

AD-A084 654

WEIDLINGER ASSOCIATES NEW YORK

F/G 9/2

TRANAL USER'S GUIDE, PART 1. (SMALL STRAIN, SMALL DISPLACEMENT —ETC(U)

MAR 79 J L BAYLOR, J P WRIGHT, C F CHUNG

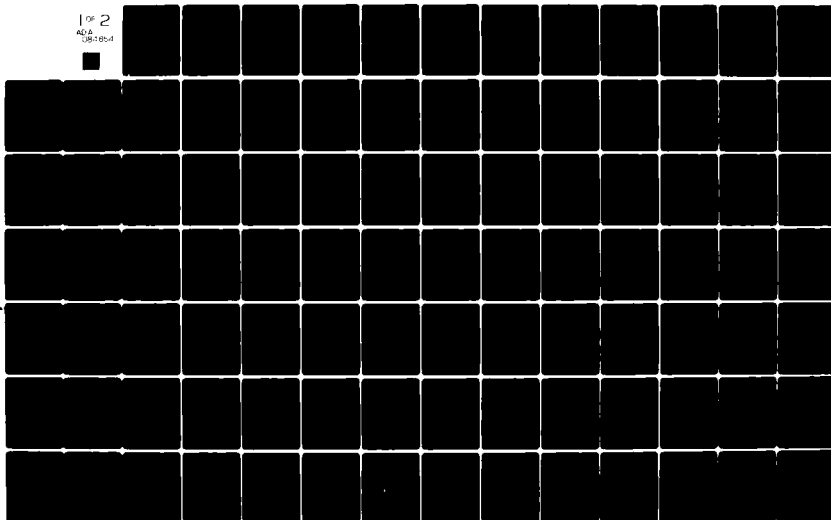
DNA001-76-C-0125

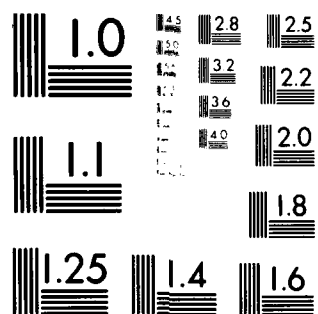
UNCLASSIFIED

DNA-4960F

NL

1 of 2
ADA
180-054





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

LEVEL III

AD-E 300 767

12

DNA 4960F

ADA 084654

TRANAL USER'S GUIDE, PART I (SMALL STRAIN, SMALL DISPLACEMENT VERSION)

Weidlinger Associates
110 East 59th Street
New York, New York 10022

1 March 1979

Final Report for Period 1 October 1975—31 December 1978

CONTRACT No. DNA 001-76-C-0125

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.

THIS WORK SPONSORED BY THE DEFENSE NUCLEAR AGENCY
UNDER RDT&E RMSS CODE B344076464 Y99QAXSC06202 H2590D.

Prepared for
Director
DEFENSE NUCLEAR AGENCY
Washington, D. C. 20305

DTIC
ELECTED
MAY 23 1980
S D

DC FILE COPY

80 4 12 054

Destroy this report when it is no longer
needed. Do not return to sender.

PLEASE NOTIFY THE DEFENSE NUCLEAR AGENCY,
ATTN: STTI, WASHINGTON, D.C. 20305, IF
YOUR ADDRESS IS INCORRECT, IF YOU WISH TO
BE DELETED FROM THE DISTRIBUTION LIST, OR
IF THE ADDRESSEE IS NO LONGER EMPLOYED BY
YOUR ORGANIZATION.



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER DNA 4960F ✓	2. GOVT ACCESSION NO. AD-A084657	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) TRANAL USER'S GUIDE, PART I (SMALL STRAIN, SMALL DISPLACEMENT VERSION)		5. TYPE OF REPORT & PERIOD COVERED Final Report for Period 1 Oct 75—31 Dec 78
7. AUTHOR(s) J. L. Baylor J. P. Wright C. F. Chung		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Weidlinger Associates ✓ 110 East 59th Street New York, New York 10022		8. CONTRACT OR GRANT NUMBER(s) DNA 001-76-C-0125 ✓
11. CONTROLLING OFFICE NAME AND ADDRESS Director Defense Nuclear Agency Washington, D.C. 20305		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Subtask Y99QAXSC062-02
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE 1 March 1979
		13. NUMBER OF PAGES 156
		15. SECURITY CLASS (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES This work sponsored by the Defense Nuclear Agency under RDT&E RMSS Code B344076464 Y99QAXSC06202 H2590D.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) TRANAL Three-Dimensional Calculations		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) TRANAL (transient nonlinear analysis) is a system of computer programs for doing nonlinear three-dimensional structure-medium interaction calculations. This report describes the small strain, small displacement version of TRANAL; the large strain, large displacement version is described in a separate report. Theoretical background, input data, program organization,		

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. ABSTRACT (Continued)

graphics capability, data management and sample problems are discussed in sufficient detail that a new user should be able to set up and run a calculation.

TRANAL is based on a finite element approach for the development of the discrete form of the equations of motion and an eight-noded isoparametric element is used to represent the continuum. Material behavior is represented by the cap model, which is based on classical elastic, nonideal plasticity theory with an associated flow rule. An explicit time integration scheme is used, and a special integration technique, called "subcycling", permits the use of different size time steps in different elements, thereby reducing computation time considerably.

Due to their complexity, transient nonlinear 3-D calculations generally require relatively large computational capability. Hence, this version of TRANAL has been written primarily for the CDC 7600 (although it is operational on several other CDC systems, including a 6600, a CYBER/175 and a CYBER/176, and could be transferred to other large computer systems)

Accession For	
NTIS GRA&I	☒
DDC TAB	☐
Unannounced	☐
Justification_____	
By_____	
Distribution/	
Availability Codes	
Dist.	Avail and/or special
☒	☐

DTIC
ELECTE
MAY 23 1980
S D D

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

PREFACE

This is the first of several reports intended to document the use of TRANAL which was developed, under DNA sponsorship, to provide improved capability for doing three-dimensional structure-medium interaction calculations. Although such calculations were done in the 1960's, it has only been during the last several years that large calculations involving more than 50,000 degrees of freedom have been performed routinely, as part of protective structures technology development. Documentation for a code like TRANAL should include information on theory, design, implementation, input data, operating instructions, sample problems, applications, etc. An attempt has been made to cover all of these areas as fully as possible. Nonetheless, it should be noted that many details have of necessity, been omitted, and that specialized coding, created for certain applications, exists but is not included here. Additional information about the code can be obtained by contacting Weidlinger Associates or the Defense Nuclear Agency.

We would like to acknowledge the support of DNA, particularly E. Sevin and K. Goering of SPSS for their cooperation and encouragement in the development of TRANAL. We would also like to acknowledge the many useful conversations with M. Baron, I. Sandler, J. Isenberg, H. Levine, D. Vaughan, and F. Wong, all of Weidlinger Associates. Finally, we would like to note the contribution of G. Fenves who, as a summer student, developed the free-format routines on which Section 7 is based.

TABLE OF CONTENTS

<u>Section</u>		<u>Page</u>
1	INTRODUCTION	1-1
2	SOIL ISLAND METHOD	2-1
3	METHOD OF ANALYSIS	3-1
	3-1 Equations of Motion	3-1
	3-2 Constitutive Models	3-2
	3-3 Three Dimensional Solid Element	3-4
	3-4 Integration Scheme.	3-7
	3-5 Pseudo-Viscosity.	3-8
	3-6 Numerical Stability	3-9
	3-7 Subcycling	3-12
4	FINITE ELEMENT REPRESENTATION OF SOIL ISLAND AND STRUCTURE.	4-1
	4-1 Coordinate System and Element Subdivision	4-1
	4-2 Region Definition	4-1
	4-3 Material Input.	4-1
	4-4 Coordinate Input.	4-1
	4-5 Mesh Generation	4-1
5	FILES.	5-1
	5-1 Data Files.	5-1
	5-2 TRANAL Files.	5-1
6	PROGRAM FUNCTIONAL STRUCTURE	6-1
	6-1 Zone Definition	6-1
	6-2 Zone Processing	6-1
	6-3 Data Management	6-1
	6-4 Error Processing.	6-2
	6-5 Subcycling	6-2
	6-6 Zone Interfaces	6-2
	6-7 Boundary Conditions	6-3
7	INPUT LANGUAGE	7-1
	7-1 Directive Format.	7-1
	7-2 Input Language Description.	7-1
	7-3 Input Constructions	7-2
	7-3.1 Entities	7-2
	7-3.2 Separators	7-2
	7-3.3 Comments	7-2
	7-3.4 Range Specification.	7-2
	7-3.5 Keywords	7-3
	7-4 Alternate Input Files	7-4
	7-5 Error Messages.	7-4
	7-6 Directive Abbreviations	7-4

TABLE OF CONTENTS (Cont'd)

<u>Section</u>	<u>Page</u>
8 PROCESSORS AND INPUT.	8-1
8-1 START UP Processor	8-1
8-2 START UP Input Directives.	8-1
8-2.1 TITLE	8-1
8-2.2 IDENTIFIER.	8-2
8-2.3 PRINT OPTIONS	8-2
8-2.4 ZONE DEFINITION	8-3
8-2.5 ZONE TYPE	8-4
8-2.6 MATERIALS	8-4
8-2.7 STABILITY	8-5
8-2.8 DAMPING	8-6
8-2.9 REGION MATERIALS.	8-6
8-2.10 REGION COORDS	8-7
8-3 ALTER FL Processor	8-8
8-4 ALTER FL Input Directives.	8-8
8-4.1 ALTER STABILITY	8-8
8-4.2 ALTER DAMPING	8-9
8-4.3 ALTER ELEMENT	8-10
8-4.4 ALTER NODE.	8-10
8-5 CLEAN UP Processor	8-11
8-6 CLEAN UP Input Directives.	8-12
8-6.1 PRINT OPTIONS	8-12
8-6.2 TIME DATA	8-12
8-6.3 SUBCYCLE RATIOS	8-12
8-6.4 ZONE SUBCYCLES.	8-13
8-6.5 BOUNDARY CONDITIONS	8-14
8-6.6 OUTPUT.	8-14
8-6.7 CORNER FACTOR	8-15
8-6.8 BRODE LOAD.	8-15
8-7 PROSSOR Processor.	8-16
8-8 PROSSOR Input Directives	8-17
8-8.1 TIME INCREMENTS	8-17
8-9 COMBINE Processor.	8-17
8-10 COMBINE Input Directives	8-17
8-10.1 PRINT OPTIONS	8-17
8-10.2 COPY RECORDS.	8-18
8-11 TRNSPOS Processor.	8-18
8-12 TRNSPOS Input Directives	8-18
8-12.1 PRINT OPTIONS	8-18
8-12.2 TRANSPOSE	8-18
TABLE 8-1 PROCESSOR INPUT DIRECTIVES	8-20
TABLE 8-2 DIRECTIVE ABBREVIATIONS.	8-23

TABLE OF CONTENTS (Cont'd)

<u>Section</u>	<u>Page</u>
9	OUTPUT FILES
9-1	Output Data Structure (Tables 9-1 and 9-2).
9-2	Combined Output File (Table 9-3).
9-3	Transposed Output File (Table 9-4).
TABLE 9-1	PRIMARY RESULTS FILE (Tape 3)
TABLE 9-2	SECONDARY RESULTS OUTPUT FILE (Tape 3).
TABLE 9-3	COMBINED RESULTS OUTPUT FILE (Tape 70).
TABLE 9-4	TRANSPPOSED RESULTS OUTPUT FILE (Tape 8)
10	GRAPHICS
10-1	Structural Model Plotting (TGS3D).
10-1.1	Region and Zone Plotting
10-1.2	Input.
10-1.3	Coordinate Systems and Transformation.
10-1.4	Hidden-line Algorithm.
10-1.5	Files and System Dependent Routines.
10-2	Time History Plotting (TGS HIST).
11	PLOTTING PROCESSORS AND INPUT.
11-1	TGS3D Processor.
11-2	TGS3D Input Directives
11-2.1	AXIS
11-2.2	DIMENSION.
11-2.3	DISPLAY.
11-2.4	MATERIAL OPTION.
11-2.5	PRINT OPTION
11-2.6	OBJECT
11-2.7	TITLE
11-2.8	TRANSFORM.
11-2.9	VIEW OPTION.
11-3	TGS HIST Processor.
11-4	TGS HIST Input Directives
11-4.1	PLOT INFO.
11-4.2	PRINT OPTION
11-4.3	TIME INTERVAL.
11-4.4	TITLE
TABLE 11-1	PLOTTING PROCESSOR INPUT DIRECTIVES.
TABLE 11-2	PLOTTING DIRECTIVE ABBREVIATIONS
12	RUNNING A TRANAL JOB
12-1	Introduction
12-2	START UP Execution
12-2.2	Control Cards (Record 1)
12-2.2	Input Data (Record 2).
12-2.3	UPDATE Directives for TPL1 (Record 3).
12-2.4	UPDATE Directives for TRNLDIR (Record 4)

TABLE OF CONTENTS (Cont'd)

<u>Section</u>	<u>Page</u>
12	
12-3 TGS3D Execution	12-4
12-3.1 Control Cards	12-4
12-3.2 Geometry Plot	12-4
12-4 CLEAN UP Execution.	12-5
12-4.1 Control Cards	12-5
12-4.2 Input Data.	12-5
12-4.3 UPDATE Directives	12-5
12-5 First PROSSOR Execution	12-5
12-6 Second PROSSOR Execution.	12-5
12-7 COMBINE Execution	12-5
12-8 TRNSPOS Execution	12-6
12-9 TGS HIST Execution	12-6
12-9.1 Control Cards	12-6
12-9.2 Time History Plots.	12-6
TABLE 12-1 PROCESSOR DATA FILES.	12-7
TABLE 12-2 START UP EXECUTION DECK	12-9
TABLE 12-3 TGS3D EXECUTION DECK.	12-11
TABLE 12-4 CLEAN UP EXECUTION DECK	12-12
TABLE 12-5 FIRST PROSSOR EXECUTION DECK.	12-13
TABLE 12-6 SECOND PROSSOR EXECUTION DECK	12-14
TABLE 12-7 COMBINE EXECUTION DECK.	12-15
TABLE 12-8 TRNSPOS EXECUTION DECK.	12-16
TABLE 12-9 TGS HIST EXECUTION DECK.	12-17
TABLE 12-10 TRNLDIR SOURCE DECK	12-13
APPENDIX A NOTATIONS	A-1
APPENDIX B REFERENCES.	B-1
APPENDIX C FILES AND DATA STRUCTURES	C-1
C-1 Data Structure.	C-1
C-2 Random Data Files	C-1
C-3 Sequential Data Files	C-1
C-4 Restart Files	C-2
TABLE C-1 RECORD STRUCTURE OF NODE DATA FILES. . .	C-3
TABLE C-2 APPLIED BOUNDARY VELOCITY FILE (Tape 3).	C-4
APPENDIX D CDC UPDATE AND LIBGEN FILES	D-1
D-1 UPDATE Files.	D-1
D-2 Library Files	D-1
APPENDIX E DATA MANAGEMENT LIBRARY	E-1
E-1 Introduction.	E-1
E-2 Library Routines.	E-1

SECTION 1

INTRODUCTION

With the publication of several papers in the 1940's, John von Neumann laid the groundwork for both the current generation of digital computers and wave propagation calculations. Since then there has been continued evolution of computer capabilities so that it is now relatively easy to obtain computer codes for dynamic calculations of various types. One and two dimensional calculations which were unthinkable a generation ago are now considered routine, and three dimensional calculations have become practical. Much of the basic theory for this field can be found in Refs. 1 and 2.

This report describes the use of TRANAL, a system of computer programs for doing calculations involving three-dimensional stress wave propagation in nonlinear materials. Only the small strain, small displacement version is described here; the large strain, large displacement version is described in a forthcoming DNA report. Theoretical background, program organization, input data, and sample problems are discussed in sufficient detail that a new user should be able to set up and run a calculation.

TRANAL, Refs. 3 to 5, is based on a finite element approach, and an eight-noded isoparametric element is used to represent the continuum. Material behavior is represented by the cap model, Refs. 6 and 7, which is based on classical elastic, non-ideal plasticity theory with an associated flow rule. Since a modular approach is used, the cap model routine can easily be replaced by other suitable model routines.

An explicit time integration scheme, based on the central difference method, is used since explicit methods are more efficient than implicit methods for the class of calculations under consideration (large-scale, three-dimensional wave propagation). A special integration technique, called "subcycling", permits the use of different size time steps in different sets of elements, thereby reducing computation time considerably in many problems. Numerical stability is based on a simplified form of the energy method (Rayleigh's principle), Refs. 8 and 9.

Detailed three-dimensional simulations generally involve complicated multi-material configurations and require the use of large-scale computers. TRANAL was designed for such calculations and has been applied to a variety of situations, Refs. 10 to 14, such as:

- (a) a buried concrete protective structure with arched roof,
- (b) a reinforced concrete missile silo,
- (c) a scale model of a nuclear power plant containment structure.

Calculation times on the CDC 7600 are less than 0.5 milliseconds per element per time step, and problems involving more than 50,000 degrees of freedom (20,000 elements) have been solved.

The CDC 7600 small core memory (SCM) provides storage for about 1,000 elements. Hence, for large three-dimensional problems it is necessary to utilize either large core memory (LCM) or disk. The data transfer rate between SCM and LCM is sufficiently high that, with proper data management, all of LCM can be used to hold up to about 20,000 elements and yet the execution speed is degraded only to a small degree compared to a calculation using SCM alone. TRANAL was designed with a family of data management routines, Ref. 15, which can easily be extended to use SCM, LCM and disk in an efficient manner, thereby permitting considerably more than 20,000 elements. However, the data transfer rates to and from disk are relatively slow so that the turn-around time would

be greatly increased and the cost factor would more than double. Thus, although larger problems could be attempted on the CDC 7600, it is probably more productive to plan for advanced computers, such as the CRAY-1. The CRAY-1 is a vector processor which, used solely as a scalar processor, is faster than the 7600 and, in its largest configuration, will permit the storage of more elements than in LCM of the 7600. TRANAL has been developed with vector processors in mind and can readily be re-programmed to take advantage of the increased computational capabilities of these machines.

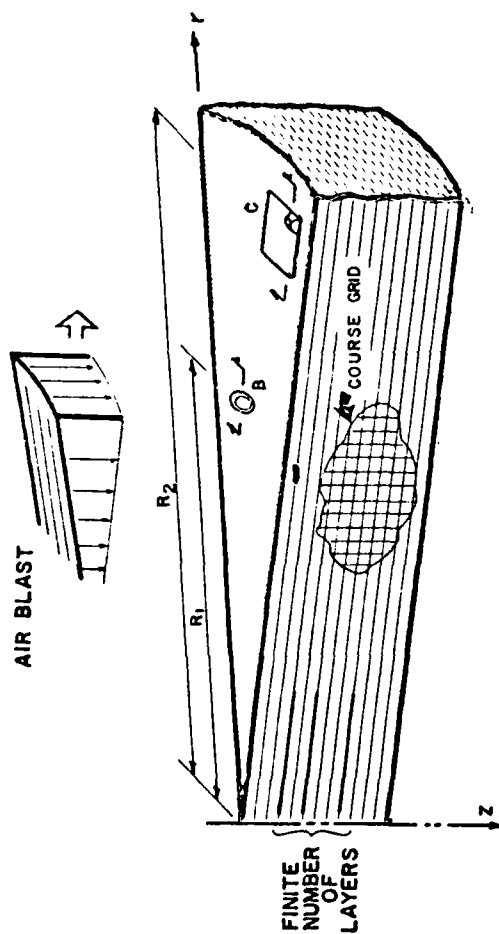
Since three-dimensional calculations can be quite complicated and tedious, an attempt has been made to include certain user-oriented features in TRANAL, such as free-format input, graphics capability, and consistency checks of the input data. Nonetheless it should be noted that TRANAL is not a general-purpose finite element program but was designed for solving large three-dimensional nonlinear wave propagation problems without using inordinate amounts of computer time. This has been achieved in the sense that it generally takes about the same amount of computer time to post-process and analyze the results as it takes to make a typical large-scale calculation (between one and ten hours on a CDC 7600).

SECTION 2

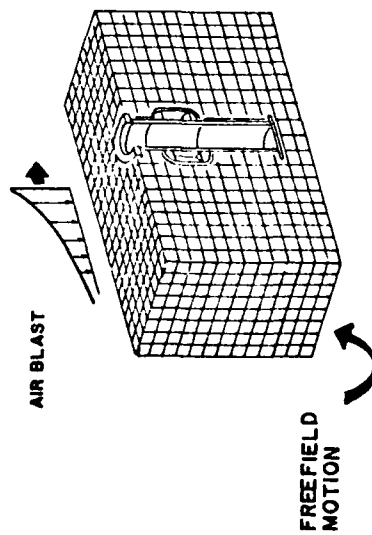
SOIL ISLAND METHOD

Since the number of elements increases rapidly as the mesh is refined in a three-dimensional structure-medium interaction (SMI) calculation, a technique called the soil island method is used to reduce the problem size. As illustrated in Fig. 2-1, ground motions are first generated by a free-field (no structure) calculation. For example, if the ground is modeled as a layered half-space, a two-dimensional axisymmetric code can be used to generate explosion-induced ground motions in the vicinity of the structure. These free-field motions can then be used as boundary values in detailed 3-D SMI calculations. Properly used, this technique extends the time domain of the simulation without an inordinate increase in the number of elements. Studies involving the soil island method can be found in Refs. 16 to 18. Figures 2-1 to 2-4 illustrate the use of this technique for a missile silo and a nuclear power plant containment structure. TRANAL can be used to do soil island calculations at the user's option, after the free-field motions have been generated (see Sections 6-7, 8-6.5 and Appendix C).

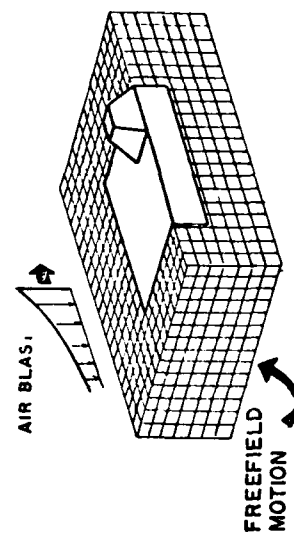
A refinement of this technique (which is currently under development but not yet available in TRANAL) is to perform a preliminary soil island calculation with a rigid inclusion or a coarse model in place of the structure. Motions and stresses are recorded at points on an imaginary surface just a few soil elements away from the structure and these are used in subsequent calculations as boundary values for a refined model of the structure. This two-stage soil island technique should result in improved results with a modest increase in computation time since it allows the number of soil elements to be maximized in the first stage and the number of structural elements to be maximized in the second stage calculation. Additional study is needed to determine appropriate procedures for nonlinear problems.



