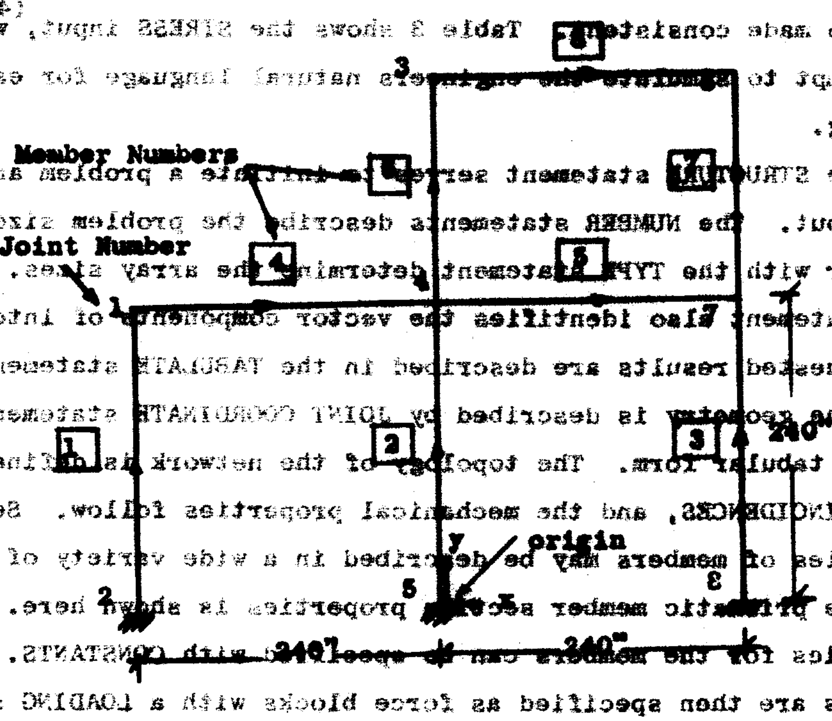


Fig. 9
Numbering of Members and Joints

Table 3
Sample Problem Specification

```
00010 STRUCTURE SAMPLE PROBLEM
00020 NUMBER OF JOINTS 8
00030 NUMBER OF MEMBERS  8
00040 NUMBER OF SUPPORTS 3
00050 NUMBER OF LOADINGS 2
00060 TYPE PLANE FRAME
00070 METHOD STIFFNESS
00080 TABULATE FORCES, REACTIONS
00090 JOINT COORDINATES
00100 1 X -240. Y 240. FREE
00110 2 X -240.     SUPPORT
00120 5 X 0. S
00130 8 X 240. S
00140 4 Y 240.
00150 7 X 240. Y 240.
00160 3 Y 420.
00170 6 X 240. Y 420.
00180 MEMBER INCIDENCES
00190 1 2 1
00200 2 5 4
00210 3 8 7
00220 4 1 4
00230 5 4 7
00240 6 4 3
00250 7 7 6
00260 8 3 6
00270 MEMBER PROPERTIES
00280 8 PRISMATIC AX 10. IZ 300.
00290 4 PRISMATIC AX 10. IZ 300.
00300 MEMBER PROPERTIES PRISMATIC
00310 1 AX 20. IZ 200.
00320 2 AX 20. IZ 200.
00330 3 AX 20. IZ 200.
00340 5 AX 10. IZ 300.
00350 6 IZ 180. AX 20.
00360 7 IZ 180. AX 20.
00370 CONSTANTS E 30000. ALL
00380 LOADING 1 UNIFORM ALL BEAMS
00390 MEMBER LOADS
00400 8 FORCE Y UNIFORM -0.1
00410 4 FORCE Y UNIFORM -0.1
00420 5 FORCE Y UNIFORM -0.1
00430 LOADING 2 WIND FROM RIGHT
00440 JOINT LOADS
00450 6 FORCE X -20.
00460 7 FORCE X -20.
00480 SOLVE THIS PART
```

statements prior to this command constitute a complete and consistent problem.

For efficient use of time-sharing, the input is typed in using the CTSS monitor input program in a form which STRESS can accept and execute. The remote console is used for controlling the processor and for immediate correction of errors so as not to delay the design. Answers to the specified problem are shown in Table 4.

The results show the forces acting on the member ends and acting on the joints. With the solution of the member end forces, the member is statically determinate, so that the forces and deformations in the interior of the member can be determined by elementary methods. Up to now the development of STRESS has concentrated on the overall problem. We are now, however, attacking such problems as the interior forces to develop a more effective design aid. The joint loads on support joints represent the reactions. While the difference between the calculated joint loads and the applied joint loads gives a measure of the solution accuracy.