A. FREE FIELD CALCULATION (2-D AXISYMMETRIC)
(FINITE DIFFERENCE)



B. SOIL ISLAND (3-D) CONTAINING
BURIED SILO STRUCTURE
(FINITE ELEMENT)



C. SOIL ISLAND (3-D) CONTAINING
PARTIALLY BURIED RADAR STRUCTURE
(FINITE ELEMENT)

FIG. 2-1 SOIL ISLAND CONCEPT

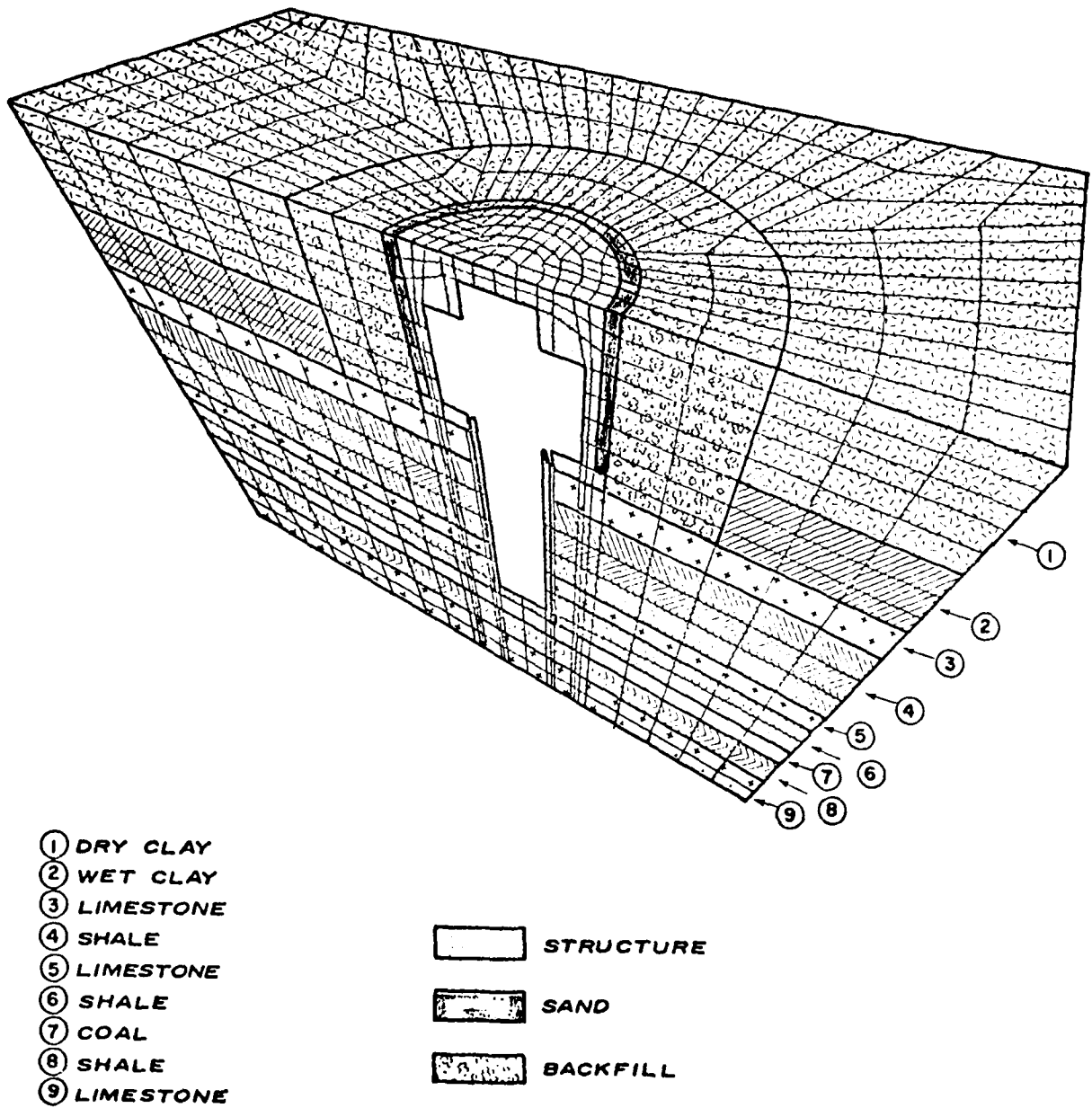


FIG. 2-2 PERSPECTIVE VIEW OF SOIL ISLAND MODEL
 USED FOR BURIED SILO

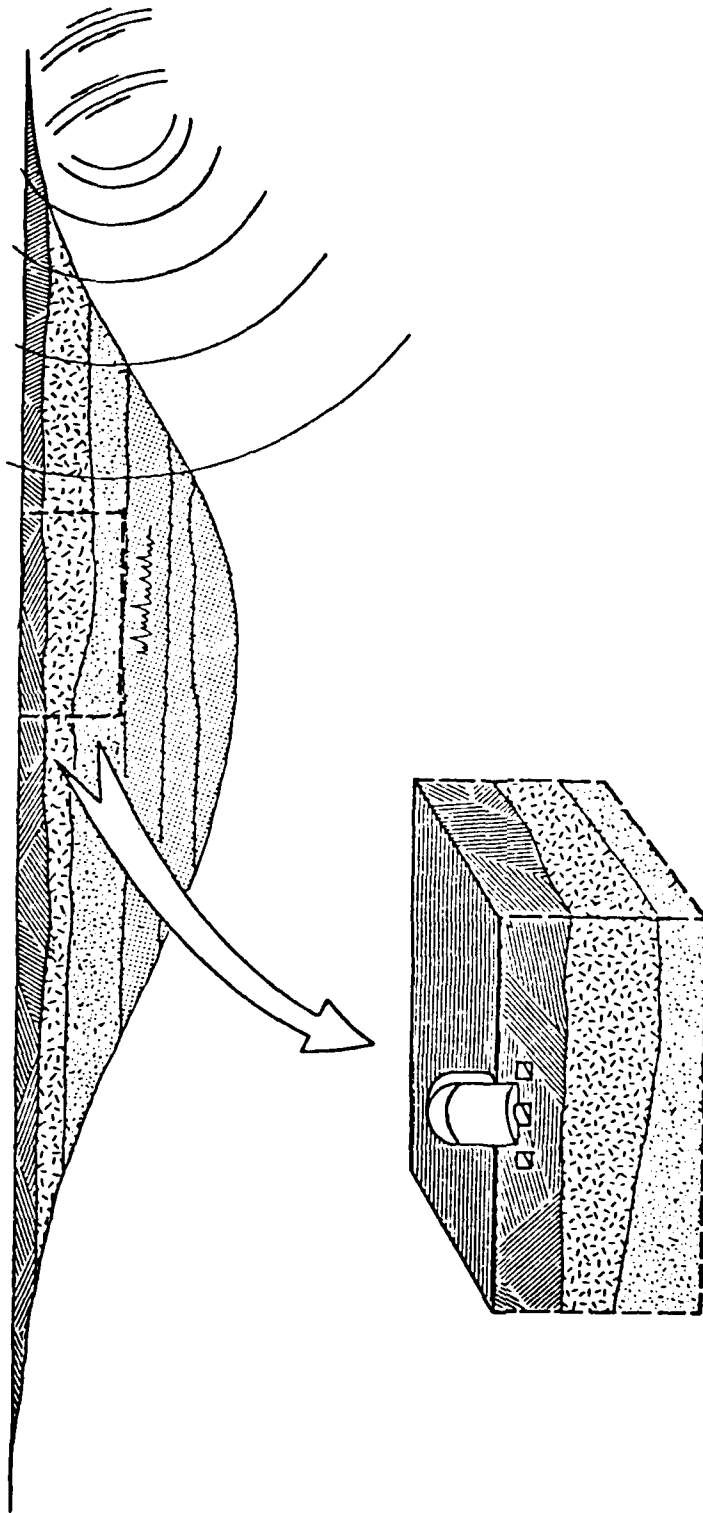


FIG. 2-3 SOIL ISLAND APPROACH APPLIED TO POWER
PLANT SUBJECTED TO SEISMIC GROUND
SHAKING

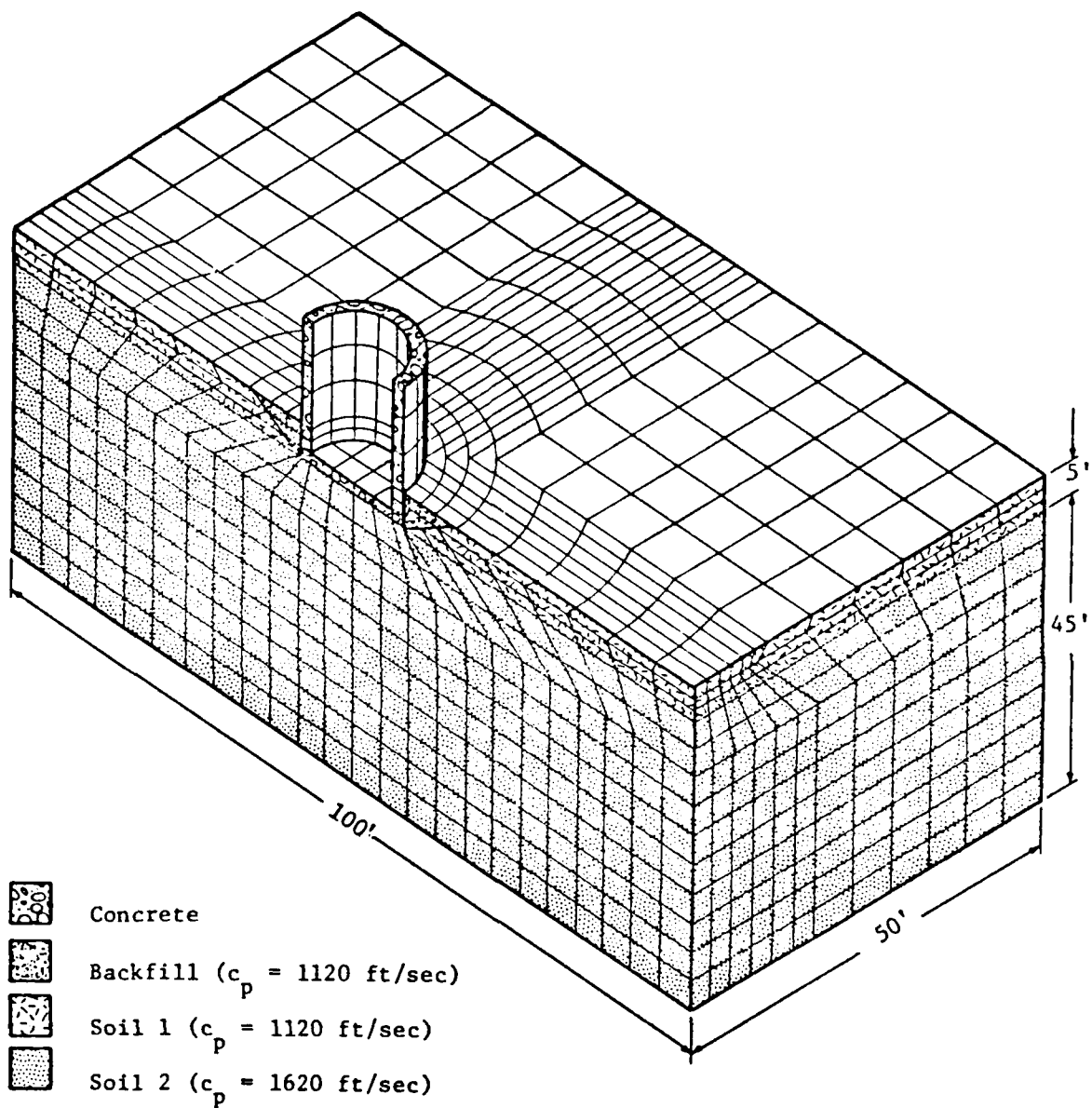


FIG. 2-4 SOIL ISLAND MODEL USED FOR SIMQUAKE
2 PRETEST CALCULATION OF 1/8 SCALE
MODEL RESPONSE

SECTION 3

METHOD OF ANALYSIS

This section presents the theoretical background on which the small strain, small displacement version of TRANAL is based. Starting from the principle of virtual work, the equations of motion are put in discrete form by means of the finite element method. Next a description of the cap model for representing material behavior is given, followed by a summary of the eight-noded isoparametric solid element. The remainder of this section is devoted to numerical solution procedures, including the explicit time integration scheme, its numerical stability, and a discussion of "subcycling" - a technique which permits different size time steps to be used in different elements.

3-1 Equations of Motion

The general equations of motion of a continuum are assumed in the form of the Principle of Virtual Work

$$\int_V [\rho \ddot{\underline{u}}^T \delta \underline{u} + \underline{\sigma}^T \delta \underline{\epsilon} - \underline{b}^T \delta \underline{u}] dV - \int_A \underline{t}^T \delta \underline{u} dA = 0 \quad (3-1)$$

where

- $\underline{u}^T = (u, v, w) = \text{displacement vector,}$
- $\underline{t}^T = (t_x, t_y, t_z) = \text{external traction vector,}$
- $\underline{\sigma}^T = (\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{yz}, \sigma_{zx}) = \text{stress (positive in tension)}$
- $\underline{\epsilon}^T = (\epsilon_{xx}, \epsilon_{yy}, \epsilon_{zz}, \epsilon_{xy}, \epsilon_{yz}, \epsilon_{zx}) = \text{strain (positive in extension)}$
- $\underline{b}^T = (b_x, b_y, b_z) = \text{body force (gravity),}$
- $\rho = \text{mass density,}$
- $\delta \underline{u} = \text{virtual displacement,}$
- $\delta \underline{\epsilon} = \text{virtual strain corresponding to } \delta \underline{u},$
- $(\dot{}) = \text{time derivative of } ().$

The equations of motion are transformed into a system of ordinary differential equations by the Finite Element Method as follows. The volume V is subdivided into n elements, V_k , $k = 1, 2, \dots, n$. Within each element the displacement field is represented as

$$\underline{u} = \underline{N}_k \underline{q}_k \quad (3-2)$$

where

- $\underline{N}_k = \underline{N}_k(x, y, z) = \text{a known matrix of interpolation functions,}$
- $\underline{q}_k = \text{nodal displacement vector of the } k\text{-th element.}$

The strain-displacement relations

$$\begin{aligned}\varepsilon_{xx} &= \frac{\partial u}{\partial x}, \dots \\ 2\varepsilon_{xy} &= \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}, \dots\end{aligned}\quad (3-3)$$

can be combined with Eq. (3-2) to produce the relation

$$\tilde{\varepsilon} = \tilde{B}_k \tilde{q}_k \quad (3-4)$$

Similarly for the virtual displacement and the virtual strain we have

$$\delta \tilde{u} = \tilde{N}_k \delta \tilde{q}_k \quad (3-5a)$$

$$\delta \tilde{\varepsilon} = \tilde{B}_k \delta \tilde{q}_k \quad (3-5b)$$

Since the Principle of Virtual Work holds for arbitrary $\delta \tilde{q}_k$, the discrete form of the equations of motion can be written in the form

$$\tilde{M} \ddot{\tilde{q}} + \tilde{Q} = \tilde{P} \quad (3-6)$$

where

$$\tilde{M} = \sum_k \tilde{M}_k = \sum_k \int_{V_k} \tilde{N}_k^T \rho \tilde{N}_k dV \quad (3-7)$$

$$\tilde{Q} = \sum_k \tilde{Q}_k = \sum_k \int_{V_k} \tilde{B}_k^T \sigma dV \quad (3-8)$$

$$\tilde{P} = \sum_k \tilde{P}_k = \sum_k \left(\int_{A_k} \tilde{N}_k^T t dA + \int_{V_k} \tilde{B}_k^T b dV \right) \quad (3-9)$$

In the preceding and throughout this report, standard continuum mechanics conventions are used; that is, strains are positive in extension and stresses are positive in tension.

3-2 Constitutive Models

Material behavior for a continuum can be represented by any reasonable constitutive (stress-strain) relations, and TRANAL is implemented in a modular fashion to facilitate the introduction of new material model subroutines. The cap model routine, CAP75, Ref. 7, is available in TRANAL and can be used to represent various types of materials (soils, rocks, concrete, etc.).

The cap model is based on classical plasticity theory with incremental stress-strain relations in the form

$$\dot{\tilde{\sigma}} = \tilde{D} \dot{\tilde{\varepsilon}} \quad (3-10)$$

where $\tilde{D}$ is the (tangent) modulus matrix which for mathematical reasons (uniqueness, stability, continuity) is assumed to be positive-definite. The cap model is defined by a convex yield surface, $Y(\sigma)$, and a plastic strain rate vector, $\dot{\tilde{\varepsilon}}^p$, which is normal to the yield surface in stress space so that

$$\dot{\epsilon}_{ij}^p = \Lambda \frac{\partial Y}{\partial \sigma_{ij}} \quad (3-11)$$

where Λ is a non-negative scalar function which must be determined as part of the solution of the problem.

The CAP75 model is based on eight material parameters for representing tensile behavior. Two parameters, K and G , are used to represent simple linear elastic behavior inside the yield surface with K being the bulk modulus of the material and G being its shear modulus. The yield surface is defined by a (fixed) failure envelope and a (moveable) hardening cap, Figs. 3-1 through 3-5. The failure envelope is defined by

$$Y(\sigma) = \sqrt{J_2'} - F_F(J_1) \quad (3-12)$$

$$F_F(J_1) = A - C \exp(BJ_1) \quad (3-13)$$

where A , B , C are material parameters,

$$J_1 = \sigma_{xx} + \sigma_{yy} + \sigma_{zz} \quad (3-14)$$

is the first invariant (trace) of the stress tensor, and

$$J_2' = \frac{1}{6}[(\sigma_{xx} - \sigma_{yy})^2 + (\sigma_{yy} - \sigma_{zz})^2 + (\sigma_{zz} - \sigma_{xx})^2 + \sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{zx}^2] \quad (3-15)$$

is the second invariant of the deviatoric stress tensor. The cap is defined by

$$Y(\sigma) = \sqrt{J_2'} - F_c(J_1, \kappa) \quad \text{for } L(\kappa) \geq J_1 \geq X(\kappa) \quad (3-16)$$

where κ is an internal state variable that measures hardening as a functional of the history of plastic volumetric strain, and $L(\kappa)$, $X(\kappa)$ define the J_1 range of the cap. The function $F_c(J_1, \kappa)$ is defined by

$$F_c(J_1, \kappa) = \frac{1}{R} \sqrt{[X(\kappa) - L(\kappa)]^2 - [J_1 - L(\kappa)]^2} \quad (3-17)$$

in which

$$L(\kappa) = \begin{cases} \kappa & \text{if } \kappa < 0 \\ 0 & \text{if } \kappa \geq 0 \end{cases} \quad (3-18)$$

$$X(\kappa) = \kappa - RF_F(\kappa) \quad (3-19)$$

where R is a material parameter.

The hardening parameter κ is implicitly defined as a functional of the plastic volumetric strain, ϵ_v^p , by means of a relation between $X(\kappa)$ and ϵ_v^p which is then coupled with Eq. (3-19) to define κ in terms of ϵ_v^p . For soils (McCormick Ranch Sand, Ref. 6) the relationship is

$$\epsilon_v^p = W[e^{DX(\kappa)} - 1] \quad (3-20)$$

in which W and D are material parameters and ϵ_v^p is a history dependent functional of the volumetric plastic strain and is given by the differential functional relation

$$\dot{\epsilon}_v^p = \begin{cases} \dot{\epsilon}_v^p & \text{if } \dot{\epsilon}_v^p \leq 0 \text{ or } \kappa < 0 \\ 0 & \text{if } \dot{\epsilon}_v^p > 0 \text{ and } \kappa \geq 0 \end{cases} \quad (3-21)$$

In Ref. 6, $\bar{\epsilon}_v^p$ could be taken as equal to ϵ_v^p because only compressive stresses were allowed; in that case, $\dot{\epsilon}_v^p$ can be positive only if $\kappa < 0$, i.e., dilatancy can occur only on the failure envelope, which can be reached only if L and hence, by Eq. (3-18), κ is negative.

If dilatancy occurs when $J_1 > 0$, Eq. (3-21) limits the shrinking of the cap to $\kappa = L(\kappa) = 0$. Such a limitation is somewhat arbitrary in view of the lack of data with respect to material behavior after the occurrence of tension failure; this ensures that the cap remains finite and Eq. (3-21) is assumed to apply in general for soils.

The tension cutoff procedure may take various forms. For simplicity, the following procedure is used. The tension cutoff criterion is chosen as

$$\sigma_1 = T \quad (3-22)$$

in which T (a material parameter) represents a limit on the maximum allowable principle tensile stress, σ_1 .

The soil cap model described by the above equations may be simply modified to represent rock behavior. In order to do this, Eq. (3-21), which limits the amount of dilatancy achievable in soils by shrinking the cap, is modified to

$$\dot{\epsilon}_v^p = \min(\dot{\epsilon}_v^p, 0) \quad (3-23)$$

This prevents the cap from shrinking and therefore allows large amounts of dilatancy as observed in rocks. Although a separate cap model has been developed for rocks, Ref. 19, it appears that Eq. (3-23) yields an adequate model for most rocks and has been used in most calculations. In addition, Eqs. (3-21) and (3-23) allow both rocks and soils to be handled by a single cap model routine.

Other generalizations of the cap model to include rate dependence and anisotropy have been made, Refs. 20 to 22, but these are not included in the present version of TRANAL.

3-3 Three Dimensional Solid Element

The three dimensional solid element is based on the standard eight-noded isoparametric element which uses a tri-linear interpolation function to represent the displacement field (Fig. 3-6). This is the most computationally efficient element which is adequate to describe the class of problems of interest and may give better results than the next higher order element, Ref. 23.

Let x, y, z be the global coordinates with

$$\begin{Bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{Bmatrix}^T = (x_1, \dots, x_8, y_1, \dots, y_8, z_1, \dots, z_8) \quad (3-24)$$

being the coordinates of the nodal points.

The following coordinate transformation will be used

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \begin{bmatrix} \bar{p} & 0 & 0 \\ 0 & \bar{p} & 0 \\ 0 & 0 & \bar{p} \end{bmatrix} \begin{Bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{Bmatrix} \quad (3-25)$$

where

$$\bar{p} = (p_1, p_2, \dots, p_8) \quad (3-26)$$

and

$$p_i = \frac{1}{8} (1 + \xi \xi_i)(1 + \eta \eta_i)(1 + \zeta \zeta_i) \quad (3-27)$$

The coordinates ξ_i, η_i, ζ_i of the i -th mode ($i = 1, 2, \dots, 8$) are shown below.

i	1	2	3	4	5	6	7	8
ξ_i	-1	-1	-1	-1	+1	+1	+1	+1
η_i	-1	+1	+1	-1	-1	+1	+1	-1
ζ_i	-1	-1	+1	+1	-1	-1	+1	+1

In the system ξ, η, ζ , the element walls are $\xi = \pm 1, \eta = \pm 1, \zeta = \pm 1$.

The displacement field within the element is assumed as

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \begin{bmatrix} \bar{p} & 0 & 0 \\ 0 & \bar{p} & 0 \\ 0 & 0 & \bar{p} \end{bmatrix} \begin{Bmatrix} \bar{u} \\ \bar{v} \\ \bar{w} \end{Bmatrix} \quad (3-28)$$

where the nodal displacement matrix (q_k in the notation of Sec. 3-1) is

$$\bar{q}_k^T = \begin{Bmatrix} \bar{u} \\ \bar{v} \\ \bar{w} \end{Bmatrix}^T = (u_1, \dots, u_8, v_1, \dots, v_8, w_1, \dots, w_8) \quad (3-29)$$

while the matrix N_k (as defined by Eq. 3-2) can be easily identified in Eq. (3-28) above.

The strain components are

$$\tilde{e} = \begin{Bmatrix} e_{xx} \\ e_{yy} \\ e_{zz} \\ 2e_{xy} \\ 2e_{yz} \\ 2e_{zx} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \\ \frac{\partial w}{\partial x} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \\ \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \end{Bmatrix} = \tilde{B} \begin{Bmatrix} \bar{u} \\ \bar{v} \\ \bar{w} \end{Bmatrix} \quad (3-30)$$

where

$$\tilde{B} = \begin{bmatrix} \frac{\partial p}{\partial x} & 0 & 0 \\ 0 & \frac{\partial p}{\partial y} & 0 \\ 0 & 0 & \frac{\partial p}{\partial z} \\ \frac{\partial p}{\partial y} & \frac{\partial p}{\partial x} & 0 \\ 0 & \frac{\partial p}{\partial z} & \frac{\partial p}{\partial y} \\ \frac{\partial p}{\partial z} & 0 & \frac{\partial p}{\partial x} \end{bmatrix} \quad (3-31)$$

To determine the elements of $\tilde{B}$ the jacobian matrix $\tilde{J}$ is needed

$$\tilde{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} = \begin{Bmatrix} \frac{\partial p}{\partial \xi} \\ \frac{\partial p}{\partial \eta} \\ \frac{\partial p}{\partial \zeta} \end{Bmatrix} [\{\bar{x}\}, \{\bar{y}\}, \{\bar{z}\}] \quad (3-32)$$

Since

$$\begin{Bmatrix} \frac{\partial p}{\partial \xi} \\ \frac{\partial p}{\partial \eta} \\ \frac{\partial p}{\partial \zeta} \end{Bmatrix} = \tilde{J} \begin{Bmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \\ \frac{\partial p}{\partial z} \end{Bmatrix} \quad (3-33)$$

the necessary partial derivatives with respect to x , y , and z in Eq. (3-31) follow as

$$\begin{Bmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \\ \frac{\partial p}{\partial z} \end{Bmatrix} = \tilde{J}^{-1} \begin{Bmatrix} \frac{\partial p}{\partial \xi} \\ \frac{\partial p}{\partial \eta} \\ \frac{\partial p}{\partial \zeta} \end{Bmatrix} \quad (3-34)$$

3-4 Integration Scheme

The time integration algorithm in TRANAL is an explicit method based on the central different operator

$$\ddot{f}^n = \frac{1}{\Delta t^2} (f^{n+1} - 2f^n + f^{n-1}) \quad (3-35)$$

The superscripts here indicate time; for example f^n corresponds to $f(t_n)$ where $t_n = n\Delta t$. Numerical stability considerations for this method are discussed in Section 3-6, and "subcycling" is discussed in Section 3-7.

In order to use this method efficiently, the "consistent" mass matrices in Eq. (3-7) are replaced by diagonal, "lumped" mass matrices defined by

$$M_{\sim k} = \frac{1}{8} \rho V_k I \quad (3-36)$$

where ρ is the mass density, V_k is the volume of the k -th element, I is the identity matrix, and the factor $(1/8)$ appears because the mass of the element is distributed equally to the eight nodes of the element. It is now simple to invert the diagonal mass matrix and solve for the nodal accelerations.

The basic time integration scheme for the small strain, small displacement version of TRANAL is as follows:

1. $\dot{q}^n, \sigma^{n-1/2}$ are assumed to be known.
2. $\dot{\epsilon}^n = B \dot{q}^n$; element strain rates.
3. $\dot{\sigma}^n = D \dot{\epsilon}^n$; element stress rates, from model routine.
4. $\sigma^{n+1/2} = \sigma^{n-1/2} + \Delta t \dot{\sigma}^n$; element stresses.
5. $Q^{n+1/2} = \int_{\sim} B^T \sigma^{n+1/2} dV_k$; nodal reactions.
6. $P^{n+1/2}$; prescribed nodal forces.
7. $\ddot{q}^{n+1/2} = M^{-1} (P^{n+1/2} - Q^{n+1/2})$; nodal accelerations.
8. $\dot{q}^{n+1} = \dot{q}^n + \Delta t \ddot{q}^{n+1/2}$; nodal velocities.
9. Kinematic conditions: e.g., prescribed displacements.

The nodal reactions in step 5 are determined by a one-point integration scheme

$$Q_{\sim} = \int B^T \sigma dV_k \quad (3-37)$$

$$= [B^T \sigma]_{\sim} V_k \quad (3-38)$$

(ξ, η, ζ) = (0,0,0)

This results in an approximation which is much like the classic finite difference equations for continuum mechanics. In fact, for rectangular elements it is equivalent to the staggered grid method which was first suggested by von Neumann and is now widely used in many finite difference codes. A theoretical drawback of this scheme is that certain strain patterns with no reaction forces may develop throughout the grid. This effect, sometimes referred to as "hour-glass" or "keystone" instability, has a long

history and has been dealt with by means of tensor viscosities, higher order integration, incompatible modes, etc. In a practical sense, this effect seems to be unimportant for most applications since experience indicates that the "unresisted strains" usually do not develop if physically reasonable initial and boundary values are used.

The primary reason for choosing an explicit method over an implicit method is that, for nonlinear 3-D continuum calculations, the number of computer operations is generally smaller, especially for a large number of elements. Consider a calculation which must span a finite time duration T for a three dimensional unit cube. Assume that the unit cube, which is homogeneous with wave speed c , is subdivided into m elements in each direction so that each element is a cube with edge length $d = 1/m$. For the explicit method the time step Δt is restricted by a numerical stability condition (see Sec. 3-6) of the form $c\Delta t \leq \alpha d$ where α is of the order of one. The number of time steps is proportional to $T/\Delta t$ which is proportional to m . Since the total number of elements is m^3 , the number of computer operations increases as m^4 for large m . Assume that the implicit method can cover the time T in one step, regardless of the number of elements. Based on the Gaussian elimination method, the number of computer operations required to solve the implicit equations is proportional to eb^2 where e is the number of equations and b is the bandwidth. For the unit cube, e is m^3 and b is proportional to m^2 . Hence the number of computer operations for an implicit method increases as m^7 for large m . Other considerations, which also favor the explicit method for wave propagation calculations, are ease of implementation and computer storage.

3-5 Pseudo-Viscosity

If a material has strong hardening behavior, discontinuities (i.e. shocks) can develop in the solution of wave propagation problems. Such discontinuities are inconsistent with certain assumptions underlying the derivation of the discrete form of the equations, and severe oscillatory motions appear in the numerical solution. (Although boundary and initial conditions contribute to shock formation, material hardening has a particularly strong effect on the amplification of numerical oscillations.) A three dimensional generalization of the classical pseudo-viscosity method of von Neumann and Richtmyer, Ref. 2, is provided to control these oscillations. The form of the viscosity is similar to that used in many finite difference codes (e.g. Ref. 24). Both linear and quadratic viscosities are available in TRANAL and the coefficients can be selected independently for each material type. Since the numerical scheme is stable for sufficiently small Δt , the default values of the coefficients are zero.

For each element a scalar (isotropic) viscous pressure

$$p_v = -a_L \rho c_p d \dot{\epsilon}_v - a_Q \rho d^2 |\dot{\epsilon}_v| \dot{\epsilon}_v \quad (3-39)$$

is calculated, where ρ is the mass density, c_p is the dilatational wave speed ($c_p = \sqrt{(K + 4G/3)/\rho}$ for the cap model of Section 3-2), $\dot{\epsilon}_v = \dot{\epsilon}_{xx} + \dot{\epsilon}_{yy} + \dot{\epsilon}_{zz}$ is the volumetric part of the strain rate, and d is a length which is defined in the section on numerical stability (Sec. 3-6). The pseudo-viscosity alters the calculation of the nodal reaction forces in step 5 of Section 3-4 in the

following way

$$\underline{Q} = \int \underline{B}^T (\underline{\sigma} - p_v \underline{I}) dV_k \quad (3-40)$$

The coefficients a_L and a_Q are arbitrary positive numbers and are of the form

$$a_i = \left\{ \begin{array}{ll} a_i^c, & \dot{\epsilon}_v < 0 \quad (\text{compression}) \\ a_i^e, & \dot{\epsilon}_v \geq 0 \quad (\text{expansion}) \end{array} \right\} \quad (3-41)$$

for $i = L$ or Q . Various studies have been published and results show that quadratic viscosity is most important in shock regions whereas linear viscosity plays a role throughout the grid. Both coefficients should be kept as small as possible to avoid distorting the solution too much; this is particularly true of the linear viscosity since the quadratic type is much more localized in its effect. As noted in Ref. 2, pseudo-viscosity is often used in compression only; that is, $a_i^e = 0$. However, some authors prefer to use $a_i^e = 0$, but $a_i^e = a_i^c$. Typically, a_Q should be less than about 10, and a_L should be less than about 0.2. As the coefficients increase the time step must decrease to maintain numerical stability.

3-6 Numerical Stability

The essence of numerical stability is that numerical errors must not be allowed to grow so rapidly that the solution of interest is lost. These errors have many sources (machine round-off, approximations in space and time, etc.) and a complete mathematical theory does not exist for transient nonlinear calculations in general.

The most important aspects of stability can be understood by considering the equivalent, linearized, constant coefficient problem at any point in time. In this case, the equations of motion, Eq. (3-6), can be written in the form

$$\underline{M} \ddot{\underline{q}} + \underline{K} \underline{q} = \underline{P}(t) \quad (3-42)$$

where

$$\underline{K} = \int \underline{B}^T \underline{D} \underline{B} dV \quad (3-43)$$

is a positive-semidefinite stiffness matrix. The central difference method for Eq. (3-42) can be written as

$$\underline{q}^{n+1} - 2\underline{q}^n + \underline{q}^{n-1} = \Delta t^2 \underline{M}^{-1} [\underline{P}^n - \underline{K} \underline{q}^n] \quad (3-44)$$

The numerical stability of this method can be studied by using modal analysis. Hence Eq. (3-44) becomes

$$\underline{y}^{n+1} - 2\underline{y}^n + \underline{y}^{n-1} = \Delta t^2 (\underline{p}^n - \omega^2 \underline{y}^n) \quad (3-45)$$

where ω^2 is an eigenvalue of the free vibration problem

$$(\underline{K} - \omega^2 \underline{M}) \underline{y} = 0 \quad (3-46)$$

and y is the corresponding eigenvector. The equations are now uncoupled completely and the numerical error propagation equation associated with Eq. (3-45) is

$$e^{n+1} - (2 - a^2)e^n + e^{n-1} = 0 \quad (3-47)$$

where $a = \omega \Delta t$, and e^n is the difference between the computed solution and the true solution. Numerical errors are properly bounded if the solution of the constant coefficient difference Eq. (3-47) satisfies the condition $|e^n| \leq 1$; this yields the condition $a \leq 2$ or

$$\omega \Delta t \leq 2 \quad (3-48)$$

which is the well-known stability condition of the central difference method for the undamped linear oscillator. Hence if the maximum system frequency, $\omega_{\max}$, is known, the time step is known for stability.

There are many methods for estimating bounds on eigenvalues (Gerschgorin's circle theorem, the power method, the Rayleigh quotient, etc.). As shown in Ref. 25, various results can be derived from the Rayleigh quotient,

$$R = U/T \quad (3-49)$$

where U and T are the "reduced" potential and kinetic energies defined by

$$U = \frac{1}{2} \tilde{u}^T \tilde{K} \tilde{u} \quad (3-50a)$$

and

$$T = \frac{1}{2} \tilde{u}^T \tilde{M} \tilde{u} \quad (3-50b)$$

where $\tilde{u}$ is an arbitrary vector. According to Rayleigh's principle, the Rayleigh quotient lies between the minimum and maximum eigenvalues of Eq. (3-46) so that

$$\omega_{\min}^2 \leq R(\tilde{u}) \leq \omega_{\max}^2 \quad (3-51)$$

Furthermore, linear constraints of the form $\tilde{c}^T \tilde{u} = 0$ are included in these results where $\tilde{c}$ is a vector of constants. The introduction of such constraints can never increase the highest system frequency above $\omega_{\max}$ (or decrease the lowest frequency below $\omega_{\min}$). Since the joining of two or more finite elements corresponds to such a constraint, these results are directly applicable to the results at hand. In particular, it follows from Rayleigh's principle that the maximum frequency of an assemblage of finite elements is less than or equal to the maximum frequency of the individual unconstrained elements of which the assemblage is composed. Hence after finding the maximum frequency of all unconstrained elements in an assemblage, Eq. (3-48) can be used to determine a permissible time step.

Since the element assembly process corresponds to constraint conditions which, according to Rayleigh's principle can never raise the maximum frequency above that of the unassembled elements, the condition determined in this way

is sufficient for stability. Whether the assembled system has a lower maximum frequency than that of the individual elements depends on the details of the finite element configuration. A few cases are considered below.

It should be noted that these ideas can be extended to include pseudo-viscosity as well as non-continuum (structural) elements. This is not surprising since energy methods, Refs. 2, 26 are often used in the analysis of numerical stability; the approach here is a simplified kind of energy method based on Rayleigh's principle.

As a simple example, consider a two-noded, constant-strain bar element with Young's modulus E , area A , density ρ , and length d . The strain or potential energy of the element is $U = k(u_2 - u_1)^2/2$, where u_1 and u_2 are the displacements of the two nodes and $k = EA/d$. The element mass, $m = \rho Ad$, is distributed equally between the two nodes, $m_1 = m_2 = m/2$, so that the Rayleigh quotient $R = k(u_2 - u_1)^2/(m_1 u_1^2 + m_2 u_2^2)$. The maximum value of R is obtained for $u_1 = -u_2$; thus the maximum frequency of the element is $\omega_{\max} = 2\sqrt{k/m} = 2c/d$, where $c = \sqrt{E/\rho}$ is the bar wave speed. Equation (3-48) then yields the restriction $c\Delta t/d \leq 1$, which is the classical "CFL" stability condition derived by Courant, Friedrichs and Lewy, Ref. 2. In this case the maximum element frequency is the same as the maximum frequency of assembled elements (assuming identical elements and ignoring end conditions).

Next consider an element in the shape of a cube with edge length d and having eight corner nodes. If the material is a simple fluid with bulk modulus K , the strain energy is given by $U = K\theta^2 d^3/2$, where $\theta = u_x + v_y + w_z$ is the dilatation, whose difference approximation for this element can be found by specializing the results in Section 3-3. The element mass, $m = \rho d^3$, is distributed equally to the eight nodes, so that $T = \sum m(u_i^2 + v_i^2 + w_i^2)/16$. The Rayleigh quotient is maximum for the purely dilatational mode of the element and it can be shown that $R = 12Kd^2/\rho$, which corresponds to $c\Delta t \leq d/\sqrt{3}$, where $c = \sqrt{K/\rho}$ is the fluid wave speed. This result is contained in Ref. 27, p. 382.

Now consider eight of these fluid elements arranged to form a cube of two elements on each side. In this arrangement it is impossible for all eight elements to be in pure dilatation simultaneously. Consequently the highest frequency of this assemblage is never as large as that of a single element. The stability condition in this situation is $c\Delta t/d \leq 1$, which is identical to the CFL condition for the one dimensional bar. Since an infinite grid of elements can be made up of these $2 \times 2 \times 2$ arrangements, the stability condition for a large assemblage of cubes should be $c\Delta t \leq \alpha d$, where α is close to one.

A similar analysis can be made for a cube made of linear elastic solid with bulk modulus K and shear modulus G . The stability condition for a single element is $c\Delta t \leq d/\sqrt{3}$, where $c = \sqrt{K/\rho}$; that is, the shear modulus has no effect in this case. For a $2 \times 2 \times 2$ arrangement of these cubes the result is $c_p \Delta t \leq d$, where $c_p = \sqrt{(K + 4G/3)/\rho}$ is the dilatational wave speed.

Although it is possible to calculate and use the highest unconstrained element frequency to determine stability, it appears that the resultant time step would be too small by a factor of about 1.5 when compared to a complete finite element assemblage. Other factors, such as variations in the size and shape of the elements can also affect the highest system frequency. For simplicity and speed of calculation, the following procedure is used in TRANAL. A length d equal to the smallest distance between the parallel planes of an equivalent parallelepiped is

$$\Delta t = \alpha \frac{d}{c_p} \quad (3-52)$$

is used in the calculation, where α is an input parameter called the stability factor; the default value of α is 0.8.

The pseudo-viscosity modifies the nodal reaction forces in a way which increases the element frequency and hence reduces the time step for numerical stability. The procedure used in TRANAL is based on the stability condition suggested in Ref. 24. Equation (3-52) is replaced by

$$\Delta t = \left(\frac{\alpha}{\psi + \sqrt{1 + \psi^2}} \right) \frac{d}{c_p} \quad (3-53)$$

where

$$\psi = a_L + a_Q |\dot{\epsilon}_v| d / c_p \quad (3-54)$$

As a general rule, ψ should be much less than one.

In order to simplify certain details of implementation (associated with subcycling) an estimate of

$$\Delta v = \max(\dot{\epsilon}_v d) \quad (3-55)$$

is required as an input value whenever $a_Q \neq 0$. In most problems with hardening materials it is usually easy to estimate this value of the particle velocity from the shock condition

$$\Delta v = \frac{\Delta p}{\rho c} \quad (3-56)$$

where Δp is the maximum expected jump in pressure at the shock front, ρ is the material density and c is the wave speed corresponding to the secant modulus to the shock pressure Δp . Since only an estimate of Δv is needed, the material behavior in either hydrostatic or uniaxial compression can be used. An estimate which is much too low may result in too large a time step and, hence, a numerically unstable calculation. An estimate which is much too high will result in too small a time step and, hence, a waste of computer time.

3-7 Subcycling

In a typical calculation, the elements vary in size and material type so that the time step required for stability is smaller in some elements than in others. In some situations, time step ratios greater than ten can occur. Since Rayleigh's principle indicates that numerical stability is a local (element) property, it is reasonable to try to use different size time steps in different elements. A scheme for doing this has been implemented in TRANAL and used successfully in many calculations. The use of this scheme, called "subcycling", can reduce the number of computer operations, sometimes by more than a factor of ten.

The time integration algorithm of Section 3-4 has been modified to permit groups of elements, called zones, to be integrated at a different rate from elements in a neighboring zone. For convenience in implementation, it is required that the ratio of time step sizes in neighboring zones must be an integer. Two zones are called neighboring if they have at least one node in common.

All zones are assumed to begin at the same point in time. The calculation proceeds by taking a single time step in all zones with the largest

(identical) time step. All zones with the next largest time step are integrated forward one time step. This process is continued until those zones with the smallest time step are encountered. All zones at this level are integrated the required number of time steps until they reach the nearest time point of all their neighbors. This process is repeated at each level until all zones have reached the same point in time. The entire process, called a cycle (or sometimes called a major cycle or major time step) can now be repeated.

Nodes which are common to two or more zones are called interface nodes. Reaction forces (Q's) at interface nodes are calculated incrementally and the entire force increment from an element in a zone is added during a time step. Velocities at interface nodes are calculated with the last possible zone with the smallest possible time step. All other quantities (stresses and non-interface velocities) are handled as implied by the scheme outlined in Section 3-4.

As an aid in estimating computer time, the number of elements processed in a cycle (major time step) is printed. Calculation time on the CDC 7600 is less than 0.5 milliseconds per element per cycle.

Since distorted elements can significantly reduce the time step, several options are provided for printing information about specific elements in each zone. A "time increment" print limit, N, can be used to obtain element data about the N elements (in each zone) which give the smallest Δt according to Eq. (3-53). Similarly an "angle" print limit can be used to obtain data about those elements which have certain angles less than 60 degrees. Also a "negative volume" print limit can be used to obtain information about elements with negative volume. By altering nodal coordinates (see Sec. 8-3) it is usually possible to increase Δt , reduce element distortion, or eliminate negative volumes. The default value for each print limit is 10.

Various forms of subcycling have been suggested (e.g. Ref. 2, pages 268-269). Subcycling falls within the general framework of operator splitting methods and is one of many possible "mixed time integration schemes", Ref. 8.

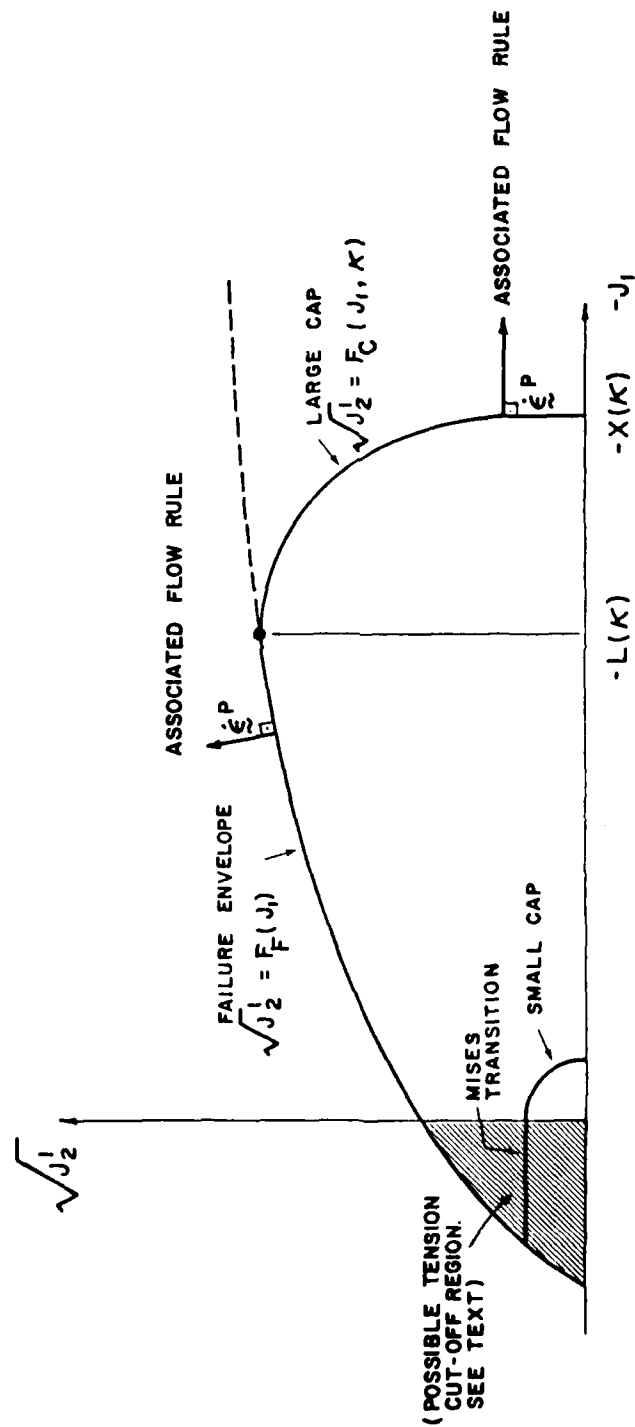
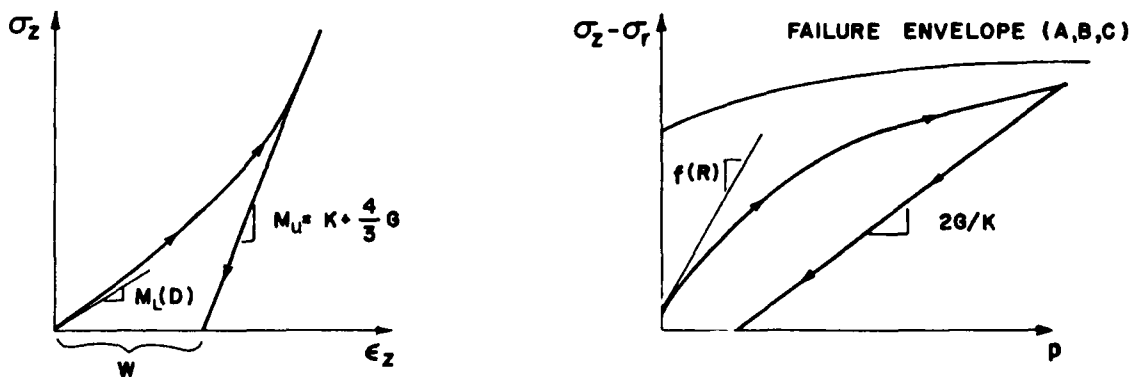


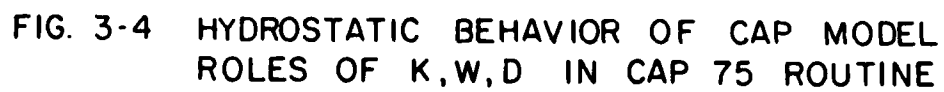
FIG. 3-1 TYPICAL YIELD SURFACES IN CAP MODEL

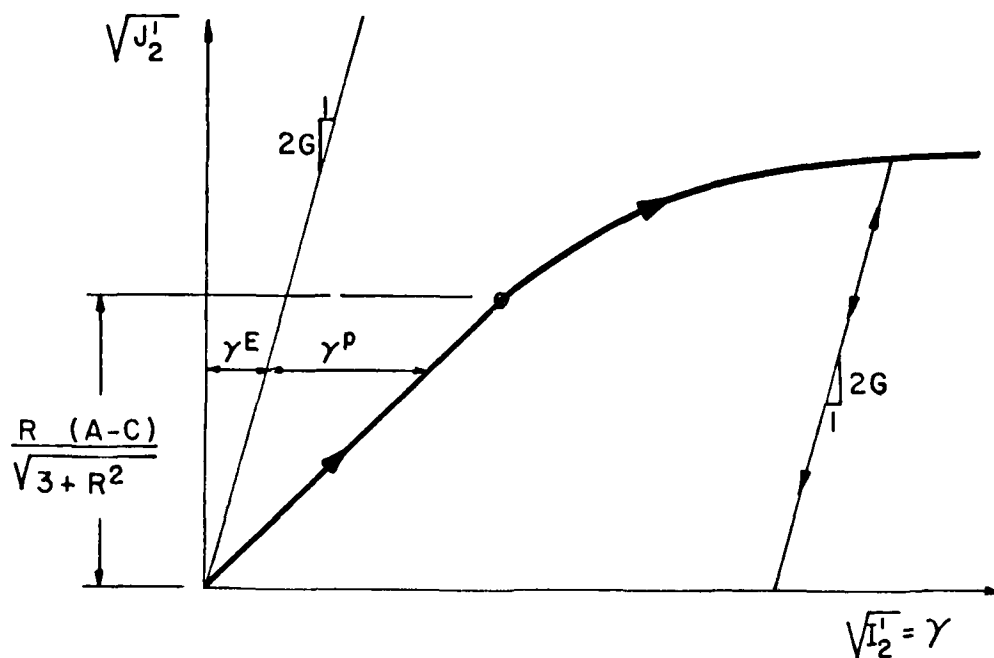
UNIAXIAL STRAIN TEST



PARAMETER	ROLE IN MODEL	EFFECT ON BEHAVIOR
K	ELASTIC BULK MODULUS	CONTROLS VOLUMETRIC BEHAVIOR DURING UNLOADING
G	ELASTIC SHEAR MODULUS	CONTROLS SHEAR BEHAVIOR DURING UNLOADING
A	MISES FAILURE LIMIT	MAXIMUM SHEAR STRESS WHICH CAN BE SUPPORTED BY MATERIAL
B	RATE AT WHICH FAILURE ENVELOPE APPROCHES MAX.	CONTROLS ANGLE OF FRICTION LOW PRESSURES
C	SHEAR STRENGTH AT ZERO PRESSURE	COHESION
W	MAXIMUM PLASTIC VOLUMETRIC STRAIN	LOCKING STRAIN - UNFILLED VOIDS
D	RATE AT WHICH COMPACTION OCCURS WITH PRESSURE	CONTROLS INITIAL LOADING MODULI
R	SHAPE FACTOR FOR CAP	CONTROLS LOADING STRESS PATH IN UNIAXIAL STRAIN

FIG. 3-2 ROLE OF PARAMETERS IN CAP MODEL (CAP 75)





PLASTIC POISSON'S RATIO IN UNCONFINED COMPRESSION IS:

$$\nu^P = -\frac{\dot{\epsilon}_3^P}{\dot{\epsilon}_1^P} = \frac{R^2 - 6}{6 + 2R^2} \quad \text{FOR } \sigma_1 \leq \frac{R(A-C)}{\sqrt{1 + R^2/3}}$$

FIG. 3-5 UNCONFINED COMPRESSION BEHAVIOR OF SIMPLE CAP MODEL ROLES OF G,R IN CAP 75 ROUTINE

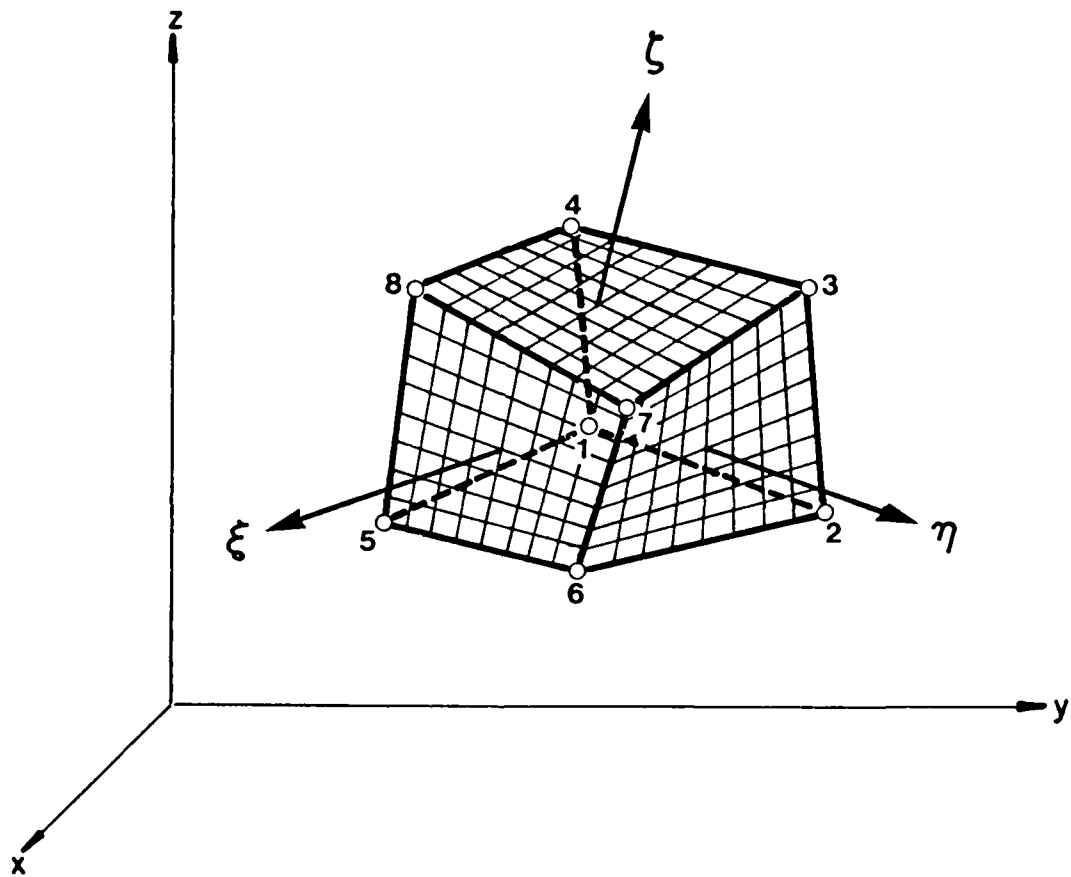


FIG. 3-6 EIGHT-NODED ISOPARAMETRIC SOLID ELEMENT

SECTION 4

FINITE ELEMENT REPRESENTATION OF SOIL ISLAND AND STRUCTURE

4-1 Coordinate System and Element Subdivision

The basic TRANAL coordinate system is Cartesian with global coordinates denoted by x, y, z . The soil island/structure configuration, called the MC (mapped configuration), is obtained by taking a rectangular parallelepiped, called the UC (un-mapped configuration), and mapping it into the MC (figure 4-1). The UC is subdivided into elements by passing sets of planes completely through the UC with the planes parallel to the faces of the parallelepiped (figure 4-2). Each element (node) is identified by a unique I, J, K triplet such that $1 \leq I \leq NI$, $1 \leq J \leq NJ$, and $1 \leq K \leq NK$ where NI , NJ , and NK are the number of elements (plus one) in the x, y , and z directions, respectively. The number of element planes is chosen so that the geometry of the MC can be adequately represented. Since each element can be assigned a material type or be declared void (empty), a variety of soil island/structure configurations can be represented.

4-2 Region Definition

For input purposes the elements are segregated into groups called regions, by selecting subsets from each of the element plane sets (figure 4-2). The selection of the region plane sets is generally made on the basis of structural boundaries and material type interfaces to be modeled in the MC. Material and geometric representation of the soil island/structure configuration is assigned by regions. (See Sections 8-2.9 and 9-2.10.)

4-3 Material Input

Each region is assigned a material type. All elements within a region are initially assumed to be composed of the material associated with the region. This default element material specification can be overridden by an element material type assignment. (See Section 8-4.3.)

4-4 Coordinate Input

The corners and mid-edge points of the regions are given the global (x, y, z) coordinates of the locations they are to occupy in the MC. Thus, the mapped configuration of each region is specified by 20 nodes (8 corners and 12 mid-edge points). (See Section 8-2.10.)

4-5 Mesh Generation

The element mapping from the UC to the MC is facilitated by representing each region by a 20 noded prism with quadratic isoparametric shape functions. The coordinate vector of any generic point within a region is given by

$$\vec{x} = \langle x, y, z \rangle = \sum_{i=1}^{20} N_i \vec{x}_i \quad (4-1)$$

where N_i are the region shape functions and $\vec{x}_i$ are the region nodal MC global coordinate vectors. For the corner nodes

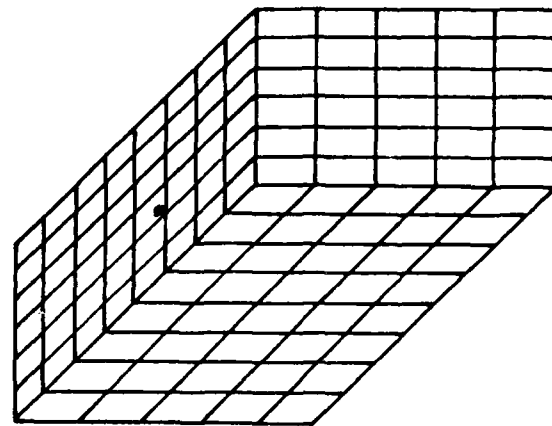
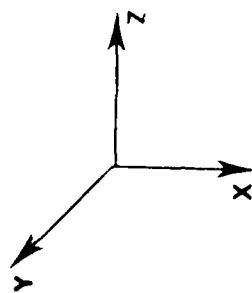
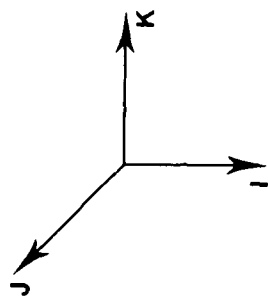
$$N_i = \frac{1}{8}(1 + \xi\xi_i)(1 + \eta\eta_i)(1 + \zeta\zeta_i)(\xi\xi_i + \eta\eta_i + \zeta\zeta_i - 2) \quad (4-2)$$

and for a typical mid-edge node $\xi_i = 0$, $\eta_i = \pm 1$, $\zeta_i = \pm 1$,

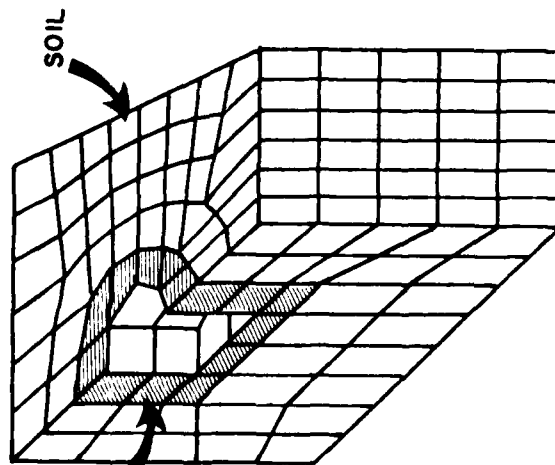
$$N_i = \frac{1}{4}(1 - \xi^2)(1 + \eta\eta_i)(1 + \zeta\zeta_i) \quad (4-3)$$

where ξ , η and ζ are region dimensionless local coordinates and $\langle \xi_i, \eta_i, \zeta_i \rangle$ is the coordinate vector of region node i . The six faces of a region are specified by $\xi = \pm 1$, $\eta = \pm 1$ and $\zeta = \pm 1$.

Since the element configuration of the regions is given, the number of element subdivisions in the ξ , η and ζ directions are known for each region. In a region the element subdivisions of the (ξ, η, ζ) axes are assumed to be uniformly spaced. Consequently, the local region coordinates of all element nodes are known. By applying (4-1) to each element node in every region, the global coordinates of the element nodes in the MC are determined. These MC nodal coordinates can be modified by a nodal coordinate re-assignment (via later input). (See Section 8-4.4.)



UN - MAPPED



MAPPED

FIG. 4-1 SOIL ISLAND STRUCTURE CONFIGURATION: MAPPED & UN-MAPPED

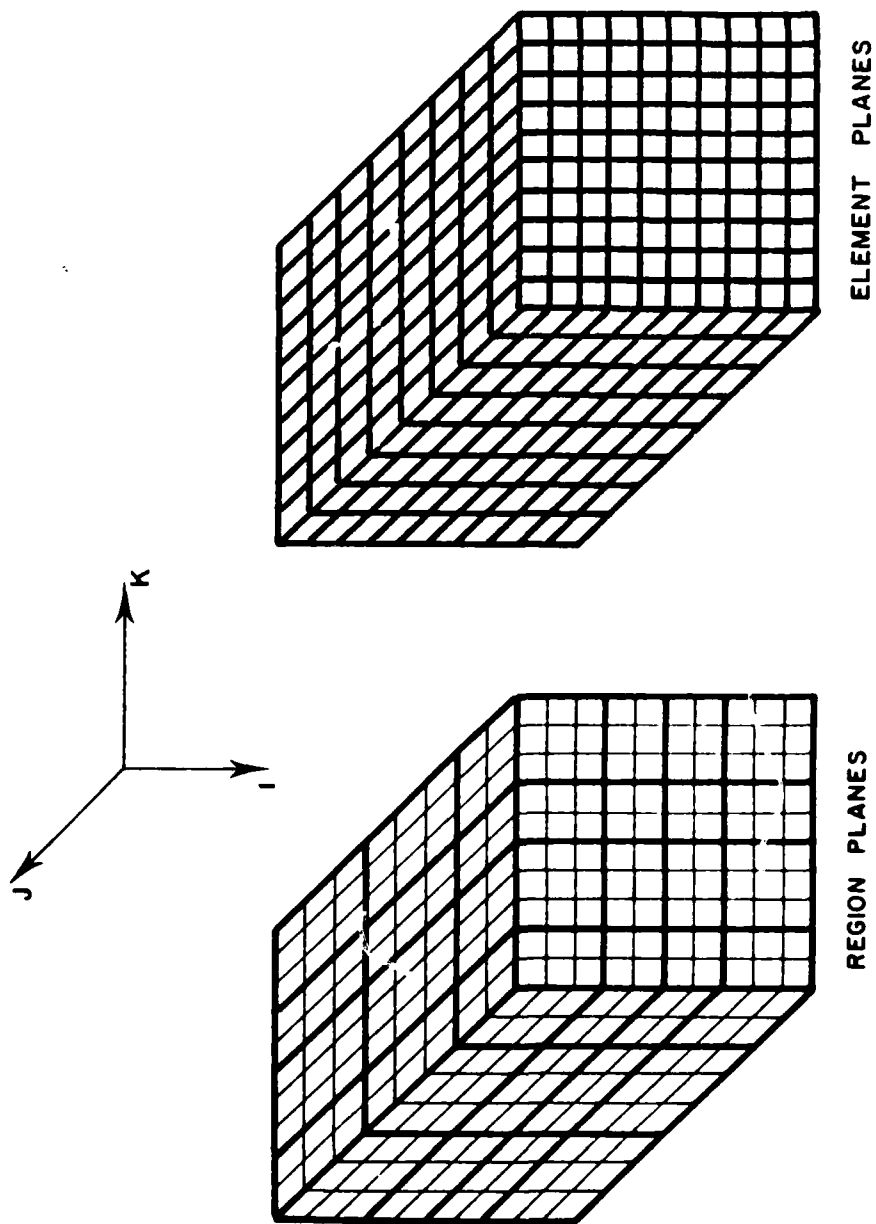


FIG. 4-2 UN - MAPPED CONFIGURATION: ELEMENT & REGION PLANES

SECTION 5

FILES

5-1 Data Files

Element and nodal data are maintained as random files in LCM on the CDC 7600. The input files, the results output files, the SCM common file, the LCM image file, and the applied velocity boundary condition file are stored as sequential files on disk.