The engineer may then wish to alter the problem for his developing design. In most cases the alterations will be a function of the obtained results which were not known during creation of the input file. He might then describe the differences in the new problem to the processor and obtain results for immediate comparison and evaluation of the merits of the tact of the design. Table 5 shows the changes necessary to analyze the same structure with new member properties as suggested by the first analysis. Table 6 shows the effects of the changes.

The STRESS system is in a continuing state of development. It is expected that its capability will be extended to include dynamic analysis, investigation of structural stability, and the behavior of inelastic structures. It is hoped that ultimately STRESS will become part of a larger system which will be an aid to automatic structural optimization.

Table 4
Sample Problem Results

STRUCTURE SAMPLE PROBLEM
LOADING 1 UNIFORM ALL BEAMS

MEMBER FORCES

MEMBER	JOINT	AXIAL FORCE	SHEAR FORCE	BENDING MOMENT
1	2	10.545	-1.229	-92.604
1	1	-10.545	1.229	-202.414
2	5	38.982	0.481	44.045
2	4	-38.982	-0.481	71.498
3	8	22.473	0.748	65.663
3	7	-22.473	-0.748	113.813
4	1	1.229	10.545	202.414
4	4	-1.229	13.455	-551.690
5	4	-1.846	13.366	628.394
5	7	1.846	10.634	-300.563
6	4	12.161	-2.594	-148.203
6	3	-12.161	2.594	-318.751
7	7	11.839	2.594	186.750
7	6	-11.839	-2.594	280.204
8	3	2.594	12.161	318.751
8	6	-2.594	11.839	-280.204

STRUCTURE SAMPLE PROBLEM
LOADING 1 UNIFORM ALL BEAMS

JOINT LOADS

JOINT	X FORCE	Y FORCE	MOMENT
		SUPPORT REACTIONS	
2	1.2292	10.5447	-92.6044
5	-0.4814	38.9819	44.0448
8	-0.7478	22.4734	65.6634
		APPLIED JOINT LOADS	
1	-0.0000	0.0000	-0.0000
3	0.0000	0.0000	0.0000
4	0.0000	-0.0000	-0.0000
6	-0.0000	0.0000	-0.0000
7	0.0000	0.0000	-0.0000

STRUCTURE SAMPLE PROBLEM
LOADING 2 WIND FROM RIGHT

MEMBER FORCES

MEMBER	JOINT	AXIAL FORCE	SHEAR FORCE	BENDING MOMENT
1	2	11.195	-13.334	-1776.267
1	1	-11.195	13.334	-1423.969
2	5	10.377	-14.732	-1890.313
2	4	-10.377	14.732	-1645.392
3	8	-21.573	-11.934	-1669.204
3	7	21.573	11.934	-1194.941
4	1	13.334	11.195	1423.969
4	4	-13.334	-11.195	1262.902
5	4	16.467	13.385	1434.174
5	7	-16.467	-13.385	1778.188
6	4	8.188	-11.599	-1051.684
6	3	-8.188	11.599	-1036.215
7	7	-8.188	-8.401	-583.247
7	6	8.188	8.401	-928.867
8	3	11.600	8.188	1036.215
8	6	-11.600	-8.188	928.867

STRUCTURE SAMPLE PROBLEM
LOADING 2 WIND FROM RIGHT

JOINT LOADS

JOINT	X FORCE	Y FORCE	MOMENT
		SUPPORT REACTIONS	
2	13.3343	11.1953	-1776.2675
5	14.7321	10.3774	-1890.3127
8	11.9339	-21.5727	-1669.2038
		APPLIED JOINT LOADS	
1	0.0000	-0.0000	-0.0000
3	0.0001	0.	-0.0000
4	-0.0002	0.0000	-0.0000
6	-20.0001	-0.0000	-0.0000
7	-20.0001	0.0000	-0.0000

PART 1 OF PROBLEM COMPLETED.

Table 5
Modification Specifications

```
STRESS IS READY FOR INPUT.
TYPE
modification of first part - second cycle for member sizes
TYPE
changes
TYPE
member properties prismatic
TYPE
1 iz 800.6
TYPE
2 iz 889.9
TYPE
3 iz 800.6
TYPE
4 iz 583.3
TYPE
5 iz 800.6
TYPE
6 iz 446.3
TYPE
7 iz 339.2
TYPE
8 iz 446.3
TYPE
solve
PROBLEM CORRECTLY SPECIFIED.  SOLUTION WILL PROCEED.
```