The SCM common file and the LCM image file are the restart files. The SCM common file is a copy of active blank common and all labeled common necessary to restart. The LCM image file is a copy of the user's LCM data space.

For more information on data files see Appendix C. Detailed information about the results output files is given in Section 9.

5-2 TRANAL Files

All TRANAL routines are maintained as a CDC UPDATE program sequential library file. A compiled binary object code image of the TRANAL routines is kept on a separate file in system library format.

More information about these files is given in Appendix D.

SECTION 6

PROGRAM FUNCTIONAL STRUCTURE6-1 Zone Definition

In general, the number of elements used to model the soil island and structure will be large enough that all element and nodal data necessary for processing will not fit in small core memory (SCM). Hence, the regions are separated into computational groups, called zones, by selecting a subset from each of the UC region plane sets (figure 6-1). The zone plane sets must be selected so that the element and nodal data, associated with any given zone, will fit in SCM. Because of subcycling, it is also necessary to consider the material configuration of the soil island and structure.

6-2 Zone Processing

Zone processing takes place in SCM one zone at a time. All element and nodal data records, which are necessary to process a zone are transmitted to blank common (the dynamic array - DA) from LCM. The zone is then processed. Data records that were modified during processing are transferred back to their original locations in LCM. This procedure is followed for each zone every time it is processed.

The order in which the zones are processed is determined by the subcycle configuration of the zones. In the case of a single zone problem or in certain subcycle configurations, a zone may reside in core for more than one time step without requiring data from neighboring zones. In these situations, data records are written over the previous records only after the requisite multiple time steps are completed. Hence, a single zone problem is completed with minimal input/output activity.

6-3 Data Management

A family of routines (the data management library - DML) facilitates the handling of data arrays in the DA and the transfer of data records between external storage (LCM and disk) and the DA. The DML uses the labeled common block /DTA MGR/ which contains the control arrays NAME, I POINT, N TYPE, and MODE (the DML arrays).

All SCM data arrays reside in the DA. The dimensions of an array are located in front of the array with the slowest varying dimension next to the first array element. The Hollerith name, the DA pointer, the dimensionality, and the general characteristics of each array are stored in the NAME, I POINT, N TYPE, and MODE arrays, respectively. The DA pointer is the relative address in blank common of the first element of an array. The general characteristic of an array is described by a Hollerith word such as 10HOLLERITH, 10NOCTAL 10HINT OR REL, etc.

Scalars are not stored in the DA. Instead, the Hollerith name of a scalar is put in NAME, the value of the scalar is stored in I POINT and the corresponding values of N TYPE and MODE are set to zero.

Scalars and data arrays are manipulated by way of their names in the NAME array. In other words, the DML is a name-oriented family of routines. (see Appendix E).

6-4 Error Processing

Error checks are built into the DML. For example, if an attempt is made to allocate two entities (data arrays and/or scalars) with the same name, an error is detected. When an error is encountered, a message describing the error is printed along with traceback information. Program execution is terminated and control is returned to the system.

Data consistency error checks are built into many of the data processing control routines. All errors are considered fatal. However, while processing input and boundary condition files, execution is not immediately terminated when a data consistency error is detected. Instead, processing is continued as long as possible so that if other errors exist on these files they can also be found at the same time.

6-5 Subcycling

All elements in a given zone are processed with the same time increment (DT). The largest DT at which a zone can be processed is determined by the element with the severest stability requirement (a function of size and wave speed). Each zone has its own DT requirement which, in general, is unique. The DT of a zone is required to be either an integer multiple or integer divisor of the DT for each of the other zones.

The largest DT for all zones is called the major time increment (MDT). A DT smaller than the MDT is a subcycle time increment (SDT). The integer ratio MDT/SDT is the number of subcycles (NS) that are processed for a given zone within a MDT. The time at the end of a MDT is the major time (MT) and a time at the end of a SDT is a subcycle time (ST).

The NS for a given zone must be an integer multiple of the NS for each zone with a larger DT, and an integer divisor of the NS for each zone with a smaller DT.

Processing proceeds one MDT at a time. All MDT zones, i.e. $NS=1$ zones, are processed first. Within the MDT, the SDT zones, i.e. $NS > 1$ zones, are processed so that the following two conditions are always satisfied.

- 1) The ST for any given zone cannot be greater than the ST of any adjacent zone with a smaller NS.
- 2) The ST for any given zone can only be one SDT greater than the ST of any adjacent zone with the same or a greater NS.

Processing within the MDT is complete when the ST for all zones is equal to the MT.

6-6 Zone Interfaces

In general, nodes on zone interfaces are associated with elements in more than one zone. When a zone is processed the interface nodal velocities are not immediately updated. Instead, force increments are computed and added to the interface nodal forces. After all zones surrounding an interface node set have been processed, the velocities of the node set are updated. This zone interface nodal velocity updating procedure is carried out each time a zone is processed, i.e. every subcycle.

Boundary Conditions

Force type boundary conditions are implemented by adding applied force increments to the zone interface nodal forces at the soil island boundary before the nodal velocities are updated.

Velocity type boundary conditions are applied only after the soil island boundary zone interface nodal velocities have been updated by the forces.

Both force and velocity type boundary conditions are applied on a subcycle basis.

A set of routines based on Ref. 28 is provided for applying an air overpressure shock wave to the boundary surfaces of the soil island. For an element face on the soil island boundary, the overpressure increment at the centroid of the element face is determined. From the overpressure increment the force increment normal to the element face at the centroid is computed and distributed to the corresponding nodal forces.

A specialized set of routines is provided for applying velocity boundary conditions from two dimensional rotationally symmetric calculations (soil island method). The routines that apply the kinematic (interface velocity) boundary conditions assume the x axis to be down and the y axis to be in the radial direction. The kinematic boundary conditions are presumed to be rotationally symmetric and are applied to only the positive y-z surface and the x-z surfaces of the soil island. The structure of the interface boundary condition file is given in Table C-2 of Appendix C.

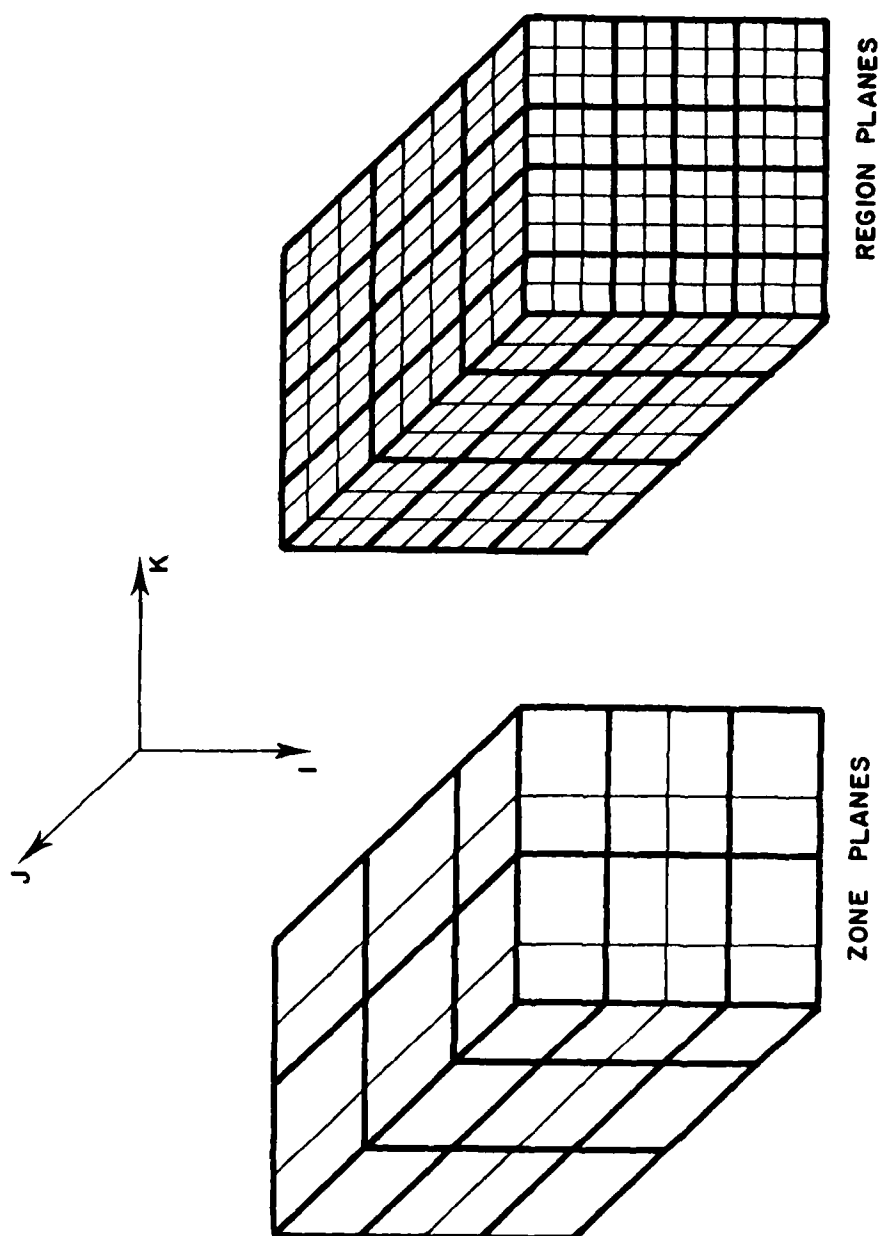


FIG. 6-1 UN-MAPPED CONFIGURATION: REGION & ZONE PLANES

SECTION 7

INPUT LANGUAGE

Problems to be analyzed by TRANAL are described by directives. A directive may define a physical quantity, such as a soil island configuration, or request an action, such as a list of the region materials.

Directives may appear in any order. However, certain directives must be included in the input while others are optional. Tables 8-1 and 11-1 are lists of the TRANAL directives, a brief description of their function, and if they are required.

7-1 Directive Format

All the directives have the same general format:

< directive prefix > < directive > < directive data >

< directive prefix > is a dollar sign (\$) or an asterisk (*). A dollar sign is used if an echo output of the directive and its data is desired. An asterisk suppresses the echo output.

< directive > is a valid directive from Tables 8-1 and 11-1.

< directive data > is a specific set of information associated with each directive. (See Section 8, Processors and Input.)

- Notes:
- 1) The brackets, < >, are not input, but are used only to describe the Input Language.
 - 2) Any number of blanks may be placed between the directive prefix and the directive, as well as between the words of multi-word directives.

7-2 Input Language Description

The following conventions are used to describe the TRANAL Input Language:

< int 1 >	}	entities; the brackets, < >, are not included.	
< xl >			
< name 1 >			
:			
etc.	}	labels; as required in the directives.	
TO			
BY			
VOID			
:			
etc.	}	or, choose either < 1 > or < 2 >.	
< 1 > < 2 >			
{ < 1 > }			< 1 > is optional.
[< 1 > < 2 >] _{n1} ⁿ²			repeat the entities enclosed in brackets between n1 and n2 times.

7-3 Input Constructions

7-3.1 Entities

A directive and its data consists of entities distinguished from each other by separators. Directives may appear on any number of cards, but an entity may not be split between cards.

There are four types of entities: integer, real, label and name.

Integer: A signed or unsigned number without a decimal point.

Examples: 123 -321 0

Real: A signed or unsigned number with a decimal point and optionally a signed or unsigned exponent. The exponent may be delimited by an E or a sign.

Examples: 123.45 -321. -456.7E2 -456.7+2 756-2.

Label: An alphanumeric entity used in various directives.

Examples: TO BY VELOCITY CORNERS.

Name: An arbitrary string of characters, excluding the slash (/), enclosed in quotation marks.

Examples: 'SAMPLE RUN #1' 'TEST *1*' 'MAT-CONCRETE' 'A36 STEEL'.

7-3.2 Separators

Entities may be separated by one or more blanks, a single comma (,), or a combination of blanks and a comma. Spacing between entities, even across card boundaries, is arbitrary.

7-3.3 Comments

The input may be annotated with comments by use of a slash (/). All characters on a card to the right of a slash appear on the echo output but are ignored in the processing. A slash may appear in any card column.

7-3.4 Range Specification

The range specification specifies a set of integers. When a range specification (< range >) appears, it is to be replaced by the following syntax:

< int 1 > { TO < int 2 > { BY < int 3 > } }

The set includes all integers from < int 1 > to < int 2 > in steps of < int 3 >. If BY < int 3 > is omitted, a step of one is assumed. If TO < int 2 > is omitted, the upper limit is assumed to be < int 1 > .

< int 2 > must be greater than or equal to < int 1 >, and all integers must be greater than zero.

Example

<u>range</u>	<u>integers included</u>
6	6
3 TO 5	3 4 5
1 TO 7 BY 3	1 4 7
2 TO 9 BY 4	2 6
7 TO 5	invalid
-3	invalid

7-3.5 Keywords

Several directives allow input with a keyword format. A keyword is a label associated with a value or list of values. The use of keywords is optional. If keywords are used, the labels and associated values may appear in any order. If keywords are not used, the data must be in the order described in the directive's syntax description.

If keywords are used and a datum has a default value, the omission of the keyword will cause the assignment of the default value to the datum.

An equal sign (=) is a valid separator between a keyword and its value(s).

Example

A directive, called KEYEXAM, requires three pieces of data: DATA1, ADAT2, and BDAT3.

The syntax is described as:

```
$|* KEYEXAM DATA1 < x1 > ADAT2 < x2 > BDAT3 < x3 >
```

```
< x1 > }
< x2 > } real values
< x3 > }
```

Two valid choices of entering the directive that have same result are:

```
$ KEYEXAM ADAT2 = 1.0 DATA1 = 2.0 BDAT3 = 3.0
```

```
$ KEYEXAM 2.0 1.0 3.0
```

Notice that if keywords are used, the values may appear in any order, whereas in the non-keyword format the order given in the directive syntax description must be followed.

7-4 Alternate Input Files

Certain directives allow input from files other than the primary input file to facilitate transfer of large amounts of data, see Tables 8-1 and 11-1. For example the directive,

\$ REGION MATERIALS TAPE = 18

on the primary input file, would direct the input processor to read the region material information from the file attached to TAPE18. The alternate input files must be formatted card images with standard syntax rules for the directives.

7-5 Error Messages

If an error occurs during the processing of the input, an informative message is printed on the output. Processing of the directive in error terminates, but the syntax of the remaining directives is checked. Further TRANAL processing terminates after the input phase.

There are two types of error messages: syntactical and directive. The syntactical error messages indicate where in the line the error occurred and a brief message as to the cause. The directive error messages, listed with each directive description, indicate an error in the use of the directive.

7-6 Directive Abbreviations

All directives have an abbreviation. The abbreviations for a given processor are unique. However, the abbreviations are not necessarily unique between processors. The abbreviations are underlined in Tables 8-1 and 11-1, and are listed in Tables 8-2 and 11-2.

SECTION 8

PROCESSORS AND INPUT

The running of a TRANAL job is broken down into a series of execution steps called processors. In general, a processor operates from or on data files and SCM common data sets produced by preceeding processors.

Random files are not required to appear on processor program cards unless the system mass storage input/output routines are used. If the system mass storage routines are used, the random files will reside on disk rather than in LCM and this is not recommended. As long as the user's LCM data space can be transferred directly from LCM to disk and vice versa, the LCM image file (tape 60) is not needed on processor program cards.

The processor driver programs are maintained on the UPDATE program library file TRNLDIR. (See Appendix D.)

8-1 START UP Processor

START UP is basically an initialization processor. The soil/island structure zone, region, element, geometric, and material configurations are read from the input file(s). The element information file is built, and the node coordinate file is constructed. A file is cataloged by creating the file with data records containing zeros.

Based upon the stability requirement that must be satisfied by each individual element, a set of recommended zone time increments is determined and a suggested zone subcycle configuration is ascertained. If negative volume elements are encountered while calculating time increments, an element identification message is printed on the output file each time an element with a negative volume is discovered. However, the number of such messages is restricted by a print limit. The total number of negative volume elements in each zone is also printed.

The files associated with START UP are:

Input	- tape 1.
System output	- tape 2.
Region corner coordinates	- tape 11.
Region x-edge coordinate directives	- tape 12.
Region y-edge coordinate directives	- tape 13.
Region z-edge coordinate directives	- tape 14.
Node coordinates	- tape 15.
Element information	- tape 32.
SCM common	- tape 50.
LCM image	- tape 60.

Secondary (i.e. alternate) input files may be required.

8-2 START UP Input Directives

8-2.1 TITLE

The TITLE directive reads a user written description of the problem for the primary results output file.

Syntax

\$|* TITLE < name >

< name > is a description enclosed in quotation marks. As many as 78 characters may be used (name).

Default

If the directive is omitted, a date and time title is created.

Example

\$ TITLE 'BURIED T-SECTION 7-1-78'

8-2.2 IDENTIFIER

The IDENTIFIER directive establishes a problem identifier for the results output file(s).

Syntax

\$|* IDENTIFIER < identifier >

< identifier > is a problem identifier enclosed in quotation marks. Only the first ten characters are recognized (name).

Default

There is no default; the IDENTIFIER directive must be included in the input.

Example

\$ IDENTIFIER 'TEST RUN'

8-2.3 PRINT OPTIONS

The PRINT OPTIONS directive prints user chosen data on the echo output listing after processing of the input.

Syntax

\$|* PRINT OPTIONS ALL | { < option > }₀[∞]

< option > is a valid print option (label).

ALL causes the printing of all the available options.

The available print options are:

MATERIAL PROPERTIES
REGION MATERIALS
REGION CORNERS

Note: Any number of blanks may separate the words in two word options.

Default

Omission of the PRINT OPTIONS directive will suppress printing of all the options.

Example

```
$ PRINT OPTIONS REGION MATERIALS, MATERIAL PROPERTIES
```

8-2.4 ZONE DEFINITION

The ZONE DEFINITION directive establishes the configuration of the soil island.

Syntax

```
$|* ZONE DEFINITION [ < coord > ZONES < int 1 >  
      REGIONS [ < int 2 > ]n1 ELEMENTS [ < int 3 > ]n2 ]3  
< coord > is X | Y | Z (label).  
< int 1 > is the number of zones in < coord > (integer).  
n1       is the number of zones; the value of < int 1 >.  
< int 2 > is the number of regions in each zone (integer).  
n2       is the number of regions; the sum of the values of  
          < int 2 > for a given < coord >.  
< int 3 > is the number of elements in each region (integer).
```

Defaults

There are no defaults; the soil island must be fully specified.

Example

```
$ ZONE DEFINITION X ZONES 2  
REGIONS 2 3 ELEMENTS 1 2 3 4 5  
Z ZONES 2  
REGIONS 1 3  
ELEMENTS 4 3 2 1  
Y ZONES 2  
REGIONS 2 2 ELEMENTS 1 2 3 4
```

8-2.5 ZONE TYPE

The ZONE TYPE directive defines the type of elements in each zone.

Syntax

\$|* ZONE TYPE [< range > VOID | < zon type >]₀[∞]

< range > is a range of zone numbers.

< zon type > a valid zone type (label).

The zone types available are:

HEX8

Default

If a zone is not specified, it is assumed to be HEX8.

Example

\$ ZONE TYPE 3 TO 4 HEX8, 1 VOID, 2 HEX8

Error Message

**** SPEC. EXCEEDS NUM. OF ZONES

An attempt was made to specify a zone that does not exist.

8-2.6 MATERIALS

The MATERIALS directive defines the constitutive properties of the materials in the problem.

Syntax - keyword format

\$|* MATERIALS { TAPE < int > } { ALL [< prop > < value >]₀[∞] }
[NAME [< name >]₁¹ TYPE < type > [< prop > < value >]₀[∞]]₁[∞]
< int > is an alternate input file tape number (integer).
< prop > is a constitutive material property keyword (label).
< value > is the value associated with each material property (real).
< name > is the material name enclosed in quotation marks (name).
Each name may have up to 10 characters.
< type > is a valid material type (label).

Note: Although the MATERIALS directive employs keywords, the NAME data must precede the TYPE data (if given), and the TYPE data must precede the material properties. NAME and TYPE are keywords though, and as such the labels are optional.

The valid material types are:

ELASTIC
SOIL
ROCK

The constitutive material properties for an ELASTIC material are:
RHO K G

The constitutive material properties for SOIL and ROCK materials are:
RHO K G A B C D R W TCUT GEOP

Defaults

1. The default material type is ELASTIC.
2. If a material property is not specified, the value give in the ALL portion of the directive is used.

Example

```
$ MATERIALS ALL G = 10.0 K = 12.0

NAME 'MAT1' RHO = 2.0
'MAT3' RHO = 4.0 K = 13.0 G = 11.1
NAME = 'MAT4' TYPE = ELASTIC 4.0, 12., 10.3
'MAT5' ROCK 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0
          9.0 10.0 11.0
```

Error Message

```
**** < prop > UNDEFINED FOR MATERIAL < name >
      In a keyword format, the quantity < prop > was not defined
      in an ALL specification or with the material name.
```

8-2.7

STABILITY

The STABILITY directive defines the stability factor and the print limits.

Syntax - keyword format

```
$|* STABILITY FACTOR < value > VOLUME < int 1 > ANGLE < int 2 >

< value > is the stability factor (real).
< int 1 > is the volume print limit (integer).
< int 2 > is the angle print limit (integer).
```

Default

If the directive or values are omitted, a stability factor of 0.8 and limits of 10 are assumed.

Example

```
$ STABILITY .5 5 5
```

Error Messages

```
**** STABILITY OUT OF RANGE
      The stability factor must be greater than zero and less
      than or equal to one.
```

8-2.8 DAMPING

The DAMPING directive establishes the damping properties of the materials used in the problem.

Syntax - keyword format

```
$ * DAMPING VELOCITY < x1 > [ NAME [ < name > ]1∞ CLINEAR < x2 >  
    CQUADRATIC < x3 > ELINEAR < x4 > EQUADRATIC < x5 > ]0∞
```

< x1 > is the velocity factor for the time increment computation (real).

< name > is a material name defined in the MATERIALS directive (name).

$\left. \begin{array}{l} < x2 > < x3 > \\ < x4 > < x5 > \end{array} \right\}$ are damping properties (real).

Defaults

1. The default for VELOCITY is 0.0.
2. The default damping coefficients for each material are all 0.0.

Example

```
$ DAMPING VELOCITY = .2 'MATERIAL 1' .1 1. .2 4. 'MATERIAL 2'  
    'MATERIAL26' 'MATERIAL 4' .3 6. .4 12.  
    NAME = 'MATERIAL27' CQUADRATIC = .7
```

Error Messages

1. **** LIST OF NAMES MISSING
 No materials were specified after the NAME keyword.
2. **** < name > NOT A VALID MATERIAL
 < name > was not defined in the MATERIALS directive.

8-2.9 REGION MATERIALS

The REGION MATERIALS directive associates a material defined in the MATERIALS directive with each region in the soil island.

Syntax

```
$|* REGION MATERIALS { TAPE < int > } [ < range > VOID | < name > ]1∞
```

< int > is an alternate input file tape number (integer).

< range > is a range of region numbers.

< name > is a valid material name (name).

Default

There are no defaults; all regions must be specified.

Example

```
* REGION MATERIALS  
2 VOID  
1 'MAT1'  
3 TO 6 'MAT2'  
7 TO 8 VOID
```

Error Messages

1. **** SPEC EXCEEDS NUM. OF REGIONS
A region outside the soil island was used.
2. **** < name > NOT A VALID MATERIAL
< name > was not properly declared in the MATERIALS directive.
3. **** REGION < x > MUST BE VOID
All regions in a void zone must be declared VOID.
4. **** MATERIAL NOT SPECIFIED FOR REGION < x >
All regions must be assigned a material, or declared VOID.

8-2.10 REGION COORDS

The REGION COORDS directive defines the corner and edge coordinates of the regions.

Syntax

```
$|* REGION COORDS { TAPE < int > } CORNERS [ < x1 > < y1 > < z1 > ]nl  
[ < coord > EDGES ALL LINEAR | [ < range > LINEAR |  
< range > QUADRATIC [ < x2 > < y2 > < z2 > ]nln2 ]1∞ ]o3  
< int > is an alternate input file tape number (integer).  
< x1 > }  
< y1 > } are the corner coordinates of the regions (real).  
< z1 > }  
nl is the number of corners.  
< coord > is a coordinate X | Y | Z (label).  
< range > is a range of region edge numbers.  
< x2 > }  
< y2 > } are edge coordinates for quadratic edges (real).  
< z2 > }  
n2 is the number of edges specified in the < range >.
```

Default

All the edges are assumed to be linear unless declared quadratic.

Example

```
* REGION COORDS  
  CORNERS  
    . . .  
    . . .  
    . . .  
    . . .  
    . . .  
  
X EDGES  
  1 TO 6 LINEAR  
  7 QUADRATIC 1., 2., 3.  
  8 TO 11 LINEAR  
 12 TO 13 QUADRATIC 4.0, 5.0, 6.0  
                   4.5, 5.5, 6.5
```

Y EDGES ALL LINEAR

Error Messages

1. **** < coord > ALREADY SPECIFIED
The edges in a given direction may be specified only once.
2. **** < coord > IS AN INVALID SPEC.
An attempt was made to specify an invalid coordinate or edge type.
3. **** EDGE (< x >) OUT OF RANGE
An attempt was made to specify an edge not in the soil island.

3-3 ALTER FL Processor

The ALTER FL processor changes components of the element information file and/or nodal coordinates in the node coordinate file according to input specifications. After all element and coordinate file modifications have been made, a new recommended set of zone time increments, and a new suggested zone subcycle configuration are determined. Just as for the START UP processor, the element volumes are checked and an element identification message is printed when a negative volume is found.

The ALTER FL processor is executed, as many times as necessary, in order to establish the proper element information and node coordinate files. The files required by ALTER FL are:

Input	-	tape 1.
System output	-	tape 2.
Node coordinates	-	tape 15.
Element information	-	tape 32.
SCM common	-	tape 50.
LCM image	-	tape 60.

3-4 ALTER FL Input Directives

8-4.1 ALTER STABILITY

The ALTER STABILITY directive modifies the stability factor and the print limits.

Syntax - keyword format

\$|* ALTER STABILITY FACTOR < value > VOLUME < int 1 > ANGLE < int 2 >

< value > is the redefined stability factor (real).

< int 1 > is the volume print limit (integer).

< int 2 > is the angle print limit (integer).

Default

If the directive or values are omitted, the stability factor and print limits are left unchanged.

Example

```
* ALTER STABILITY .7 20 15
```

Error Message

```
**** STABILITY OUT OF RANGE
      The stability factor must be greater than zero
      and less than or equal to one.
```

8-4.2 ALTER DAMPING

The ALTER DAMPING directive modifies the damping properties of the materials.

Syntax - keyword format

```
$|* ALTER DAMPING VELOCITY < x1 > [ NAME [ < name > ]1∞ CLINEAR < x2 >
      CQUADRATIC < x3 > ELINEAR < x4 > EQUADRATIC < x5 > ]0∞
< x1 >    is the velocity factor for the time increment computation (real)
< name >  is a material name defined in the MATERIALS directive (name).
          (See Section 8-2.6.)
< x2 > < x3 > } are damping properties (real).
< x4 > < x5 > }
```

Defaults

1. If the directive is omitted, the velocity factor and all damping properties are left unchanged.
2. If the directive is given, any quantity not specified is left unchanged.

Example

```
$ ALTER DAMPING
VELOCITY = .4
'DRY CLAY' .1 .2 3. 4.
NAME = 'WET CLAY' ELINEAR = .5
```

Error Messages

1. **** LIST OF NAMES MISSING
No materials were specified after the NAME keyword.
2. **** < name > NOT A VALID MATERIAL
< name > was not defined in the MATERIALS directive.
(See Section 8-2.6.)

8-4.3 ALTER ELEMENT

The ALTER ELEMENT directive modifies the element information file (tape 32).

Syntax

```
$|* ALTER ELEMENT [ VOID | < name > [ < range i >
    < range j > < range k > ]0∞ ]0∞
< name >      is a material name defined in the MATERIALS directive
               (name). (See Section 8-2.6.)
< range i > }
< range j > } are ranges of element I, J, K numbers in the soil island.
< range k > }
```

Defaults

1. If the directive is omitted, the file is not modified.
2. If the directive is given, only file entries corresponding to the specified elements are replaced.

Example

```
$ ALTER ELEMENT
'CONCRETE' 1 4 7
'CLAY' 3 TO 6 7 2 TO 12 BY 3
VOID 5 3 TO 8 BY 2 4
```

Error Messages

1. **** < name > NOT A VALID MATERIAL
 < name > was not defined in the MATERIALS directive.
 (See Section 8-2.6).
2. **** < x > NOT IN SOIL ISLAND
 An attempt was made to specify an I, J, K station not in
 the soil island.

8-4.4 ALTER NODE

The ALTER NODE directive modifies the node coordinate file (tape 15).

Syntax - keyword format

```
$|* ALTER NODE [ < range i > < range j > < range k >
    COORDS [ < coord x > < coord y > < coord z > ]nlnl ]0∞
< range i > }
< range j > } are ranges of node I, J, K numbers in the soil island.
< range k > }
nl          total number of nodes specified in the I, J, K ranges.
< coord x > }
< coord y > } node coordinates (real).
< coord z > }
```

The order of the specified nodes is determined by the normal FORTRAN subscript convention, i.e. the I range increases most rapidly, and the K range increases least rapidly.

Defaults

1. If the directive is omitted, the file is not modified.
2. If the directive is given, only the coordinates of the specified nodes are replaced.

Example

```
* ALTER NODE 2 3 1 COORDS = .7 1.6 .9
3 2 TO 6 BY 2 4
2.1 2.6 3.2 3.6 5.7 4.3
5.9 6.8 7.7
```

Error Message

```
**** < x > NOT IN SOIL ISLAND
      An attempt was made to specify an I, J, K station not in the
      soil island.
```

8-5 CLEAN UP Processor

The CLEAN UP processor completes all file preparations necessary for the PROSSOR processor. The three main header records for the primary results output file (tape 3) are built and written on the file. The random boundary condition file (tape 6) is established. Node masses are constructed as the fourth component of the coordinates and written on the node coordinate file (tape 15). The force, stress, and velocity files are cataloged.

CLEAN UP also establishes the zone subcycle configuration, builds the subcycle control tables, removes all inactive data arrays from the DA, and sets static type data array pointers and parameters.

CLEAN UP requires the following files:

Input	-	tape 1.
System output	-	tape 2.
Results output	-	tape 3.
Applied boundary		
Velocities (sequential)	-	tape 5.
Applied boundary		
Velocities (random)	-	tape 6.
Node coordinates	-	tape 15.
Node forces	-	tape 20.
Element stresses	-	tape 30.
Element information	-	tape 32.
Node velocities	-	tape 40.
SCM common	-	tape 50.
LCM image	-	tape 60.

8-6 CLEAN UP Input Directives

8-6.1 PRINT OPTIONS

This directive has the same function and usage as the PRINT OPTIONS directive for the START UP processor (Section 8-2.3).
The available print options for CLEAN UP are:

VELOCITY OUTPUT
STRESS OUTPUT
STRAIN OUTPUT

8-6.2 TIME DATA

This directive defines the integration starting time and the major time increment.

Syntax - keyword format

\$|* TIME DATA START < time 1 > DELTAT < time 2 >

< time 1 > is the integration starting time (real).
< time 2 > is the major time increment (real).

Defaults

1. START is 0.0 if not specified.
2. If not specified, DELTAT is the major time increment determined by the START UP or the ALTER FL processor.
3. If the directive is omitted, both START and DELTAT are defaulted.

Examples

```
$ TIME DATA 1.5 0.55
* TIME DATA DELTAT = 0.55
  START = 1.5
```

Error Message

- **** SPEC. MUST BE NON-NEGATIVE
- a. START must be greater than or equal to zero.
 - b. DELTAT must be greater than zero.

8-6.3 SUBCYCLE RATIOS

The SUBCYCLE RATIOS directive establishes the subcycle ratios for the analysis.

Syntax

\$|* SUBCYCLE RATIOS [< ratio >]₀[∞]

< ratio > is the subcycle ratio (integer).

Note: Each subcycle ratio must be an integral multiple of the preceding ratio.

Defaults

1. If the directive is given and no ratios are specified, no subcycling takes place.
2. If the directive is omitted, the subcycle ratio set determined by the START UP or the ALTER FL processor is used.

Note: If the directive is given, all non-zero zone ratios are set to one (1).

Example

```
$ SUBCYCLE RATIOS 3 6 12 24
```

Error Message

```
**** INVALID MULTIPLE ( < ratio > ) IN SUBCYCLES
The restriction on integral multiples of subcycle ratios
must be followed.
```

8-6.4 ZONE SUBCYCLES

This directive defines the subcycle ratio for each zone.

Syntax - keyword format

```
$|* ZONE SUBCYCLES [ < range > RATIO < ratio > ]0∞
```

< range > is a range of zone numbers.
< ratio > is a subcycle ratio (integer).

Defaults

1. If no ratio information is given after the directive or the directive is omitted, the existing zone subcycle configuration is used.
2. If the directive is given, only the subcycle ratios of the zones specified are changed.

Example

```
* ZONE SUBCYCLES 1 RATIO = 24 2 RATIO 3
4 TO 10 BY 2 6
```

Error Messages

1. **** SPEC. EXCEEDS NUM. OF ZONES
An attempt was made to specify a zone that does not exist.
2. **** INVALID SUBCYCLE (< ratio >) IN ZONE TYPE
An attempt was made to specify an invalid subcycle ratio.
3. **** NO LEVEL ONE ZONE
At least one zone with a subcycle ratio of one (1) must exist.

8-6.5 BOUNDARY CONDITIONS

The BOUNDARY CONDITIONS directive defines the boundary conditions on the six sides of the soil island.

Syntax

\$|* BOUNDARY CONDITIONS ALL < con > | [< con >]₆⁶
< con > is a valid boundary condition (label).

The All specification applies the same boundary condition to six sides. The available boundary conditions are:

FREE BOUNDARY
APPLIED VELOCITY
OVER PRESSURE
SOIL ISLAND
SYMMETRY PLANE

Default

If the boundary condition directive is omitted, all six sides are given FREE BOUNDARY.

Example

\$ BOUNDARY CONDITIONS APPLIED VELOCITY, SYMMETRY PLANE
SOIL ISLAND, SOIL ISLAND, OVERPRESSURE, SOIL ISLAND

8-6.6 OUTPUT

The OUTPUT directive defines the quantities and locations in the soil island where results are to be recorded for further analysis.

Syntax

\$|* OUTPUT [< type > [< range i > < range j > < range k >]₀[∞]]₀³

< type > a valid output quantity (label).

$\left. \begin{array}{l} \text{< range i >} \\ \text{< range j >} \\ \text{< range k >} \end{array} \right\}$ are ranges of I, J, K numbers in the soil island.

The available output quantities are:

VELOCITY
STRESS
STRAIN

Default

No output is produced for a quantity unless the stations are specified.

Example

```
$ OUTPUT STRAIN 1 TO 7 BY 3, 4, 11 TO 15
VELOCITY 1 1 1, 2 1 3
STRESS 1 1 1 TO 5, 2 8 8
```

Error Messages

1. **** < type > ALREADY SPECIFIED
An output quantity may only be specified once in the directive.
2. **** < x > NOT IN SOIL ISLAND
An attempt was made to specify an I, J, K station not in soil island.

8-6.7 CORNER FACTOR

The CORNER FACTOR directive establishes the corner fuzz factors for matching LAYER type boundaries to the soil island.

Syntax - keyword format

```
$|* CORNER FACTOR HORIZ < x1 > VERT < x2 >
```

$\left. \begin{array}{l} < x1 > \\ < x2 > \end{array} \right\}$ are the corner fuzz factors (real).

Defaults

1. HORIZ = 0.1
2. VERT = 0.1

Examples

```
$ CORNER FACTOR VERT = .2
$ CORNER FACTOR .1 .2
```

8-6.8 BRODE LOAD

This directive defines the parameters for the air overpressure load.

Syntax - keyword format

```
$|* BRODE LOAD YIELDKT < yield > HOBFEET < height >
```

< yield > is the weapon's yield in kilotons (real).
< height > is the height of burst in feet (real).

Defaults

1. If the directive is omitted, the air overpressure load is not applied.
2. If the OVER PRESSURE boundary condition is specified (Section 8-6.5), the directive must be on the input file, and both YIELDKT and HOBFEET must be given.

Examples

```
$ BRODE LOAD 250.0 50.0
* BRODE LOAD HOBFEET = 50.0
  YIELDKT = 250.0
```

Error Messages

**** SPEC. MUST BE NON-NEGATIVE

- a. YIELDKT must be greater than zero.
- b. HOBFEET must be greater than or equal to zero.

8-7 PROSSOR Processor

The PROSSOR processor writes its identification record on the results output file (tape 3). Time integration is performed and the results output buffer is written on tape 3 at the completion of each major time increment.

PROSSOR requires the following files:

Input	-	tape 1.
System output	-	tape 2.
Results output	-	tape 3.
Applied boundary		
Velocities (sequential)	-	tape 5.
Applied boundary		
Velocities (random)	-	tape 6.
Node coordinates	-	tape 15.
Node forces	-	tape 20.
Element stresses	-	tape 30.
Element information	-	tape 32.
Node velocities	-	tape 40.
SCM common	-	tape 50.
LCM image	-	tape 60.

The results output file (tape 3) for the first execution of PROSSOR must be the tape 3 produced by CLEAN UP and must be rewound. Each subsequent execution of PROSSOR produces a separate result output file.

The SCM common file (tape 50) and the LCM image file (tape 60) are the restart files. The $N^{\text{th}} + 1$ execution of PROSSOR must use tape 50 and tape 60 produced by the N^{th} execution ($N \geq 1$). The LCM image file contains a complete copy of the user's LCM data space, i.e., the random files: tape 6, tape 15, tape 20, tape 30, tape 32, and tape 40.

8-8 PROSSOR Input Directives

8-8.1 TIME INCREMENTS

The TIME INCREMENTS directive establishes the number of major time increments that will be processed.

Syntax

\$|* TIME INCREMENTS < number >

< number > is the number of major time increments (integer).

Default

There is no default; the TIME INCREMENTS directive must be included in the input.

Example

\$ TIME INCREMENTS 60

8-9 COMBINE Processor

This processor combines the results output files (tape 3's) produced by multiple executions of PROSSOR.

COMBINE requires the following files:

Input	-	tape 1.
System output	-	tape 2.
Combined results	-	tape 70.
Results output	-	tape 7N, (N=1, 2, 3, ...).

Tape 7N must be the results output file (tape 3) produced by the Nth execution of PROSSOR.

8-10 COMBINE Input Directive

8-10.1 PRINT OPTIONS

This directive has the same function and usage as the PRINT OPTIONS directive for the START UP processor (Section 8-2.3).

The only available print option for COMBINE is:

RESULTS HEADER

8-10.2 COPY RECORDS

The COPY RECORDS directive describes the manner in which the results output files are to be combined.

Syntax

```
$|* COPY RECORDS [ < int 1 > ]nlnl
```

< int 1 > is the number of records to be copied from a results output file to the combined file (integer).
nl is the number of results output files to be combined.

Default

If zero is given as the number of records to be copied, then all the records from that results output file will be copied.

Example

```
$ COPY REOCRDS 10 0 0
```

8-11 TRNSPOS Processor

The output data records on the combined results file are transposed to time history records by the TRNSPOS processor.

The files used by TRNSPOS are:

Input	-	tape 1.
System output	-	tape 2.
Transposed results	-	tape 8.
Combined results	-	tape 70.

8-12 TRNSPOS Input Directives

8-12.1 PRINT OPTIONS

This directive has the same function and usage as the PRINT OPTIONS directive for the START UP processor (Section 8-2.3).

The only available print option for TRNSPOS is:

RESULTS HEADER

8-12.2 TRANSDPOSE

The TRANSDPOSE directive specifies how the combined results file is to be transposed.

Syntax - keyword format

```
$|* TRANSPOSE TYPE < type > MINIMUM < int 1 >  
    { RECORDS [ < range > ]1∞ }
```

< type > is the type of transpose FULL | PARTIAL (label).

< int 1 > is the minimum number of transposed records that must fit in SCM for execution (integer).

< range > is a range of records to be transposed if TYPE is PARTIAL.

Note: If a partial transpose is specified, the RECORDS information must be given preceded by the RECORDS label.

Default

The default for MINIMUM is 10.

Examples

```
$ TRANSPOSE FULL  
$ TRANSPOSE TYPE = PARTIAL MINIMUM = 8  
  RECORDS 1 TO 7 BY 3, 8, 9, 10
```

Error Messages

1. **** TRANSPOSE TYPE NOT SPECIFIED
Type must be given as FULL or PARTIAL
2. **** < type > IS AN INVALID TYPE
Again, the type of transpose must be FULL or PARTIAL
3. **** TRANSPOSE RECORDS NOT SPECIFIED.
If a partial transpose is desired, the RECORDS label and data must be given.

TABLE 8-1
PROCESSOR INPUT DIRECTIVES

START UP Directives

Directive	Function	Section
<u>TITLE</u>	Gives a problem description for headings.	8-2.1
<u>IDENTIFIER</u> *	Provides an identifier for the results output file(s).	8-2.2
<u>PRINT OPTIONS</u>	Requests output of various quantities.	8-2.3
<u>ZONE DEFINITION</u> *	Defines the soil island/structure configuration.	8-2.4
<u>ZONE TYPE</u>	Defines the zones.	8-2.5
<u>MATERIALS</u> *†	Defines the constitutive properties of the materials in the soil island and structure.	8-2.6
<u>STABILITY</u>	Defines the stability factor.	8-2.7
<u>DAMPING</u>	Defines the damping properties of the materials.	8-2.8
<u>REGION MATERIALS</u> *†	Specifies each region's material.	8-2.9
<u>REGION COORDS</u> *†	Establishes corner and edge coordinates of the regions.	8-2.10

* Directive must be on the input file.

† Data may be on an optional alternate input file.

TABLE 8-1 (Cont'd)

ALTER FL Directives

Directive	Function	Section
<u>ALTER STABILITY</u>	Modifies the stability factor.	8-4.1
<u>ALTER DAMPING</u>	Modifies the damping properties of the materials.	8-4.2
<u>ALTER ELEMENT</u>	Modifies the element information file.	8-4.3
<u>ALTER NODE</u>	Modifies the node coordinate file.	8-4.4

CLEAN UP Directives

<u>PRINT OPTIONS</u>	Requests output of various quantities.	8-6.1
<u>TIME DATA</u>	Defines starting time and major time increment.	8-6.2
<u>SUBCYCLE RATIOS</u>	Establishes the subcycle ratios.	8-6.3
<u>ZONE SUBCYCLES</u>	Defines subcycle ratio for each zone.	8-6.4
<u>BOUNDARY CONDITIONS</u>	Defines the boundary conditions on the soil island.	8-6.5
<u>OUTPUT</u>	Requests quantities to be output during processing.	8-6.6
<u>CORNER FACTOR</u>	Defines the corner fuzz factors.	8-6.7
<u>BRODE LOAD</u>	Defines the air overpressure load parameters	8-6.8

TABLE 8-1 (Cont'd)

PROSSOR Directives

Directive	Function	Section
<u>TIME INCREMENTS</u> *	Defines the number of major time increments to be processed.	8-8.1

COMBINE Directives

<u>PRINT OPTIONS</u>	Requests output of various quantities.	8-10.1
<u>COPY RECORDS</u> *	Specifies how results output files are to be combined.	8-10.2

TRNSPOS Directives

<u>PRINT OPTIONS</u>	Requests output of various quantities.	8-12.1
<u>TRANSPOSE</u> *	Specifies how the combined results file is to be transposed.	8-12.2

* Directive must be on the input file.

TABLE 8-2
DIRECTIVE ABBREVIATIONS

START UP Directives

Directive	Abbreviation
TITLE	TI
IDENTIFIER	ID
PRINT OPTIONS	PO
ZONE DEFINITION	ZD
ZONE TYPE	ZT
MATERIALS	MA
STABILITY	ST
DAMPING	DA
REGION MATERIALS	RM
REGION COORDS	RC

ALTER FL Directives

ALTER STABILITY	AS
ALTER DAMPING	AD
ALTER ELEMENT	AE
ALTER NODE	AN

TABLE S-2 (Cont'd)

CLEAN UP Directives

Directive	Abbreviation
PRINT OPTIONS	PO
TIME DATA	TD
SUBCYCLE RATIOS	SR
ZONE SUBCYCLES	ZS
BOUNDARY CONDITIONS	BC
OUTPUT	OU
CORNER FACTOR	CF
BRODE LOAD	BL

PROSSOR Directives

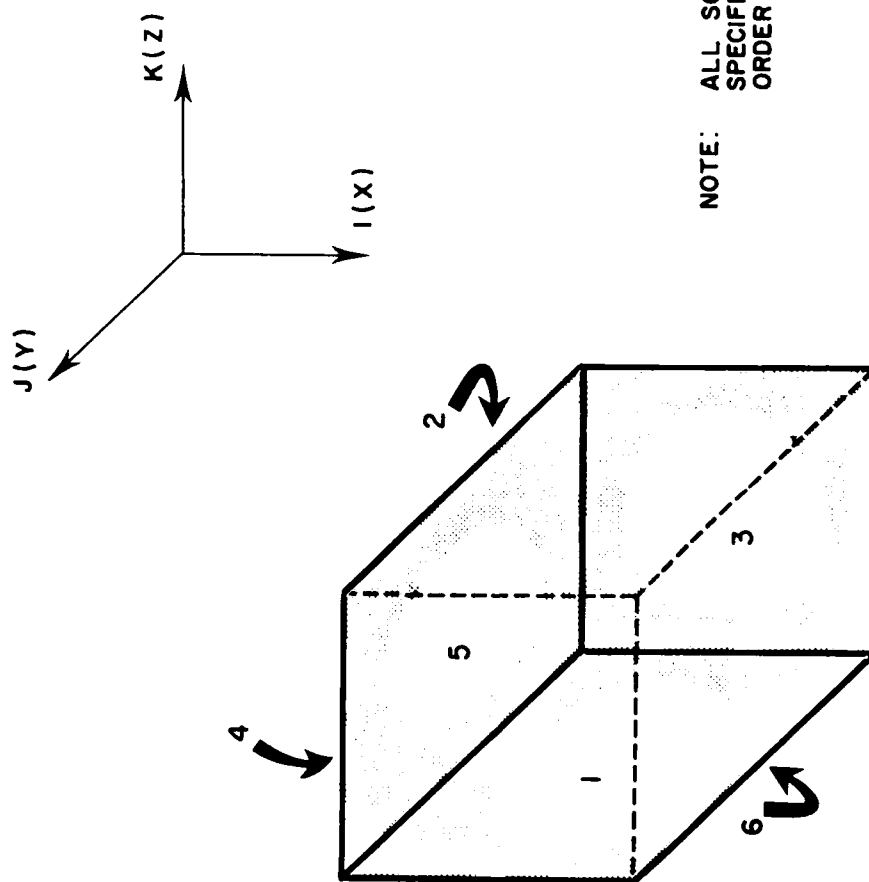
TIME INCREMENTS	TI
-----------------	----

COMBINE Directives

PRINT OPTIONS	PO
COPY RECORDS	CR

TRANSPOS Directives

PRINT OPTIONS	PO
TRANSPOSE	TR



NOTE: ALL SOIL ISLAND SURFACE INPUT SPECIFICATIONS ARE GIVEN IN THE ORDER OF THE NUMBERING SCHEME

FIG. 8-1 SOIL ISLAND SURFACE - NUMBERING SCHEME

SECTION 9

OUTPUT FILES

The results output file (tape 3) for the CLEAN UP processor and the first execution of PROSSOR must be the same file. Each subsequent execution of PROSSOR produces a separate (secondary) results output file (tape 3).

The traditional theory of elasticity sign conventions apply: velocities are positive in the positive coordinate directions; normal stresses are positive in tension; normal strains are positive in extension.

9-1 Output Data Structure (Tables 9-1 and 9-2)

The CLEAN UP processor writes the problem title (eight words) as the first header record on the primary (tape 3) file. The CLEAN UP processor also writes the second and third header records. The second header record contains the following seven words:

- number of velocity components output,
- number of velocity output stations,
- number of stress components output,
- number of stress output stations,
- number of strain components output,
- number of strain output stations, and
- number of data records on the primary output file.

The third record is composed of the global I, J, K node (element) numbers, the zone number, the zone node (element) number, and the global x, y, z coordinates of each velocity (stress, strain) output station. The number of words in the third header record is equal to eight times the total number of output stations.

Each time processor writes an identification record on its tape 3. The first time processor writes its identification record as the fourth record on the primary result output file. Each succeeding time processor writes its identification record as the first record on the corresponding secondary result output file. The identification record is made up of the following seven words:

- a one word problem identifier,
- processor execution date,
- processor execution time,
- processor number,
- number of major time steps processed,
- processor integration start time, and
- processor integration end time.

On each tape 3 file the processor identification record is followed by the output data records: one record per major time step. The output data records are composed of the major time and the requested results at the specified output stations, respectfully. During a major time step the data record is built in a SCM buffer. The MT for the MDT is placed in the first word of the buffer. Results from output stations associated with a given zone are placed in the buffer when processing of the zone has been completed within the MDT. After the processing of all zones has been completed in the MDT, the data record is written on the result output file.

9-2 Combined Output File (Table 9-3)

The primary and secondary results output files from the individual time processors must be combined into one file before any post-processing of the output data can be performed. The record configuration of the merged file is similar to the record structure of the primary results output file. The combined file contains three header records, a time processor information record and the merged output data records.

The first header record is the eight word problem title: same as first record on the primary output file. The second record contains the following nine words:

the first six words from the second record on the primary output file,
the number of output data records combined,
the number of output files combined, and
the number of file identification words per combined file.

The third header record is a copy of the third record on the primary file.

The time processor information record (record four) consists of the problem identifier word followed by the second, third, fifth, sixth and seventh words from the time processor identification record on each output data file combined. The problem identifier word is the first word on a time processor identification record and must be the same word for each output data file. If the time step or integration time information in the time processor identification record on a results output file is not consistent with the data records to be merged, it is corrected before being placed in the time processor information record for the combined output file.

9-3 Transposed Output File (Table 9-4)

The transposed output file is made up of three header records, a record of transposed time history record numbers, a time array record, and the time history records transposed from the data records on the combined output file.

The first header record is the eight word problem title record. The second record is composed of the following ten words:

the first six words from the second header record on the combined output file,
the length of each time history array,
the number of time histories transposed,
the number of words in the time array header, and
the number of words in each time history header.

The third header record is a copy of the third record on the combined output file: the output station identification record.

The record numbers of the time histories transposed is the fourth record. A time history record number is the data sequence number associated with the output station identification record. That is, the history record number is the location minus one of the original data in the output data records on the combined output file. Since output time is the first word in the output data records, the time array has record number zero.

The fifth record is the time array. The time array consists of a six word header and the output times. The header contains the following information:

- number of words in header (6),
- time array record number (always zero for time array),
- length of time array
- Hollerith identifier (10HHSTRY TIME),
- first output time, and
- last output time.

The time history records follow the time array record. The time history records are composed of a 14 word header and an output data time history array. Each 14 word header is made up of the following:

- number of words in header (14),
- time history record number,
- length of history array,
- Hollerith identifier,
- output station I number,
- output station J number,
- output station K number,
- output station x coordinate,
- output station y coordinate,
- output station z coordinate,
- time history minimum,
- time history maximum,
- integrated history minimum, and
- integrated history maximum.

The output station I, J, K numbers are global node/element numbers. The output station x, y, z coordinates are global node/element-centroid coordinates. The Hollerith identifier is one of the following fifteen words:

- 10HX VELOCITY, 10HY VELOCITY, 10HZ VELOCITY,
- 10HXX STRESS , 10HYY STRESS , 10HZZ STRESS ,
- 10HXY STRESS , 10HYZ STRESS , 10HZX STRESS ,
- 10HXX STRAIN , 10HYY STRAIN , 10HZZ STRAIN ,
- 10HXY STRAIN , 10HYZ STRAIN , 10HZX STRAIN .

TABLE 9-1

PRIMARY RESULTS OUTPUT FILE (Tape 3)

Record Number	Number of Words	Record Description
1	8	Problem Title
2	7	Number of velocity components (NVEL CP) Number of velocity stations (NVEL ST) Number of stress components (NSIG CP) Number of stress station (NSIG ST) Number of strain components (NEPS CP) Number of strain stations (NEPS ST) Number of major time steps on primary output file
3	$8 \times \text{NTTL ST}^\dagger$	Output Station Identification: I, J, K global node/element numbers Zone number Zone node/element number x, y, z global coordinates (8 words per output station)
4	7	Time Processor Identification: Hollerith problem identifier Processor execution date Processor execution time Processor number Number of major time steps (NMTS) Integration starting time Integration termination time
5 thru 4 + NMTS	1 + Data Length*	Results Output Data Records: Major time Velocity, stress and strain output data (Arranged in accordance with output station order in record 3)

$\dagger$ NTTL ST = NVEL ST + NSIG ST + NEPS ST

* Data Length = NVEL CP x NVEL ST + NSIG CP x NSIG ST + NEPS CP x NEPS ST

TABLE 9-2

SECONDARY RESULTS OUTPUT FILE (Tape 3)

Record Number	Number of Words	Record Description
1	7	Time Processor Identification: Hollerith problem identifier (same as primary file) (All other parameters reflect the status of the secondary file)
2 thru 1 + NMTS (NMTS for secondary file)	1 + Data Length* (Same as primary file)	Results Output Data Records: (Same word configuration as primary file)

* Data Length = NVEL CP x NVEL ST + NSIG CP x NSIG ST + NEPS CP x NEPS ST

TABLE 9-3

COMBINED RESULTS OUTPUT FILE (Tape 70)

Record Number	Number of Words	Record Description
1	8	Problem Title (Same as primary file)
2	9	NVEL CP, NVEL ST NSIG CP, NSIG ST NEPS CP, NEPS ST (First 6 words from record 2 on primary file) Number of data records combined (NDRC) Number of results output files combined (NCF) Number of file identification words/file (NFIW)
3	8 x NTTL ST [†]	Output Station Identification: (Same as record 3 on primary file)
4	1 + NCF x NFIW	Time Processor Information: Hollerith problem identifier (same as primary file) Processor execution date Processor execution time NMTS Integration starting time Integration termination time (NFIW words per file combined)
5 thru 4 + NDRC	1 + Data Length* (Same as primary file)	Combined Results Output Data Records: (Same word configuration as primary file)

[†] NTTL ST = NVEL ST + NSIG ST + NEPS ST

* Data Length = NVEL CP x NVEL ST + NSIG CP x NSIG ST + NEPS CP x NEPS ST

TABLE 9-4

TRANSPPOSED RESULTS OUTPUT FILE (Tape 8)

Record Number	Number of Words	Record Description
1	8	Problem Title (same as combined file)
2	10	NVEL CP, NVEL ST } NSIG CP, NSIG ST } (First 6 words from record 2 on combined file) NEPS CP, NEPS ST } Length of each time history array (LTHA) Number of time histories transposed (NTHT) Number of words in time array header (NTAH) Number of words in each time history header (NTHH)
3	8 x NTTL ST [*]	Output Station Identification: (Same as record 3 on combined file)
4	1 + NTHT	Record numbers of time histories transposed: (First record number is zero corresponding to the time array)
5	NTAH + LTHA	Output time array: NTAH (6) } Time array record number (0) } Number of output times (LTHA) } Hollerith identifier (10HISTORY TIME) } First output time } Last output time } Output times (LTHA words) } Time array header (NTAH words)
6 thru 5 + NTHT	NTHH + LTHA	Time history records: NTHH (14) } Time history record number } Length of time history array (LTHA) } Hollerith identifier } Output station I, J, K numbers } Output station x, y, z coordinates } Time history minimum } Time history maximum } Integrated history minimum } Integrated history maximum } Time history data array (LTHA words) } Time history array header (NTHH words)

SECTION 10

GRAPHICS

10-1 Structural Model Plotting (TGS3D)

This section describes the TRANAL GRAPHICS SUBSYSTEM 3-D MODEL PLOT PROCESSOR (TGS3D). It is intended for the TRANAL user who desires pictorial output of his structural model for two reasons. They are:

- 1) Visual inspection of model for input errors.
- 2) Generation of plots for reports.