Table 6
Modification Results

STRUCTURE SAMPLE PROBLEM
MODIFICATION OF FIRST PART - SECOND CYCLE FOR MEMBER SIZES
LOADING 1 UNIFORM ALL BEAMS

MEMBER FORCES

MEMBER	JOINT	AXIAL FORCE	SHEAR FORCE	BENDING MOMENT
1	2	10.982	-1.729	-127.062
1	1	-10.982	1.729	-287.964
2	5	38.177	0.712	68.235
2	4	-38.177	-0.712	102.705
3	8	22.841	1.017	92.710
3	7	-22.841	-1.017	151.376
4	1	1.729	10.982	287.964
4	4	-1.729	13.018	-532.213
5	4	-1.831	12.993	585.341
5	7	1.831	11.007	-347.003
6	4	12.166	-2.848	-155.832
6	3	-12.166	2.848	-356.873
7	7	11.834	2.848	195.627
7	6	-11.834	-2.848	317.079
8	3	2.848	12.166	356.873
8	6	-2.848	11.834	-317.079

STRUCTURE SAMPLE PROBLEM
MODIFICATION OF FIRST PART - SECOND CYCLE FOR MEMBER SIZES
LOADING 1 UNIFORM ALL BEAMS

JOINT LOADS

JOINT	X FORCE	Y FORCE	MOMENT
		SUPPORT	REACTIONS
2	1.7293	10.9823	-127.0625
5	-0.7123	38.1766	68.2353
8	-1.0170	22.8411	92.7096
		APPLIED	JOINT LOADS
1	-0.0000	0.0000	0.0000
3	0.0000	0.0000	0.0000
4	-0.0000	-0.0000	-0.
6	-0.0000	0.0000	-0.0000
7	0.0000	0.0000	-0.0000

STRUCTURE SAMPLE PROBLEM
MODIFICATION OF FIRST PART - SECOND CYCLE FOR MEMBER SIZES
LOADING 2 WIND FROM RIGHT

MEMBER FORCES

MEMBER	JOINT	AXIAL FORCE	SHEAR FORCE	BENDING MOMENT
1	2	9.680	-12.474	-1790.689
1	1	-9.680	12.474	-1203.048
2	5	11.910	-15.680	-2144.816
2	4	-11.910	15.680	-1618.268
3	8	-21.589	-11.847	-1760.015
3	7	21.589	11.847	-1083.169
4	1	12.474	9.680	1203.047
4	4	-12.474	-9.680	1120.039
5	4	16.769	14.103	1590.743
5	7	-16.769	-14.103	1793.879
6	4	7.487	-11.384	-1092.514
6	3	-7.487	11.384	-956.610
7	7	-7.487	-8.616	-710.710
7	6	7.487	8.616	-840.170
8	3	11.384	7.487	956.610
8	6	-11.384	-7.487	840.170

STRUCTURE SAMPLE PROBLEM
MODIFICATION OF FIRST PART - SECOND CYCLE FOR MEMBER SIZES
LOADING 2 WIND FROM RIGHT

JOINT LOADS

JOINT	X FORCE	Y FORCE	MOMENT
		SUPPORT	REACTIONS
2	12.4739	9.6795	-1790.6894
5	15.6795	11.9096	-2144.8164
8	11.8466	-21.5892	-1760.0153
		APPLIED	JOINT LOADS
1	0.0000	0.0000	-0.0000
3	0.0000	0.0000	0.0000
4	0.0000	-0.0000	0.0000
6	-20.0000	-0.0000	0.0000
7	-20.0000	0.0000	-0.0000

PROBLEM COMPLETED.

The development reported herein is the work of a group within the Civil Engineering Department at M.I.T. Special credit is due Prof. S. J. Fenves of the University of Illinois who was a visiting member of the M.I.T. faculty during the year 1962-63 and was largely responsible for the initial concept.

The STRESS project has been partially supported from a major grant for the improvement of engineering education made to M.I.T. by the Ford Foundation. Additional support was provided by Project MAC, an M.I.T. research program, sponsored by the Advanced Research Project Agency, Department of Defense, under Office of Naval Research, Contract No. Nonr-4102(01). Reproduction in whole or in part is permitted for any purpose of the U.S. Government. The work was done in part at the M.I.T. Computation Center, and the aid and support of the Center and its personnel are gratefully acknowledged.

References:

1. "Machine Analysis of Networks and Its Applications," by F. H. Branin, Jr., Technical Report 855, IBM Development Laboratory, Poughkeepsie, New York.

2. "A Network-Topological Formulation of Structural Analysis," S. J. Fenves and F. H. Branin, Jr., Journal of the Structures Division, A.S.C.E., August, 1963.

3. "Notes on Matrix Analysis," J. Connor, Civil Engineering Department, M.I.T., Fall 1963. (Unpublished)

4. "STRESS: A User's Manual," S. J. Fenves, R. D. Logcher, S. P. Mauch, and K. F. Reinschmidt, M.I.T. Press, Cambridge, Massachusetts, March 1964.