10-1.1 Region and Zone Plotting

A structural complex being analyzed by TRANAL is broken up into workable substructural units, called zones. The zones are themselves subdivided into elements. During the data input phase of a TRANAL run, a physical subdivision of the structure known as regions is also available. Therefore it is possible to generate two types of 3-D plots. They are:

- 1) Pictures of the structural system detailed by regions.
- 2) Pictures of the structural system detailed by zones and elements.

10-1.2 Input

TGS3D is controlled by input directives which reside on the TRANAL input file (tape 1). The syntax of the TGS3D input directives is similar to the directives of the other TRANAL processors. Refer to Section 11 for details. However, it is important to note that the order of the TGS3D input directives is significant contrary to other TRANAL input directives. There are two reasons for this. First, TGS3D directives are generally repeated for a run to generate many plots. It is therefore necessary to associate together directives which refer to a particular plot. Second, the action specified by the appearance of a directive are usually called for immediately and not deferred. Because even for a single plot, directives may be repeated to yield a desired plot and the order of the actions is important. It would not do to rotate about the x axis and then the y when it is the y axis rotation that is desired to be performed first.

10-1.3 Coordinate Systems and Transformation

TRANAL accepts a model defined in a right-handed coordinate system. This is the input system, system I. (See figure 10-1a.)

Associated with a display device there is a coordinate system. This is system D, also known as the picture plane. As most computer graphics displays are 2-D, system D is only a 2-D system. (See figure 10-1b.)

In order to map the points of the TRANAL model from system I onto system D we define an eye coordinate system for convenience. This is system E with the eye of the observer of the model assumed to be at the origin. The direction of view is the direction of the positive Z_E axis. The X_E axis is vertical

and the Y_E axis is horizontal. (See figure 10-1c.) These assumptions dictate that the X_E and Y_E axes are to align with the Y_D and X_D axes of the display respectively.

The mapping of a model in the eye coordinate system to the display system is a projection. This is because system E is 3-D and system D is 2-D. The types of projections are orthogonal and perspective. In orthogonal projection, the Z_E component is simply discarded; the projection lines do not converge at the eye. (See figure 10-2a.) On the other hand, perspective projection divides the X_E and Y_E components by the Z_E component to give the illusion of depth on the 2-D display. In this case the projection lines converge at the eye. (See figure 10-2b.)

System I and E coincide with each other initially. Before any transformations the view obtained is that when the observer's eye is at the origin of the input system looking out in the direction of Z_I .

To take a different view of the model from another vantage point, one must reposition the model, defined in system I, relative to system E such that the Z_E axis points at the model as if the eye were looking at it from the same vantage point. This is accomplished through transformations.

The transformations to convert the model defined in system I to system E include rotation about, scale and translate along the x, y, or z axes.

All rotations are right-handed. That is, if the thumb of the right hand is pointed in the positive direction of the axis about which rotation is performed then the fingers curl in the direction of positive rotations. (See figure 10-3.)

One can also scale or magnify the model. This is a change in size. To maintain proportion the scale factors must be the same in all directions. If the x, y, and z scale factors are not equal then a stretching effect is realized.

A negative scale factor in the x, y, or z axis yields a reflection about the corresponding axis.

Finally one may translate the model. This can have the effects of scaling, shifting, or a combination of both. If only the z components of the model are changed then a scaling effect is realized because the model is being moved closer to or farther from the eye. A change in the x (y) components causes the model to be shifted from right (up) to left (down) on the plot. A change in all of the three components results in a combination of the above effects. All of the above transformations can be specified using the TRANSFORM directive. (Refer to Section 11-2.8.)

There is also a DIMENSION directive to scale the model. It is used to specify the extent of the model in the X_I , Y_I , and Z_I directions. By specifying an extent twice the actual extent of the model, an overall effect of scaling by 1/2 is realized. In addition, the DIMENSION directive can be used as a windowing option. This is the case when the limit parameters of the DIMENSION directive are within the real limits of the input model thus selecting a portion of the model for display. (See Section 11-2.2.)

10-1.4 Hidden-line Algorithm

The hidden-line problem deals with the removal of objects blocked from view by objects closer to the eye. The hidden-line removal capability of TGS3D is based on the Watkin's scan-line algorithm. In fact, TGS3D uses an adaptation of the Watkin's algorithm as implemented by Michael Archuleta of Lawrence Livermore Laboratory Ref. 29.

The algorithm was chosen because it is relatively fast and thus economical. The speed of the Watkin's technique comes from the examination of the projected image (with the Z_E component retained) on the picture plane rather than the object itself. The computational requirements of the algorithm grows proportional only to the visible complexity of the final picture and has an upper bound equal to the resolution of the display device. That is, it is not necessary to ask the hidden-line question beyond the limits of the display device.

In addition to speed, the algorithm was chosen because it will be simple to add capabilities like plotting in color and with contours. These add greatly to the information content of the resultant plots as well as being esthetically pleasing.

Basically the Watkin's technique is to divide the picture plane into a finite number of scan-lines. Within each scan-line the hidden-line question is asked. The hidden-line problem is greatly simplified as it now becomes a two dimensional problem. (See figure 10-4.)

The division of the picture plane into a finite number of scan-lines is not a severe limitation. As implemented the maximum divisions is 1024. In fact, most display terminals have the maximum capability of generating pictures on a 1024 x 1024 grid only. It is also easy to extend it if the need arises.

For further information on the algorithm, consult Newman and Sproull Ref. 30.

10-1.5 Files and System Dependent Routines

TGS3D input directives is read off the input file (tape 1). Non-graphics output goes on the system output file (tape 2). The plot file (tape 99) is reserved for graphics output. Input data are on files tape 11, tape 12, tape 13, tape 14, tape 32, tape 15 and tape 50.

For region plots it is necessary to have tape 11, tape 12, tape 13, tape 14 and tape 50 available. These contain region data such as material type and coordinates.

For zone element plots it is necessary to have files tape 32, tape 15, and tape 50 available. The element information file (tape 32) contains the element information such as the element material type; the node coordinates file (tape 15) contains the zone element node coordinates; the SCM common file (tape 50) contains the SCM common variables some of which define the zone element configuration.

All of the above data files must be available if both region and zone element plots are desired in a single run.

Besides the above, there is a plot file (tape 99). This file contains the information that is to drive the display device.

The information needed by a display device to generate the pictures comes from the TGS processors. This information is transformed from the device independent form to a device dependent form by device dependent routines associated with the display. This is the information that goes onto the plot file (tape 99).

There are no assumptions made about the display device used other than that it must be able to draw a straight line between two points. The TGS processors communicate line segment information to the device dependent routine, TGSLINE. TGSLINE must be modified for different installation and device setups

so that it performs the independent to dependent transformation on the line segment information.

In addition, display devices usually require initialization and termination. Initialization may include a specification of the account number, programmer name, what type of paper to use, etc. This is done in the routine TGSINIT.

Setting up TGSINIT also involves defining the size of the plot page. This is done by defining the variables SIZEX, SIZEY in labeled common /RESOLVE/ of TGSINIT.

The variable SIZEX (SIZEY) determine the limits of the x (y) coordinates for the display device. The values of the coordinates generated by TGS and passed to TGSLINE will be in the range 0. to SIZEX (0. to SIZEY) for the X_D (Y_D) direction.

Figure 10-5a shows a plot page on a Calcomp plotter of 10 inches by 10 inches. Figure 10-5b shows a plot page on a Tektronix storage tube terminal to be 780 rasters by 780 rasters. These are specified by setting SIZEX and SIZEY to 10.00 units for the Calcomp and 780 units for the Tektronix.

Termination is done in TGSTERM. This involves flushing all buffers used, closing the plot file, repositioning the pen, etc.

Shown in figure 10-6 is a listing of a simple set of device dependent routines TGSINIT, TGSLINE and TGSTERM for the Berkeley system using a Calcomp drum plotter. The routines assume the values of the size variables mentioned above.

10-2 Time History Plotting (TGSHIST)

This section describes the TRANAL GRAPHICS SUBSYSTEM TIME HISTORY GRAPHICS PROCESSOR (TGSHIST). It is intended for the TRANAL user who desires to render his TRANAL results into graphic form.

Any of the results such as velocities, stresses, and strains may be plotted. An option also allows the integral of the quantity selected (i.e. displacements, impulses) to be plotted. Similar quantities may even be cross-plotted with each other for comparison purposes.

The selection of which quantity goes where is by the use of the PLOT INFO directive. The options CURVE and TAPE select up to four different curve numbers (time histories) from up to four different input files. However, there are no requirements for the curves to reside on different input files.

The input files are the transposed results output files (type tape 8). On input to TGSHIST they are renamed as tape 91, tape 92, tape 93 and tape 94. For a description of the tape formats see the table, Transposed Results Output File.

The plot page is portioned as in figure 10-8. The TITLE directive defines the text that goes into the title slot. Both the Y label and X label slots come from the input files. The time interval, (t_1, t_2) , is specified by the TIME INTERVAL directive; or if not used, is automatically calculated. The Y interval, (y_1, y_2) , is defined by the PLOT INFO directive options MIN and MAX. Again, this

also calculated if not specified. The slot identified by key is reserved for information that is used to distinguish the various curves on a plot.

The plot output goes onto the plot file (tape 99) just as in TGS3D. The required setup for a particular installation and display device also follows a similar procedure as for TGS3D.

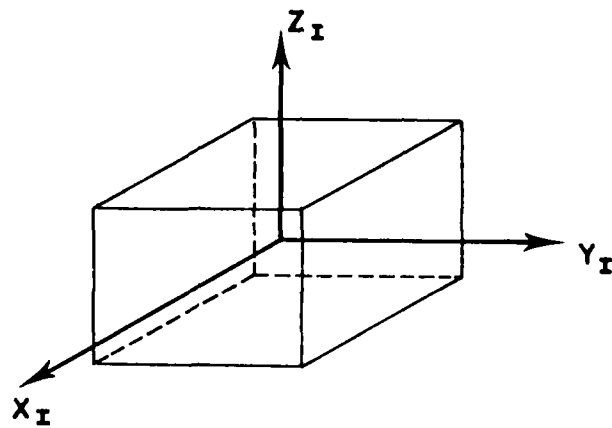


FIG. 10-1a TRANAL MODEL DEFINED IN INPUT SYSTEM, SYSTEM I

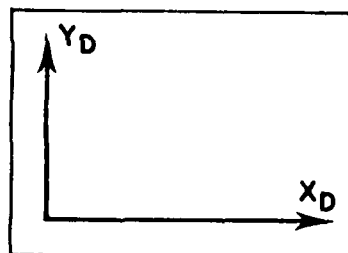


FIG. 10-1b DISPLAY DEVICE SYSTEM, SYSTEM D

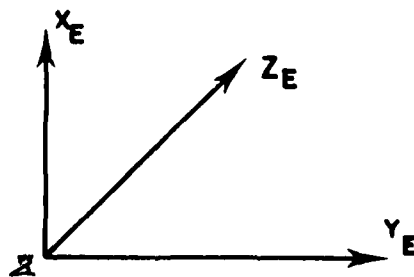


FIG. 10-1c THE EYE COORDINATE SYSTEM

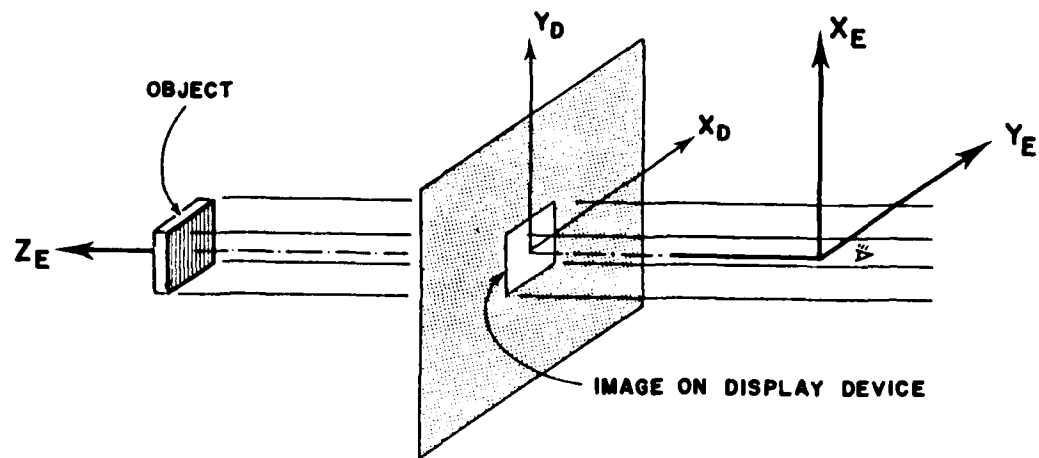


FIG. 10-2a ORTHOGONAL PROJECTION

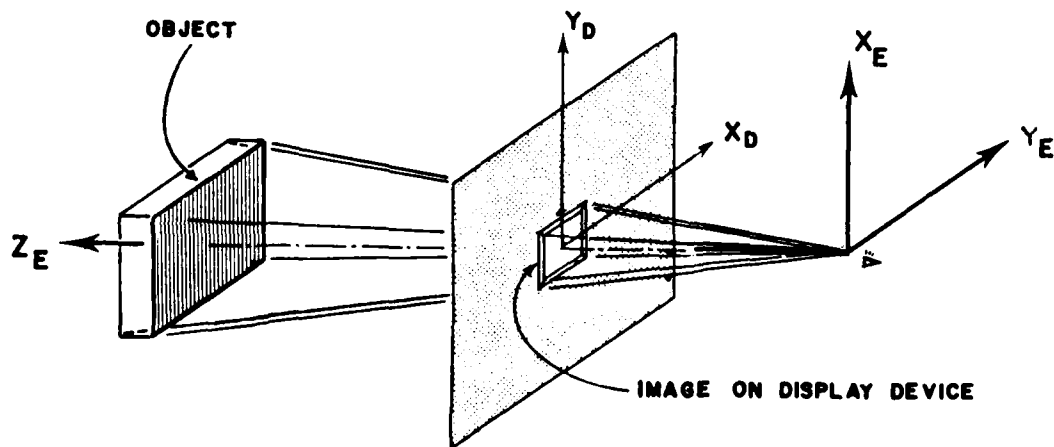


FIG. 10-2b PERSPECTIVE PROJECTION

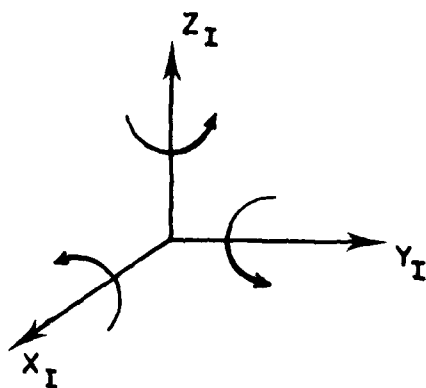


FIG. 10-3 RIGHT - HANDED ROTATION

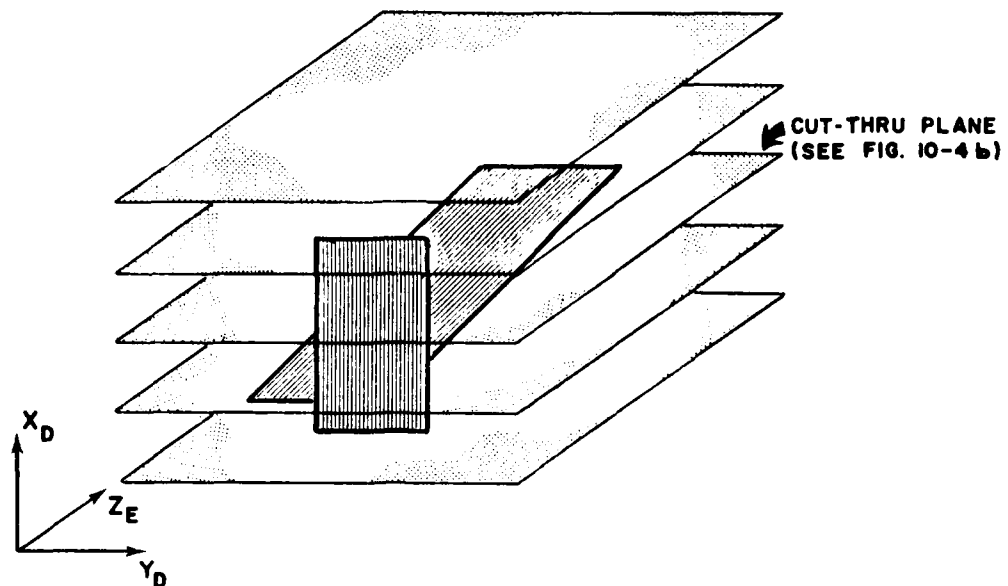


FIG. 10-4a PICTURE PLANE DIVIDED INTO SCAN-LINES

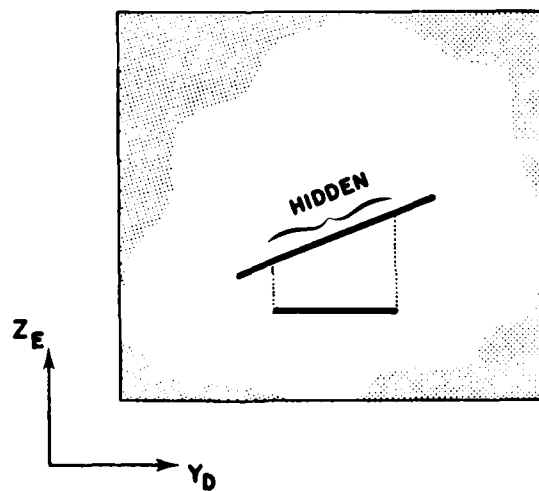


FIG. 10-4b SIMPLIFIED 2D PROBLEM

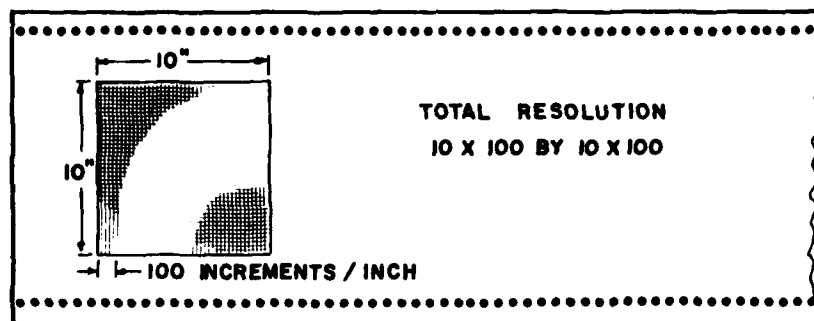


FIG. 10-5a A PLOT PAGE ON A COLCOMP PLOTTER

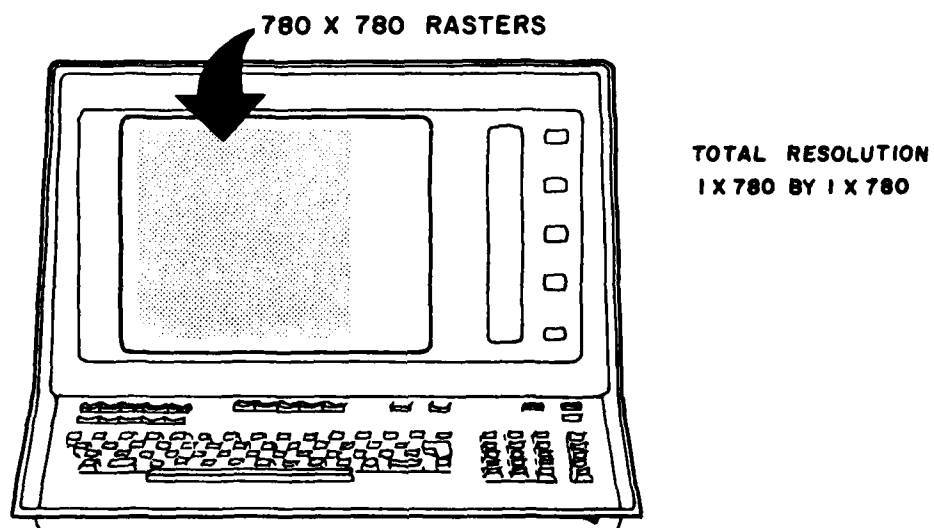


FIG. 10-5b A PLOT PAGE ON A TEKTRONIX TERMINAL

```

SUBROUTINE TGS INIT
C
C   THIS IS THE DEVICE INITIALIZATION ROUTINE
COMMON / RESOLVE / SIZEX, SIZEY
C
C   THE PLOT PAGE SIZE IS DEFINED
SIZEX = .SIZEY = 10.
C   THE PHYSICAL PAGE SIZE IS DEFINED
.SIZE = SIZEY + 1.
C   WE WANT BLANK PAPER FOR 3D PLOTS
PAPER = 10HBLANK
C   PLOT FILE IS TAPE99
LFNUM = 1 GT INT ( 10HPLOT FILE )
C   DONT CARE WHAT PEN TO USE
PENTYPE = 0.
C   NOW INITIALIZE
CALL PLOTS ( ISIZE, PAPER, LFNUM, PENTYPE )
RETURN
END

SUBROUTINE TGS LINE ( X1, Y1, X2, Y2 )
C
C   THIS ROUTINE SIMULATES THE LINE SEGMENT DRAW CAPABILITY.
C   IT FIRST MOVES THE PEN UP TO THE BEGINNING OF THE LINE SEGMENT
C   THEN FOLLOWS THAT BY A PEN DOWN AND A MOVE TO THE LINE SEGMENT
C   END POINT.
C
DATA IUP, IDOWN / 1, 2 /
CALL PLOT ( X1, Y1, IUP )
CALL PLOT ( X2, Y2, IDOWN )
RETURN
END

SUBROUTINE TGS TERM
C
C   THIS ROUTINE DOES THE TERMINATION FUNCTION
C
DATA ITERM / 40 /
C   ITERM OF 40 SPECIFIES A FLUSH OF ALL BUFFERS AND A TERMINATION
C   OF THE PLOT SESSION.
CALL PLOT ( 0., 0., ITERM )
RETURN
END

```

FIGURE 10 - 6 A SIMPLE SET OF DEVICE DEPENDENT ROUTINES

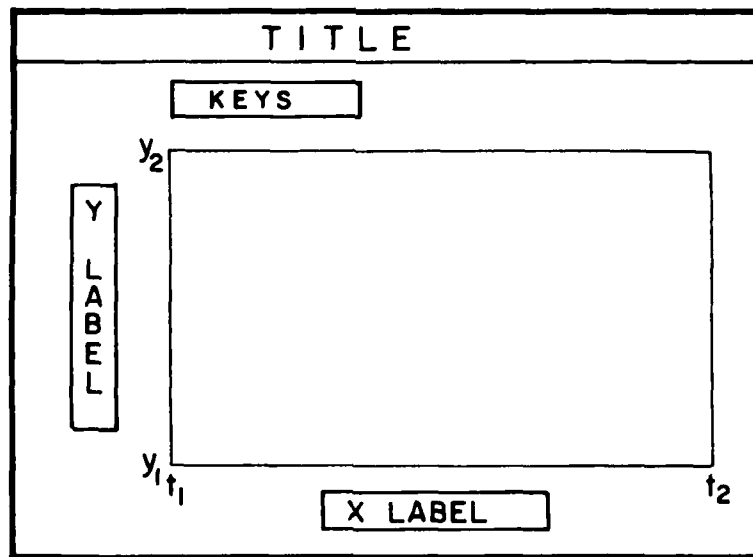


FIG. 10-7 LAYOUT OF A PLOT PAGE FOR TIME HISTORY GRAPHS

SECTION 11

PLOTTING PROCESSORS AND INPUT

11-1 TGS3D Processor

The TGS3D processor generates 3-D plots of the input model. The following files are required:

Input	-	tape 1.
System output	-	tape 2.*).
Node coordinates	-	tape 15*).
Element information	-	tape 32*).
SCM common	-	tape 50 .
Region data	-	tape 11, tape 12, tape 13, tape 14**).
Plot file	-	tape 99.
Work file	-	tape 98.

11-2 TGS3D Input Directives

The following directives are used to specify and manipulate the structural model to be displayed. Most of these directives have a global effect in the sense that once invoked they remain in effect until respecified. For example, if several different views of the model chosen by the OBJECT directive are desired, the OBJECT directive does not need to be used more than once. Only the TRANSFORM directive needs to be respecified to yield the different views. The exception is the DISPLAY directive. It must be used every time to cause plotting of a picture.

The order of the directives on input is also important because these directives cause immediate actions and may not be commutative.

11-2.1 AXIS

In addition to a plot of the structural model, the AXIS directive may be used to request a plot of the input coordinate system, system I. System I is the coordinate system of the model definition. The system is transformed just as the model is transformed. A plot of it is represented by its coordinate axes. Either one or more of the axes X_I , Y_I , or Z_I may be selected.

Syntax

```
$|*  AXIS  < coord > | ALL
< coord > is X | Y | Z
ALL      selects all three axes  $X_I$ ,  $Y_I$ ,  $Z_I$  .
```

Default

If the directive AXIS is not used then none of the system I axes are drawn.

Example

```
$ AXIS X Y Z
```

*) Only if zone and element plots are selected.

***) Only if region plots are selected.

11-2.2 DIMENSION

The DIMENSION directive is used to specify the dimension of the object selected to be plot. This may differ from the real object dimension.

Syntax - keyword format

```
$|* DIMENSION  X1 < xmin > Y1 < ymin > Z1 < zmin >  
                X2 < xmax > Y2 < ymax > Z2 < zmax >
```

```
< xmin > }  
< ymin > }      Object lower limits (real)  
< zmin > }  
< xmax > }  
< ymax > }      Object upper limits (real)  
< zmax > }
```

This causes the dimension of the object to be $|X2 - X1|$, $|Y2 - Y1|$, and $|Z2 - Z1|$ in the X_1 , Y_1 , and Z_1 directions respectively. The center of the object is also shifted to $\frac{1}{2}((X1 + X2), (Y1 + Y2), (Z1 + Z2))$.

Default

The default dimension is the actual object dimension as calculated from input. The center of the object is unchanged.

Example

```
$  DIMENSION  X1 = 0.    Y1 = 0.    Z1 = 0.  
                X2 = 300. Y2 = 300. Z2 = 300.
```

11-2.3 DISPLAY

The DISPLAY directive requests the immediate action of plotting the object selected by the OBJECT directive, any axes selected by the AXIS directive, and any titling information on a TITLE directive.

Syntax

```
$|* DISPLAY
```

Default

There is no default option. This is a required directive.

Example

```
* DISPLAY
```

11-2.4 MATERIAL OPTION

The MATERIAL OPTION directive is used to select from the object already chosen that portion of the object having materials as specified by the material options.

Syntax

\$|* MATERIAL OPTION [ONLY | EXCEPT]₁¹ [< material >]₁[∞]

< material > is a material name (label).

ONLY causes that portion of the object with the materials [< material >]₁[∞] to be plot. The rest of the object

is ignored.

EXCEPT causes that portion of the object not with the materials [< material >]₁[∞] to be plot.

There is a special material known as VOID which is always ignored. However, for region plots it is possible to plot a VOID region by using the ONLY option with VOID as material. This is the only exception.

Example

\$ MATERIAL OPTION ONLY SAND CONCRETE

11-2.5 PRINT OPTION

The PRINT OPTION directive lets the user select the quantity of print output. During the course of a plot session, it is sometimes reassuring to obtain intermediate output which reflect the actions specified by other input directives. The above directive allows the intermediate output not to be printed (NONE), some of them to be printed (PARTIAL), or all of them to be printed (FULL).

Syntax

\$|* PRINT OPTION [NONE | PARTIAL | FULL]₁¹

NONE causes no intermediate information on the output file. However, errors will still be printed.

PARTIAL gives intermediate results such as calculated dimensions, default values, etc. Errors are also available.

FULL gives extensive detailed information including object coordinates, and data manager tables. This is suitable in debug sessions. Again, errors are available.

Default

If the directive is not used, the PARTIAL option is selected.

Example

* PRINT OPTION NONE

11-2.6 OBJECT

Two categories of pictures can be generated. They are:

- 1) The structural model detailed by regions.
- 2) The structural model detailed by zones and elements.

The OBJECT directive enables the selection of one of the two above categories of pictures. In addition, it may specify only a portion of the entire input model.

Syntax

```
$|* OBJECT [ REGION [ < ijk > ]3 FACE [ < face > ]16 |  
              ZONE [ < ijk > ]3 ELEMENT [ < ijk > ]3  
              FACE [ < face > ]16 ]1
```

REGION	specifies the regions to be plotted.
ZONE	specifies the zones to be plotted.
ELEMENT	specifies the elements of the specified zones to be plotted.
FACE	specifies which faces of the specified region or zone element are to be plotted.
< ijk >	is I < range i > J < range j > K < range k > (range).
< face >	is 1 2 3 4 5 6 (integer).

Default

This directive must be fully specified. However, a zero may appear in place of a < ijk > range specification to represent all of the regions, zones, or elements. See figure 11-1 for the region, zone, element, and face numbering schemes.

Example

```
$ OBJECT REGION I 1 TO 10 BY 2 J 0 K 0  
          FACE 2 6
```

11-2.7 TITLE

The TITLE directive is used for identification. It associates descriptive information with the plot and print outputs.

Syntax

```
$|* TITLE < name >  
< name > is a description enclosed in single quotation marks  
          As many as 78 characters may be used (label).
```

Default

If the directive is omitted, a blank title is used.

Example

```
$ TITLE 'TAB/S STRUCTURE REGION PREVIEW'
```

11-2.8 TRANSFORM

The TRANSFORM directive is used to specify the transformation of the object chosen from the input coordinate system (system I) to the eye system (system E). The transformation may be composed of several simple transformations. These simple transformations may all be on one TRANSFORM directive or distributed over several TRANSFORM directives. It is important to sequence them correctly however, as transformations are not generally commutative.

Syntax

\$[* TRANSFORM [< option >]₁[∞]

< option > is one of the following simple transformations:

SCALE < x factor > < y factor > < z factor >

< x factor > x direction scale factor (real).

< y factor > y direction scale factor (real).

< z factor > z direction scale factor (real).

ROTATE [[X | Y | Z]₁¹ < angle >]₁[∞]

< angle > The ROTATE option causes the object chosen to be rotated about the x, y, or z axis
< angle > degrees (real).

TRANSLATE < x value > < y value > < z value >

< x value > x direction translation term (real).

< y value > y direction translation term (real).

< z value > z direction translation term (real).

RESET The RESET option causes all previously specified transformations to be voided. The object reverts to its defined input position. If this option is not used then the previously specified transformation remains in effect. Thus one need not respecify ever the same transformation as the previous one.

VIEW VECTOR X1 < x eye > Y1 < y eye > Z1 < z eye >
X2 < x look > Y2 < y look > Z2 < z look >

X1 through Z2 are of keyword format.

< x eye > }
< y eye > } coordinates of the eye (real).
< z eye > }

< x look > }
< y look > } coordinates of the point the eye
< z look > } looking at (real).

AD-A084 654

WEIDLINGER ASSOCIATES NEW YORK

F/8 9/2

TRANAL USER'S GUIDE, PART 1. (SMALL STRAIN, SMALL DISPLACEMENT —ETC(U)

MAR 79 J L BAYLOR, J P WRIGHT, C F CHUNG

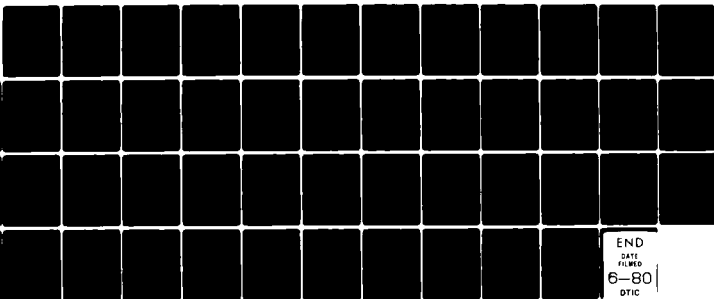
DNA001-76-C-0125

UNCLASSIFIED

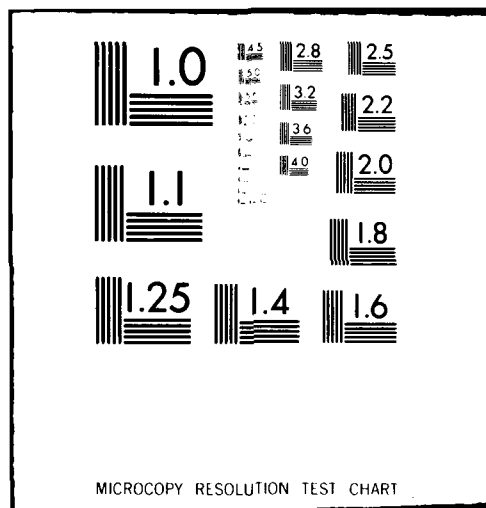
DNA-4960F

NL

2 of 2
ADA
1984054



END
DATE
FILMED
6-80
DTIC



This option directs TGS3D to calculate the rotations and translation necessary to arrive at the state with the eye at (X1, Y1, Z1) and looking at (X2, Y2, Z2).

Default

If no transformations are specified then the object chosen will be displayed as if it were already in the eye system (system E), i.e., the X_I , Y_I and Z_I axes become the X_E , Y_E , and Z_E axes, respectively.

But once a transformation is specified it remains in effect until RESET is used.

The default value is 0. for any component not specified in the keyword format VIEW VECTOR option.

Example

```
$ TRANSFORM
    VIEW VECTOR  -10.      -10.      -10.,
                  0.,       0.,       0.
    ROTATE       Z   90.
```

Error Messages

```
****  INVALID VIEW VECTOR
       The view vector specified has zero length.
```

11-2.9 VIEW OPTION

The directive VIEW OPTION is used to select whether it is the orthogonal projection or perspective projection that is to be used to transform the object from system E to system D. In addition, whether objects blocked from view by objects closer to the eye should be hidden or divulged is also specified with this directive.

Syntax

```
$|* VIEW OPTION [ PERSPECTIV | ORTHOGONAL ]1  
    [ HIDE | DIVULGE ]1
```

Default

If this directive is not used then it is assumed that ORTHOGONAL and DIVULGE were chosen.

Example

```
* VIEW OPTION PERSPECTIV HIDE
```

11-3 TGSHIST Processor

The TGSHIST processor generates time history plots of resultant quantities, i.e. velocities, stresses, and strains.

The following files are required:

Input	-	tape 1.
System output	-	tape 2.
Transposed result output	-	tape 91, tape 92, tape 93, tape 94.
Temporary	-	tape 97.
Plot	-	tape 99.

11-4 TGSHIST Input Directives

The following directives are used to specify which of the time histories on the transposed result files are to be graphed.

11-4.1 PLOT INFO

The PLOT INFO directive specifies the number of plots for this graphing session, the number of input tapes available, and whether the integral option is desired. In addition, the curves that are to be plot and their corresponding input tapes are specified.

Syntax

```
$|* PLOT INFO NOPLOT < plots > NOTAPE < tapes > INTEGRATE < int >  
    [ CURVES [ < curveno > ]14 TAPES [ < tapeno > ]14 { MINMAX < ymin >  
    < ymax > } ]1nl
```

NOPLOT, NOTAPE, and INTEGRATE are of keyword format.

< plots >	indicates the number of plots (integer).
< tapes >	indicates the number of input tapes. The maximum is four (4) (integer).
< int >	is a 0 or 1. 0 means integral option not desired. 1 means integral option desired.
< curveno >	indicates the curves to be plot (integer).
< tapeno >	represents the input tapes from which the curves are to be read in and plot. It may be 1 for tape 91, 2 for tape 92, etc. (integer).
< ymin > < ymax > nl	} minimum and maximum vertical axis scales. is the number of plots; the value of < plots > .

Default

This is a required directive.

Example

```
$ PLOT INFO NOPLOT = 2 NOTAPE = 4  
    curves 1 1 1 1 tapes 1 2 3 4  
    curves 2 2 2 2 tapes 1 2 3 4
```

11-4.2 PRINT OPTION

The PRINT OPTION directive lets the user select the quantity of print output.

Syntax

```
$|* PRINT OPTION [ NONE | PARTIAL | FULL ]11
```

For more information refer to section 11-2.5.

11-4.3 TIME INTERVAL

The TIME INTERVAL directive is used to select the time interval for all plots.

Syntax

```
$|* TIME INTERVAL < time 1 > < time 2 >
```

< time 1 > is the start of the time axis on plots (real).
< time 2 > is the end of the time axis on plots (real).

Default

If not used, the time interval is calculated such that it encompasses all time histories on the input files.

Example

```
$ TIME INTERVAL 0.0 10.0
```

11-4.4 TITLE

The TITLE directive is used for identification purposes. In particular, it identifies the plots.

Syntax

```
$|* TITLE < name >
```

Refer to section 11-2.7 for further information.

TABLE 11-1
PLOTING PROCESSOR INPUT DIRECTIVES
TGS3D DIRECTIVES

Directives	Function	Section
<u>AXIS</u>	Selects any, all or none of the input coordinate system axes to display.	11-2.1
<u>DIMENSION</u>	Specifies the extent of the object to be displayed.	11-2.2
<u>DISPLAY</u> *	Plot the object selected.	11-2.3
<u>MATERIAL OPTION</u>	Qualifies the portion of the object choosen to be displayed.	11-2.4
<u>PRINT OPTION</u>	Used to select the quantity of print output.	11-2.5
<u>OBJECT</u> *	Selects the regions, or zones and elements to plot.	11-2.6
<u>TITLE</u>	Used to identify the print and plot outputs produced.	11-2.7
<u>TRANSFORM</u>	Used to transform the object choosen from the input system (I) to the eye system (E).	11-2.8
<u>VIEW OPTION</u>	Used to select the manner of the mapping from the eye system (E) to the computer display device.	11-2.9

TGSHIST DIRECTIVES

<u>PLOT INFO</u> *	Specifies which time history is to be plot.	11-4.1
<u>PRINT OPTION</u>	Specifies the quantity of print output.	11-4.2
<u>TIME INTERVAL</u> *	Specifies the time interval for the history plot.	11-4.3
<u>TITLE</u>	Identifies outputs from this plot session.	11-4.4

* Directive must be on the input file.

TABLE 11-2
PLOTING DIRECTIVE ABBREVIATIONS
TGS3D DIRECTIVES

Directive	Abbreviation
AXIS	AX
DIMENSION	DM
DISPLAY	DS
MATERIAL OPTION	MO
PRINT OPTION	PO
OBJECT	OB
TITLE	TI
TRANSFORM	TR
VIEW OPTION	VO

TGSHIST DIRECTIVES

PLOT INFO	PI
PRINT OPTION	PO
TIME INTERVAL	TM
TITLE	TI

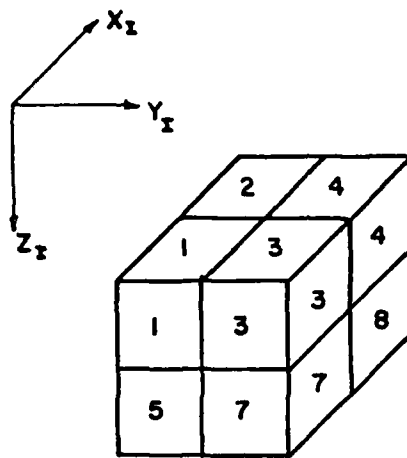


FIG. II-1a REGION, ZONE AND ELEMENT NUMBERING SCHEME

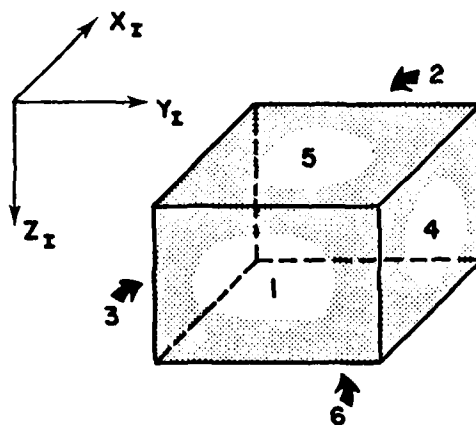


FIG. II-1b FACES OF AN ELEMENT OR REGION

SECTION 12

RUNNING A TRANAL JOB12-1 Introduction

Each TRANAL processor produces data files and/or modifies existing data files. The data files associated with each processor is given in Table 12-1. In order to run a processor one must (a) produce the binary object code driver program file, (b) supply all requisite library files, and (c) provide the necessary data files. After the processor has been executed, the data files should be saved on a permanent storage device. The processors may be executed one per job or several may be executed sequentially in a job.

To illustrate the methodology of running a TRANAL job, a problem analogous to one of the original TRANAL verification problems is used. The illustrative sample problem is similar to problem P1 of Ref. 4. It is a six foot square clay over shale (one dimensional) column loaded by an air overpressure shock on the (clay) top. Even though the sample problem is one dimensional, TRANAL uses three dimensional elements to model the column. Consequently, the lateral dimensions of the column must be selected so that the stability condition is satisfied by the user specified time increment. Otherwise, roundoff errors enhanced by lateral wave propagation effects can build up and eventually dominate the solution. (This is also true for two dimensional problems.) This is the basis for selecting a six foot by six foot cross section for the column. The physical and element configurations, the load, the uniaxial compression curves, and the cap model parameters are shown in figure 12-1.

The ALTER FL processor is not used. The ALTER FL processor is only required when the element information file and/or the node coordinate file are to be modified. The execution of the ALTER FL processor should be obvious from the illustrative problem and the input directives description in Section 8-4.

The records of an execution deck (file INPUT) are separated by end of records cards (cards with 7, 8, 9 in the first column). The deck is ended by an end of file card (a card with 6, 7, 8, 9 in the first column). The control cards must be the first record on file INPUT. The functional characteristics of a control card are only discussed when the card first appears in Tables 12-2 through 12-9 (processor execution decks). However, files associated with a control card are discussed whenever clarification is necessary. The numbering on the right-hand side of the control cards in the tables is only for identification purposes, and does not appear on the actual cards. These control cards are for the CDC 7600 computer facility of Lawrence Berkeley Laboratory (LBL) at Berkeley, California. To obtain detailed information on a control card, consult the Primary Reference Index for the PSS Library Handbook on the Berkeley system. Control card streams equivalent to the ones presented in Tables 12-2 through 12-9 can be formulated for other CDC computer installations.

A listing of the source deck for the file TRNLDIR, which contains the processor driver programs and UPDATE directives to file TRNLPL for the sample problem, is given in Table 12-10 (see Appendix D).

12-2 START UP Execution

The execution deck for the START UP processor is given in Table 12-2. The driver program and UPDATE directives are identified by the \$DECK, STARTUP card in Table 12-10.

12-2.1 Control Cards (Record 1)

JOB CARD (Card 1): The JOB CARD must be the first card of the deck. It contains information related to job identification and facility resources required. Each user must supply an appropriate JOB CARD.

*7,PSS (Card 2): This card must follow the JOB CARD control card. It specifies that the job is to be executed on the CDC 7600, and that the data cells (PSS) permanent storage devices are required.

COPY(INPUT,IR,TAPE1/BR) (Card 3): One record is copied from file INPUT (the execution deck) to file TAPE1, and TAPE1 is rewound after the copy. In this case the second record (the processor input data) is copied.

FETCHPS(TRANAL,OLDPL,TPL1) (Card 4): The subset file TPL1 of library TRANAL on the PSS is retrieved, and written on file OLDPL. TPL1 is the UPDATE program library which contains the random file I/O routines, and the routines related to the dynamic allocation of LCM and SCM.

UPDATE(F,L=1) (Card 5): The file OLDPL is modified according to directives on the next record on file INPUT (record 3), and the resulting file is written on file COMPILE. OLDPL, INPUT, and COMPILE are the default program library, input, and compile output files, respectively. For detailed information on UPDATE and the argument option parameters consult the CDC UPDATE Reference Manual.

RETURN(OLDPL) (Card 6): The file OLDPL is removed from the system (both file space and name).

FTN4(I,B=TBL1,L=0) (Card 7): A compiled binary object code image of the source file COMPILE (from UPDATE) is produced and written on file TBL1. For detailed information on FTN4 and the argument option parameters consult the Users' Guide to the FTN4 FORTRAN Compiler on the BKY 7600.

Card 9: TRNLDIR is the UPDATE program library (\$ master control character) which contains the processor driver programs and UPDATE directives to file TRNLPL.

Card 10: UPDATE directives for the START UP processor are written on file DRCTVS.

Card 12: TRNLPL is the UPDATE program library which contains all the TRANAL routines.

Card 13: The directives for this UPDATE are on the file DRCTVS. The driver program for the START UP processor, and all modified TRANAL routines are written to file COMPILE. The driver program must be the first program unit on COMPILE. It must be immediately followed by the TRANAL routine AAA AAA.

Card 16: The compiled binary object code image of the input source file COMPILE is written on the file PRGMBL.

Card 18: TRNLLIB is the compiled binary object code image in system library format of the file TRNLPL. TRNLLIB is stored on PSS with record and file marks removed.

SCATTER(TRNLLIB) (Card 19): The record and file marks are restored to TRNLLIB and the expanded file is written on file LIBRARY.

LINK(X,F=PRGMBL,F=TBL1,P=LIBRARY,CS=ID,RF) (Card 21): The F files are loaded in the order in which they appear (PRGMBL, TBL1). External references are satisfied (i.e. loaded) from P files (LIBRARY) and system library files. SCM variables and arrays are initialized to indefinites. All F and P files are returned when LINK is through with them. The program (processor) is then executed.

Labeled common blocks are loaded ahead of the routine in which they first appear. The routine AAA AAA contains all global labeled common blocks dimensioned to their maximum lengths. To prevent truncation of any global labeled common block, they should all be loaded with AAA AAA. To this end, AAA AAA is always loaded immediately after the driver program. So that the driver program is always the first program unit loaded, PRGMBL must be the first F file on the LINK card.

SAVELCM(TAPE60) (Card 23): The user's LCM data space (the random files) is written on file TAPE60.

STOTAPE(TAPE50=EXAMPLE/STARTUP/TAPE50,TAPENO) (Card 24): File TAPE50 is written on the GSS tape (assumed to already exist) with LBL tape library number TAPENO. File TAPE50 is written as data set TAPE50 with the directory name path EXAMPLE/STARTUP/.

EXIT.(Card 26): If an error is encountered control is transferred to this card, and (in this case) the job is terminated.

12-2 Input Data (Record 2)

The input directives and data sets for the START UP processor are discussed in Section 8-2. The input is for a clay over shale column, see figure 12-1. The column is represented by 90 elements in the x direction, and by one element in the y and z directions. The elements are separated into two zones: a 20 element clay zone, and a 70 element shale zone. All dimensional entities for the sample problem are given in terms of inches, kips, and milliseconds.

12-2.3 UPDATE Directives for TPL1 (record 3)

This record contains the modification directives for UPDATE on control card 5. In this case, no modifications are made before file COMPILE is produced.

12-2.4 UPDATE Directives for TRNLDIR (Record 4)

The modification directives in this record are for UPDATE on control card 10. The \$COMPILE, STARTUP card causes all card images associated with the \$DECK, STARTUP card to be written on file DRCTVS.

12-3 TGS3D Execution

The execution deck for the TGS3D processor is given in Table 12-3. The driver program is identified by the \$DECK, TGS3D card in Table 12-10. It's corresponding update directives are identified by the \$IDENT, PROGRAM card of Table 12-3 (record 4).

12-3.1 Control Cards

Two control cards need some explanation in the TGS3D execution deck. Card 23 is a GETTAPE card. This card retrieves a file (TAPE50) off a magnetic tape whose identifying number is TAPE50. TAPE50 is a file saved by the STOTAPE of the STARTUP job.

Card 27 is a DISPOSE card. This card informs the system that file TAPE99 is a plot output file and should be diverted to a system Calcomp plotter.

Notice that library TL1B2 and FELDA is needed for a plot job. TL1B2 contains the three device dependent routines TGSINIT, TGSLINE, and TGSTERM discussed in Section 10-1.5. These are set up to call the Calcomp routines in the system library FELDA. They must be modified to reflect plotting on other devices. An OLDPL form of TL1B2 is TPL2 on the TRANAL library.

Also note that only TAPE50 and TAPE60 is used here because this is a zone plot job. TAPE50 is the labeled commons and TAPE60 is all the random files.

If region plotting is desired then TAPE11, TAPE12, TAPE13, and TAPE14 must be retrieved instead of TAPE60 as well as declared on the program card of the TGS3D processor. Of course STARTUP must have saved TAPE11, TAPE12, TAPE13, and TAPE14. Otherwise the same driver and control card setup can be used.

The TGS3D driver determines which type of plotting is desired from the input directives (record 2). However only one type of plotting can be done during a single run. In other words, zone and region plotting in the same job is not allowed at the present time.

12-3.2 Geometry Plot

The plot resulting from the input directives in Table 12-3 is shown in figure 12-2. The top 20 elements represent the subcycled clay zone. Only 11 of the 70 shale elements are shown in figure 12-2.

12-4 CLEAN UP Execution

The driver program and UPDATE directives for the CLEAN UP processor are identified by the \$DECK,CLEANUP card in Table 12-10. The execution deck is given in Table 12-4.

12-4.1 Control Cards

All of the control cards have been explained in Sections 12-2.1 and 12-3.1. The directory name paths and data set names on the GETTAPE and STOTAPE control cards (cards 21, 22, 27, 28, and 29) are self-explanatory. The CLEAN UP processor requires files TAPE50 and TAPE60 produced by the START UP processor.

12-4.2 Input Data

The directives and data sets for CLEAN UP are discussed in Section 8-6. Zone 1 (the clay elements) is subcycled with a ratio of 5. The air over-pressure shock is applied to surface 5 (clay top) of the column. It should be noted that subcycling of the sample problem is not really necessary. However, the element size in the clay was selected so that subcycling could be included in the sample problem.

12-4.3 UPDATE Directives

The UPDATE directive records (records 3 and 4) are self-explanatory, see Sections 12-2.3 and 12-2.4.

12-5 First PROSSOR Execution

The PROSSOR execution deck is given in Table 12-5, and the driver program and UPDATE directives are identified by the \$DECK,PROSSOR card in Table 12-10. The files TAPE3, TAPE50, and TAPE60 produced by CLEAN UP are required for the first execution of PROSSOR. The control cards are self-explanatory. Input for PROSSOR is discussed in Section 8-8: 60 major time increments are specified. The UPDATE directive is self-explanatory.

12-6 Second PROSSOR Execution

The execution deck is given in Table 12-6. Files TAPE50 and TAPE60 produced by the previous (first) execution of PROSSOR are required. Fourty major time increments are specified on the input record. All else follows from Section 12-5.

12-7 COMBINE Execution

The execution deck is given in Table 12-7, and the driver program and UPDATE directives are identified by the \$DECK,COMBINE card in Table 12-10. Since random files are not required by COMBINE, the binary object code library image of TPL1, i.e. TLIB1, is retrieved from the TRANAL library on the PSS (control card 4). TLIB1 is specified as a P file on the LINK card (control card 19). The remaining control cards are self-explanatory. Input is discussed in Section 8-10: all records from two results output files (TAPE3's) are to be combined.

12-8 TRNSPOS Execution

The execution deck is given in Table 12-8, and the driver program and UPDATE directives are identified by the \$DECK,TRNSPOS card in Table 12-10. Input is discussed in Section 8-12. A partial transpose is specified: only records 1, 4, 7, 10, and 13 are to be transposed.

12-9 TGSHIST Execution

The execution deck for the TGSHIST processor is given in Table 12-9. The driver program is identified by the \$DECK,TGSHIST card in Table 12-10.

12-9.1 Control Cards

All of the control card types have appeared previously and their functions should be evident.

12-9.2 Time History Plots

The plots generated by the input directives (record 2) of this job are shown in figures 12-3, 12-4, 12-5, and 12-6. The I,J,K output station numbers of the plotted histories appear just before the date. The output station 4, 1, 1 is 6 feet below the surface, and 23, 1, 1 is 32 feet down. The dashed curve in each figure represents results from the two zone subcycled problem configuration. To illustrate the effects of subcycling the sample problem was rerun in a one zone (no subcycling) configuration. These results are represented by solid curves in the figures. Figures 12-5 and 12-6 are integrated velocities, i.e. displacements.

TABLE 12-1
PROCESSOR DATA FILES

Tape Number	File Type	Contents	PROCESSORS							
			S T A R T U P	A L T E R F L	C L E A N U P	P R O S S O R	C O M B I N E	T R N S P O S	T G S 3 D	T G H I S T
1	S	Input	X	X	X	X	X	X	X	X
2	S	System Output	X	X	X	X	X	X	X	X
3	S	Results Output			X	X				
5	S	Applied Boundary Velocities			X	X				
6	R	Applied Boundary Velocities			X	X				
11	S	Region Corner Coordinates	X						X	
12	S	Region X-Edge Coord. Directives	X						X	
13	S	Region Y-Edge Coord. Directives	X						X	
14	S	Region Z-Edge Coord. Directives	X						X	
15	R	Node Coordinates	X	X	X	X			X	
20	R	Node Forces			X	X				
30	R	Element Stresses			X	X				
32	R	Element Information	X	X	X	X			X	
40	R	Node Velocities			X	X				
50	S	SCM Common	X	X	X	X			X	
60	S	LCM Image	X	X	X	X				

S = Sequential, R = Random

TABLE 12-1 (Cont'd)

Tape Number	File Type	Contents	PROCESSORS							
			S T A R T U P	A L T E R F L	C L E A N U P	P R O S S O R	C O M B I N E	T R A N S P O S	T C S 3 D	T G H I S T
70	S	Combined Results Output					X	X		
71(3)	S	Primary Results Output					X			
7N(3) N=2,3,...	S	Secondary Results Output					X			
8	S	Transposed Results Output						X		
91	S	Tape 8 type file used as TGHIST input								X
92	S	Same as above								X
93	S	Same as above								X
94	S	Same as above								X
97	S	Temporary work file								X
98	S	Region data file							X	
99	S	Plot file							X	X

S = Sequential

TABLE 12-2

START UP EXECUTION DECK

JOB CARD	1
*7,PSS	2
COPY(INPUT,1R,TAPE1/RR)	3
FETCHPS(TRANAL,OLDPL,TPL1)	4
UPDATE(F,L=1)	5
RETURN(OLDPL)	6
FTN4(I,B=TBL1,L=0)	7
RETURN(COMPIL)	8
FETCHPS(TRANAL,OLDPL,TRNLDIR)	9
UPDATE(C=DRCTVS,L=1,MC=[S])	10
RETURN(OLDPL)	11
FETCHPS(TRANAL,OLDPL,TRNLPL)	12
UPDATE(I=DRCTVS,L=1)	13
RETURN(OLDPL)	14
RETURN(DRCTVS)	15
FTN4(I,B=PRGMBL,L=0)	16
RETURN(COMPIL)	17
FETCHPS(TRANAL,TRNLLIB,TRNLLIB)	18
SCATTER(TRNLLIB)	19
RETURN(TRNLLIB)	20
LINK(X,F=PRGMBL,F=TBL1,P=LIBRARY,CS=ID,RF)	21
REWIND(TAPE60)	22
SAVELCM(TAPE60)	23
STOTAPE(TAPE50=EXAMPLE/STARTUP/TAPE50,TAPENO)	24
STOTAPE(TAPE60=EXAMPLE/STARTUP/TAPE60,TAPENO)	25
EXIT.	26
789 (IN COLUMN ONE) ** END OF RECORD CARD **	
\$ TITLE 'SUBCYCLE Y - CLAY OVER SHALE ILLUSTRATIVE SAMPLE PROBLEM'	
\$ IDENTIFIER 'EXAMPLE'	
\$ PRINT OPTIONS ALL	
\$ ZONE DEFINITION	
X ZONES 2 REGIONS 1 1 ELEMENTS 20 70	
Y ZONES 1 REGIONS 1 ELEMENTS 1	
Z ZONES 1 REGIONS 1 ELEMENTS 1	
\$ ZONE TYPE 1 TO 2 HEX8	
\$ MATERIALS	
NAME = 'CLAY' TYPE = SOIL	
RHO = 0.17972, K = 340.0, G = 85.0, A = 0.1, B = 1.5, C = 0.07,	
D = 2.0, R = 3.0, W = 0.04, TCUT = 0.03, GEOP = 0.0	
NAME = 'SHALE' TYPE = SOIL	
RHO = 0.20518, K = 540.0, G = 115.0, A = 0.866, B = 2.5, C = 0.361,	
D = 0.5, R = 3.0, W = 0.0005, TCUT = 4.5, GEOP = 0.0	
\$ STABILITY FACTOR = 0.95 VOLUME = 5 ANGLE = 5	
\$ DAMPING VELOCITY = 0.40	
NAME = 'CLAY' CLINEAR = 0.10 CQUADRATIC = 4.00	
\$ REGION MATERIALS 1 'CLAY' 2 'SHALE'	

TABLE 12-2 (CONT'D)

\$	REGION COORDS	CORNERS		
	0.00	2.20440+04	-36.00	
	2.40000+02	2.20440+04	-36.00	
	5.28000+03	2.20440+04	-36.00	
	0.00	2.21160+04	-36.00	
	2.40000+02	2.21160+04	-36.00	
	5.28000+03	2.21160+04	-36.00	
	0.00	2.20440+04	36.00	
	2.40000+02	2.20440+04	36.00	
	5.28000+03	2.20440+04	36.00	
	0.00	2.21160+04	36.00	
	2.40000+02	2.21160+04	36.00	
	5.28000+03	2.21160+04	36.00	
789	(IN COLUMN ONE)	**	END OF RECORD CARD	**
*IDENT,MS-CORE				
789	(IN COLUMN ONE)	**	END OF RECORD CARD	**
\$IDENT,PROGRAM				
\$COMPILE,STARTUP				
789	(IN COLUMN ONE)	**	END OF RECORD CARD	**
6789	(IN COLUMN ONE)	**	END OF FILE CARD	**

TABLE 12-3
TGS3D EXECUTION DECK

JOB CARD	1
*7,PSS	2
COPY(INPUT,1R,TAPE1/BR)	3
FETCHPS(TRANAL,OLDPL,TPL1)	4
UPDATE(F,L=1)	5
RETURN(OLDPL)	6
FTN4(I,B=TBL1,L=0)	7
RETURN(COMPIL)	8
FETCHPS(TRANAL,OLDPL,TRNLDIR)	9
UPDATE(C=DRCTVS,L=1,MC=[S])	10
RETURN(OLDPL)	11
FETCHPS(TRANAL,OLDPL,TRNLPL)	12
UPDATE(I=DRCTVS,L=1)	13
RETURN(OLDPL)	14
RETURN(DRCTVS)	15
FTN4(I,B=ZNPLT,L=0)	16
RETURN(COMPIL)	17
FETCHPS(CALDRA,FELDA,FELDA)	18
FETCHPS(TRANAL,TLIB2,TLIB2)	19
FETCHPS(TRANAL,TRNLLIB,TRNLLIB)	20
SCATTER(TRNLLIB)	21
RETURN(TRNLLIB)	22
GETTAPE(TAPE50=EXAMPLE/PROSORA/TAPE50,TAPENO)	23
GETTAPE(TAPE60=EXAMPLE/PROSORA/TAPE60,TAPENO)	24
LOADLCM(TAPE60)	25
LINK(X,F=ZNPLT,F=TBL1,P=TLIB2,P=LIBRARY,P=FELDA,CS=ID,RF)	26
DISPOSE(TAPE99=PL,SC=BKY)	27
EXIT.	28
789 (IN COLUMN ONE) ** END OF RECORD CARD **	
\$TITLE 'ZONE PLOT'	
\$VIEW OPTION PERSPECTIV HIDE	
\$OBJECT ZONE I=0 J=1 K=1	
ELEMENT I=0 J=1 K=1	
FACE 1 2 3	
\$TRANSFORM	
TRANSLATE 0.,-22044.,0.	
ROTATE Z 90. Y -45. Z -30.	
TRANSLATE 0., 0., 300.	
\$DISPLAY	
789 (IN COLUMN ONE) ** END OF RECORD CARD **	
*IDENT,MS-CORE	
789 (IN COLUMN ONE) ** END OF RECORD CARD **	
\$IDENT,PROGRAM	
\$COMPILE,TGS3D	
789 (IN COLUMN ONE) ** END OF RECORD CARD **	
6789 (IN COLUMN ONE) ** END OF FILE CARD **	

TABLE 12-4
CLEAN UP EXECUTION DECK

JOB CARD	1
*7,PSS	2
COPY(INPUT,1R,TAPE1/BR)	3
FETCHPS(TRANAL,OLDPL,TPL1)	4
UPDATE(F,L=1)	5
RETURN(OLDPL)	6
FTN4(I,B=TBL1,L=0)	7
RETURN(COMPILE)	8
FETCHPS(TRANAL,OLDPL,TRNLDIR)	9
UPDATE(C=DRCTVS,L=1,MC= [S])	10
RETURN(OLDPL)	11
FETCHPS(TRANAL,OLDPL,TRNLPL)	12
UPDATE(I=DRCTVS,L=1)	13
RETURN(OLDPL)	14
RETURN(DRCTVS)	15
FTN4(I,B=PRGMBL,L=0)	16
RETURN(COMPILE)	17
FETCHPS(TRANAL,TRNLLIB,TRNLLIB)	18
SCATTER(TRNLLIB)	19
RETURN(TRNLLIB)	20
GETTAPE(TAPE50=EXAMPLE/STARTUP/TAPE50,TAPENO)	21
GETTAPE(TAPE60=EXAMPLE/STARTUP/TAPE60,TAPENO)	22
LOADLCM(TAPE60)	23
LINK(X,F=PRGMBL,F=TBL1,P=LIBRARY,CS=ID,RF)	24
REWIND(TAPE60)	25
SAVELCM(TAPE60)	26
STOTAPE(TAPE3=EXAMPLE/CLEANUP/TAPE3,TAPENO)	27
STOTAPE(TAPE50=EXAMPLE/CLEANUP/TAPE50,TAPENO)	28
STOTAPE(TAPE60=EXAMPLE/CLEANUP/TAPE60,TAPENO)	29
EXIT.	30
789 (IN COLUMN ONE) ** END OF RECORD CARD **	
\$ PRINT OPTIONS ALL	
\$ TIME DATA START = 116.0 DELTAT = 1.00	
\$ SUBCYCLE RATIOS 5	
\$ ZONE SUBCYCLES 1 RATIO = 5 2 RATIO = 1	
\$ BOUNDARY CONDITIONS SYMMETRY PLANE, SYMMETRY PLANE,	
SYMMETRY PLANE, SYMMETRY PLANE,	
OVER PRESSURE, FREE BOUNDARY	
\$ OUTPUT VELOCITY 1 1 1, 4 1 1, 21 1 1, 23 1 1 STRESS 1 1 1	
\$ BRODE LOAD YIELDKT = 1000.0 HOBFEET = 0.0	
789 (IN COLUMN ONE) ** END OF RECORD CARD **	
*IDENT,MS-CORE	
789 (IN COLUMN ONE) ** END OF RECORD CARD **	
\$IDENT,PROGRAM	
\$COMPILE,CLEANUP	
789 (IN COLUMN ONE) ** END OF RECORD CARD **	
6789 (IN COLUMN ONE) ** END OF FILE CARD **	

TABLE 12-5
FIRST PROSSOR EXECUTION DECK

JOB CARD	1
*7,PSS	2
COPY(INPUT,1R,TAPE1/BR)	3
FETCHPS(TRANAL,OLDPL,TPL1)	4
UPDATE(F,L=1)	5
RETURN(OLDPL)	6
FTN4(I,B=TBL1,L=0)	7
RETURN(COMPIL)	8
FETCHPS(TRANAL,OLDPL,TRNLDIR)	9
UPDATE(C=DRCTVS,L=1,MC=[S])	10
RETURN(OLDPL)	11
FETCHPS(TRANAL,OLDPL,TRNLPL)	12
UPDATE(I=DRCTVS,L=1)	13
RETURN(OLDPL)	14
RETURN(DRCTVS)	15
FTN4(I,B=PRGMBL,L=0)	16
RETURN(COMPIL)	17
FETCHPS(TRANAL,TRNLLIB,TRNLLIB)	18
SCATTER(TRNLLIB)	19
RETURN(TRNLLIB)	20
GETTAPE(TAPE3=EXAMPLE/CLEANUP/TAPE3,TAPENO)	21
GETTAPE(TAPE50=EXAMPLE/CLEANUP/TAPE50,TAPENO)	22
GETTAPE(TAPE60=EXAMPLE/CLEANUP/TAPE60,TAPENO)	23
LOADLCM(TAPE60)	24
LINK(X,F=PRGMBL,F=TBL1,P=LIBRARY,CS=ID,RF)	25
REWIND(TAPE60)	26
SAVELCM(TAPE60)	27
STOTAPE(TAPE3=EXAMPLE/PROSORA/TAPE3,TAPENO)	28
STOTAPE(TAPE50=EXAMPLE/PROSORA/TAPE50,TAPENO)	29
STOTAPE(TAPE60=EXAMPLE/PROSORA/TAPE60,TAPENO)	30
EXIT.	31
789 (IN COLUMN ONE) ** END OF RECORD CARD **	
\$ TIME INCREMENTS 60	
789 (IN COLUMN ONE) ** END OF RECORD CARD **	
*IDENT,MS-CORE	
789 (IN COLUMN ONE) ** END OF RECORD CARD **	
\$IDENT,PROGRAM	
\$COMPILE,PROSSOR	
789 (IN COLUMN ONE) ** END OF RECORD CARD **	
6789 (IN COLUMN ONE) ** END OF FILE CARD **	

TABLE 12-6
SECOND PROSSOR EXECUTION DECK

JORCARD	1
*7,PSS	2
COPY(INPUT,1R,TAPE1/BR)	3
FETCHPS(TRANAL,OLDPL,TPL1)	4
UPDATE(F,L=1)	5
RETURN(OLDPL)	6
FTN4(I,R=TBL1,L=0)	7
RETURN(COMPILE)	8
FETCHPS(TRANAL,OLDPL,TRNLDIR)	9
UPDATE(C=DRCTVS,L=1,MC=[S])	10
RETURN(OLDPL)	11
FETCHPS(TRANAL,OLDPL,TRNLPL)	12
UPDATE(I=DRCTVS,L=1)	13
RETURN(OLDPL)	14
RETURN(DRCTVS)	15
FTN4(I,R=PRGMBL,L=0)	16
RETURN(COMPILE)	17
FETCHPS(TRANAL,TRNLLIB,TRNLLIB)	18
SCATTER(TRNLLIB)	19
RETURN(TRNLLIB)	20
GETTAPE(TAPE50=EXAMPLE/PROSORA/TAPE50,TAPENO)	21
GETTAPE(TAPE60=EXAMPLE/PROSORA/TAPE60,TAPENO)	22
LOADLCM(TAPE60)	23
LINK(X,F=PRGMBL,F=TBL1,P=LIBRARY,CS=ID,RF)	24
REWIND(TAPE60)	25
SAVELCM(TAPE60)	26
STOTAPE(TAPE3=EXAMPLE/PROSORB/TAPE3,TAPENO)	27
STOTAPE(TAPE50=EXAMPLE/PROSORB/TAPE50,TAPENO)	28
STOTAPE(TAPE60=EXAMPLE/PROSORB/TAPE60,TAPENO)	29
EXIT.	30
789 (IN COLUMN ONE) ** END OF RECORD CARD **	
\$ TIME INCREMENTS 40	
789 (IN COLUMN ONE) ** END OF RECORD CARD **	
*IDENT,MS-CORE	
789 (IN COLUMN ONE) ** END OF RECORD CARD **	
\$IDENT,PROGRAM	
\$COMPILE,PROSSOR	
789 (IN COLUMN ONE) ** END OF RECORD CARD **	
6789 (IN COLUMN ONE) ** END OF FILE CARD **	

TABLE 12-7
COMBINE EXECUTION DECK

JOB CARD	1
*7,PSS	2
COPY(INPUT,1R,TAPE1/BR)	3
FETCHPS(TRANAL,TLIR1,TLIB1)	4
FETCHPS(TRANAL,OLDPL,TRNLDIR)	5
UPDATE(C=DRCTVS,L=1,MC=[S])	6
RETURN(OLDPL)	7
FETCHPS(TRANAL,OLDPL,TRNLPL)	8
UPDATE(I=DRCTVS,L=1)	9
RETURN(OLDPL)	10
RETURN(DRCTVS)	11
FTN4(I,B=PRGMBL,L=0)	12
RETURN(COMPIL)	13
FETCHPS(TRANAL,TRNLLIB,TRNLLIB)	14
SCATTER(TRNLLIB)	15
RETURN(TRNLLIB)	16
GETTAPE(TAPE71=EXAMPLE/PROSORA/TAPE3,TAPENO)	17
GETTAPE(TAPE72=EXAMPLE/PROSORB/TAPE3,TAPENO)	18
LINK(X,F=PRGMBL,P=LIBRARY,P=TLIB1,CS=ID,RF)	19
STOTAPE(TAPE70=EXAMPLE/COMBINE/TAPE70,TAPENO)	20
EXIT.	21
789 (IN COLUMN ONE) ** END OF RECORD CARD **	
\$ PRINT OPTIONS ALL	
\$ COPY RECORDS 0 0	
789 (IN COLUMN ONE) ** END OF RECORD CARD **	
\$ IDENT,PROGRAM	
\$ COMPIL,COMBINE	
789 (IN COLUMN ONE) ** END OF RECORD CARD **	
6789 (IN COLUMN ONE) ** END OF FILE CARD **	

TABLE 12-8
TRNSPOS EXECUTION DECK

JOB CARD	1
*7,PSS	2
COPY(INPUT,1R,TAPE1/BR)	3
FETCHPS(TRANAL,TLIR1,TLIB1)	4
FETCHPS(TRANAL,OLDPL,TRNLDIR)	5
UPDATE(C=DRCTVS,L=1,MC=[5])	6
RETURN(OLDPL)	7
FETCHPS(TRANAL,OLDPL,TRNLPL)	8
UPDATE(I=DRCTVS,L=1)	9
RETURN(OLDPL)	10
RETURN(DRCTVS)	11
FTN4(I,B=PRGMBL,L=0)	12
RETURN(COMPIL)	13
FETCHPS(TRANAL,TRNLLIB,TRNLLIB)	14
SCATTER(TRNLLIB)	15
RETURN(TRNLLIB)	16
GETTAPE(TAPE70=EXAMPLE/COMBINE/TAPE70,TAPENO)	17
LINK(X,F=PRGMBL,P=LIBRARY,P=TLIB1,CS=ID,RF)	18
STOTAPE(TAPE8=EXAMPLE/TRNSPOS/TAPE8,TAPENO)	19
EXIT.	20
789 (IN COLUMN ONE) ** END OF RECORD CARD **	
\$ PRINT OPTIONS ALL	
\$ TRANSPOSE TYPE = PARTIAL RECORDS 1 4 7 10 13	
789 (IN COLUMN ONE) ** END OF RECORD CARD **	
\$ IDENT,PROGRAM	
\$ COMPILE,TRNSPOS	
789 (IN COLUMN ONE) ** END OF RECORD CARD **	
6789 (IN COLUMN ONE) ** END OF FILE CARD **	

TABLE 12-9
TGSHIST EXECUTION DECK

JOB CARD	1
*7,PSS	2
COPY(INPUT,1R,TAPE1/BR)	3
FETCHPS(TRANAL,TLIB1,TLIB1)	4
FETCHPS(TRANAL,OLDPL,TRNLDIR)	5
UPDATE(C=DRCTVS,L=1,MC=[S])	6
RETURN(OLDPL)	7
FETCHPS(TRANAL,OLDPL,TRNLPL)	8
UPDATE(I=DRCTVS,L=1)	9
RETURN(OLDPL)	10
RETURN(DRCTVS)	11
FTN4(I,B=HSPLT,L=0)	12
RETURN(COMPIL)	13
FETCHPS(TRANAL,TLIB2,TLIB2)	14
FETCHPS(CALDRA,FELDA,FELDA)	15
FETCHPS(TRANAL,TRNLLIB,TRNLLIB)	16
SCATTER(TRNLLIB)	17
RETURN(TRNLLIB)	18
GETTAPE(TAPE7=EXAMPLE/TRNSPOS/TAPE8,TAPENO)	19
GETTAPE(TAPE6=ONEZONE/TRNSPOS/TAPE8,TAPENO)	20
LINK(X,F=HSPLT,P=LIBRARY,P=TLIB1,P=TLIB2,P=FELDA,CS=ID,RF)	21
DISPOSE(TAPE99=PL,SC=BKY)	22
EXIT.	23
789 (IN COLUMN ONE) ** END OF RECORD CARD **	
\$TITLE 'TIME HISTORY PLOT'	
\$TIME INTERVAL 110. 190.	
\$PLOT INFO NOPLOT=2 NOTAPE=2 INTEGRATE=0	
CURVES 4 4 TAPES 1 2 MINMAX 0. .45	
CURVES 10 10 TAPES 1 2 MINMAX 0. .15	
\$PLOT INFO NOPLOT=2 NOTAPE=2 INTEGRATE=1	
CURVES 4 4 TAPES 1 2 MINMAX 0. 12.	
CURVES 10 10 TAPES 1 2 MINMAX 0. 3.	
789 (IN COLUMN ONE) ** END OF RECORD CARD **	
\$IDENT,PROGRAM	
\$COMPILE,TGSHIST	
789 (IN COLUMN ONE) ** END OF RECORD CARD **	
6789 (IN COLUMN ONE) ** END OF FILE CARD **	

TABLE 12-10
TRNLDIR SOURCE DECK

```

$DECK,STARTUP
*IDENT,PROGRAM
*INSERT,AAAAAA.1
    PROGRAM START UP ( TAPE1, OUTPUT, TAPE2 = OUTPUT, TAPE11, TAPE12,
    $                   TAPE13, TAPE14, TAPE50 )
C
    CALL INTLIZ
    CALL RSRV MS
    CALL READ SU
    CALL ELM TBL
    CALL CAT CRD
    CALL CORD PP
    CALL SUBCYL
    CALL END CMN
C
    END
$DECK,ALTERFL
*IDENT,PROGRAM
*INSERT,AAAAAA.1
    PROGRAM ALTER FL ( TAPE1, OUTPUT, TAPE2 = OUTPUT, TAPE50 )
C
    CALL BEG CMN
    CALL BEG FIL
    CALL READ AF
    CALL MOD ELM
    CALL MOD CRD
    CALL SUBCYL
    CALL END CMN
C
    END
$DECK,CLEANUP
*IDENT,PROGRAM
*INSERT,AAAAAA.1
    PROGRAM CLEAN UP ( TAPE1, OUTPUT, TAPE2 = OUTPUT, TAPE3, TAPE50 )
C
    CALL BEG CMN
    CALL BEG FIL
    CALL READ CU
    CALL CYL TBL
    CALL ITF TBL
    CALL MASS PP
    CALL OUT PRP
    CALL CAT VEL
    CALL CAT FRC
    CALL CAT SIG
    CALL ILZ SIG
    CALL ILZ LYR
    CALL JANITOR
    CALL END CMN
C
    END

```

TABLE 12-10 (CONT'D)

```

$DECK,PROSSOR
*IDENT,PROGRAM
*INSERT,AAAAAA.1
    PROGRAM PROSSOR ( TAPE1, OUTPUT, TAPE2 = OUTPUT, TAPE3, TAPE50 )
C
    CALL BEG CMN
    CALL BEG FIL
    CALL READ PR
    CALL TM PRSR
    CALL END CMN
C
    END
$DECK,COMBINE
*IDENT,PROGRAM
*INSERT,AAAAAA.1
    PROGRAM COMBINE ( TAPE1, OUTPUT, TAPE2 = OUTPUT, TAPE70,
    $                  TAPE71, TAPE72 )
C
    CALL INTLIZ
    CALL BEG CMB
    CALL READ CR
    CALL TAP CMB
C
    END
$DECK,TRNSPOS
*IDENT,PROGRAM
*INSERT,AAAAAA.1
    PROGRAM TRNSPOS ( TAPE1, OUTPUT, TAPE2 = OUTPUT, TAPE8, TAPE70 )
C
    CALL INTLIZ
    CALL BEG TPS
    CALL READ TP
    CALL FIL TPS
C
    END
$DECK,TGS3D
*IDENT,PROGRAM
*INSERT,AAAAAA.1
    PROGRAM TGS 3D ( OUTPUT, TAPE1, TAPE2=OUTPUT, TAPE99, TAPE98 )
C
C    TRANAL INITIALIZATION
C
    CALL INTLIZ
    CALL REG P3D
C    REGION PLOT
    CALL RG PLOT
C
    CALL EXIT
    END

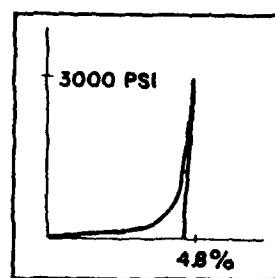
```

TABLE 12-10 (CONT'D)

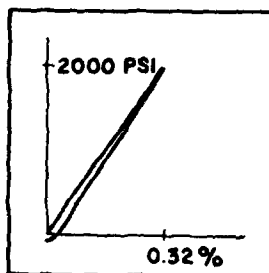
```

$DECK,TGS3D
*IDENT,PROGRAM
*INSERT,AAAAAA.1
    PROGRAM TGS 3D ( OUTPUT, TAPE1, TAPE2=OUTPUT, TAPE99, TAPE50 )
C
C    TRANAL INITIALIZATION
C
    CALL BEG CMN
    CALL BEG FIL
    CALL BEG P3D
C    ZONE PLOT
    CALL ZN PLOT
    CALL EXIT
C
    END
$DECK,TGSHIST
*IDENT,PROGRAM
*INSERT,AAAAAA.1
    PROGRAM TGS HIST( OUTPUT, TAPE1, TAPE2=OUTPUT, TAPE98, TAPE99,
      2          TAPE6, TAPE7 )
C
C    TRANAL INITIALIZATION
C
    CALL INTLIZ
    CALL BEG HIS P
C    TIME HISTORY PLOT
    CALL HIST PLT
    CALL EXIT
C
    END

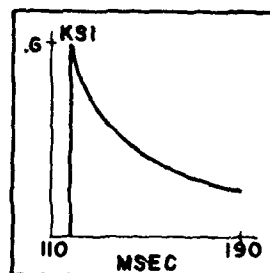
```



CLAY



SHALE



AIR OVERPRESSURE

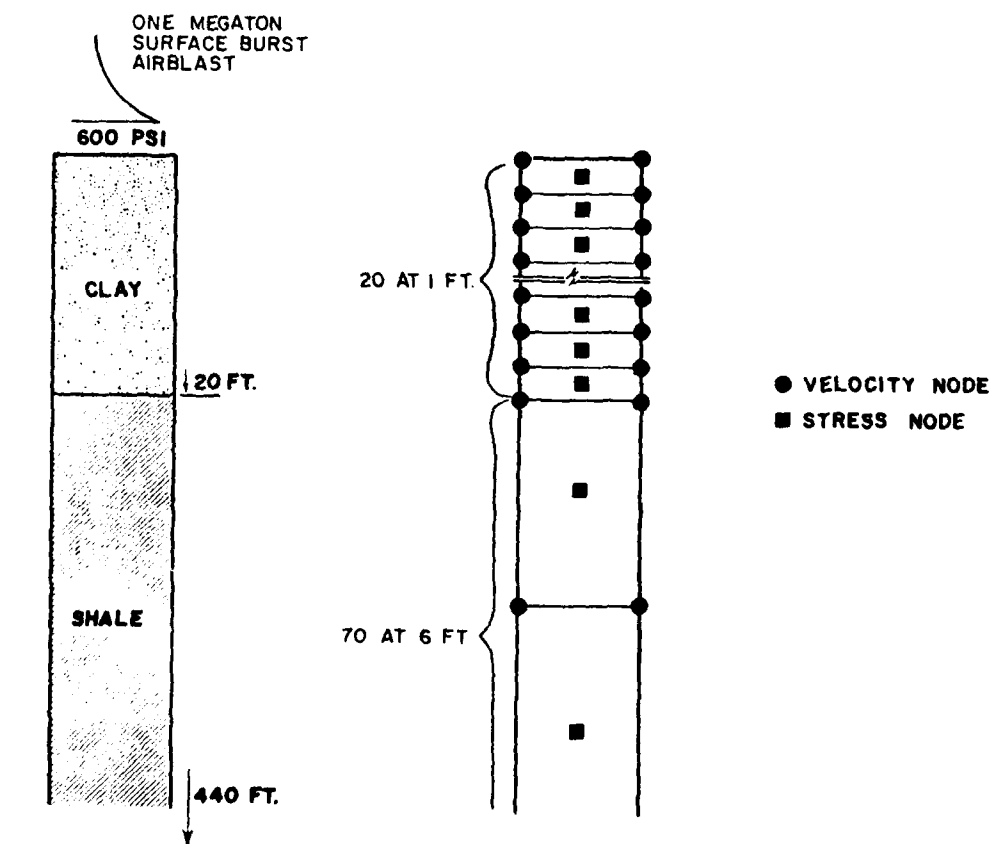


FIG. 12-1 ILLUSTRATIVE SAMPLE PROBLEM

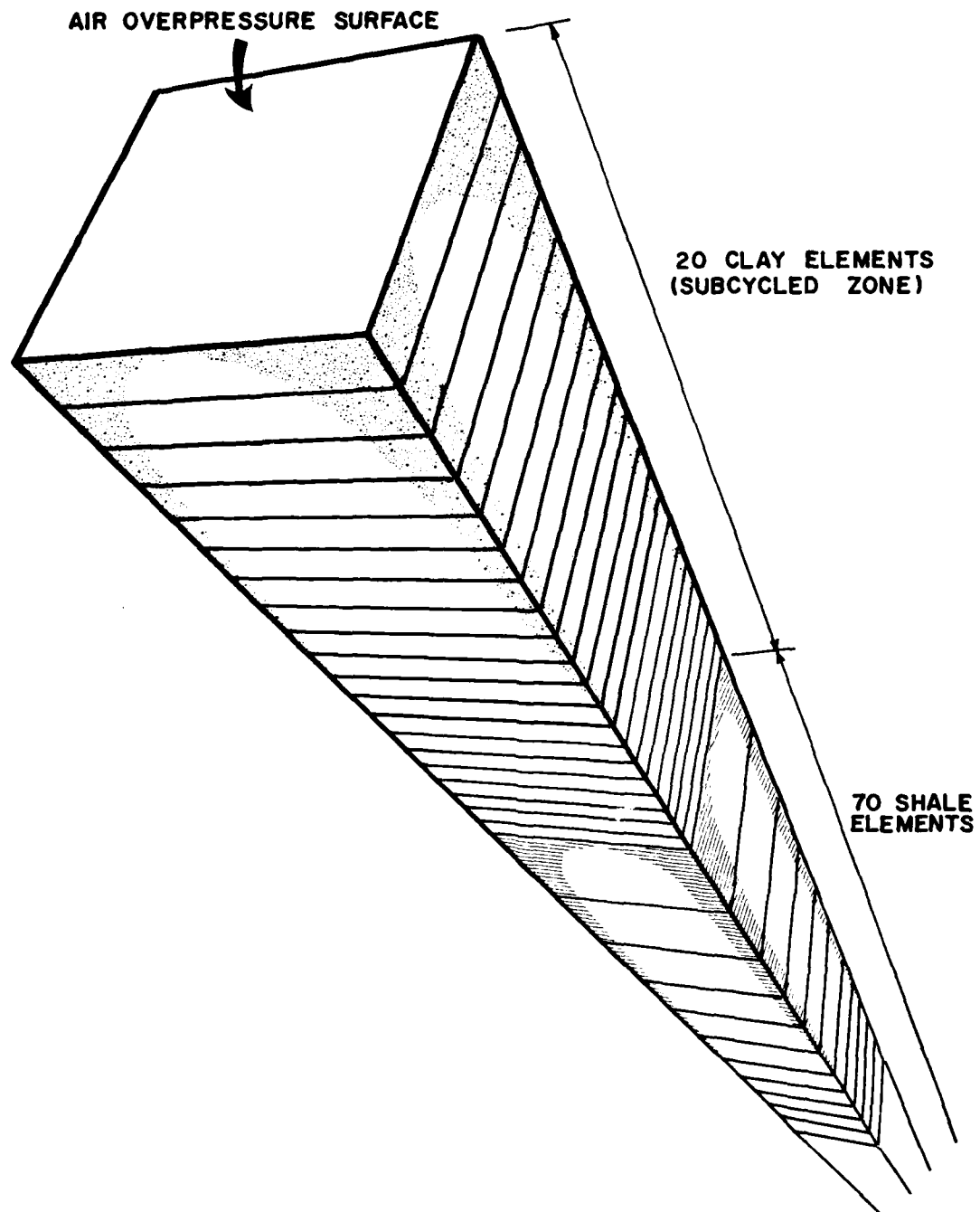


FIG. 12-2 SAMPLE PROBLEM GEOMETRY PLOT

EXAMPLE 4 1 1 79/01/26.

— SUBCYCLE N
 ---- SUBCYCLE Y

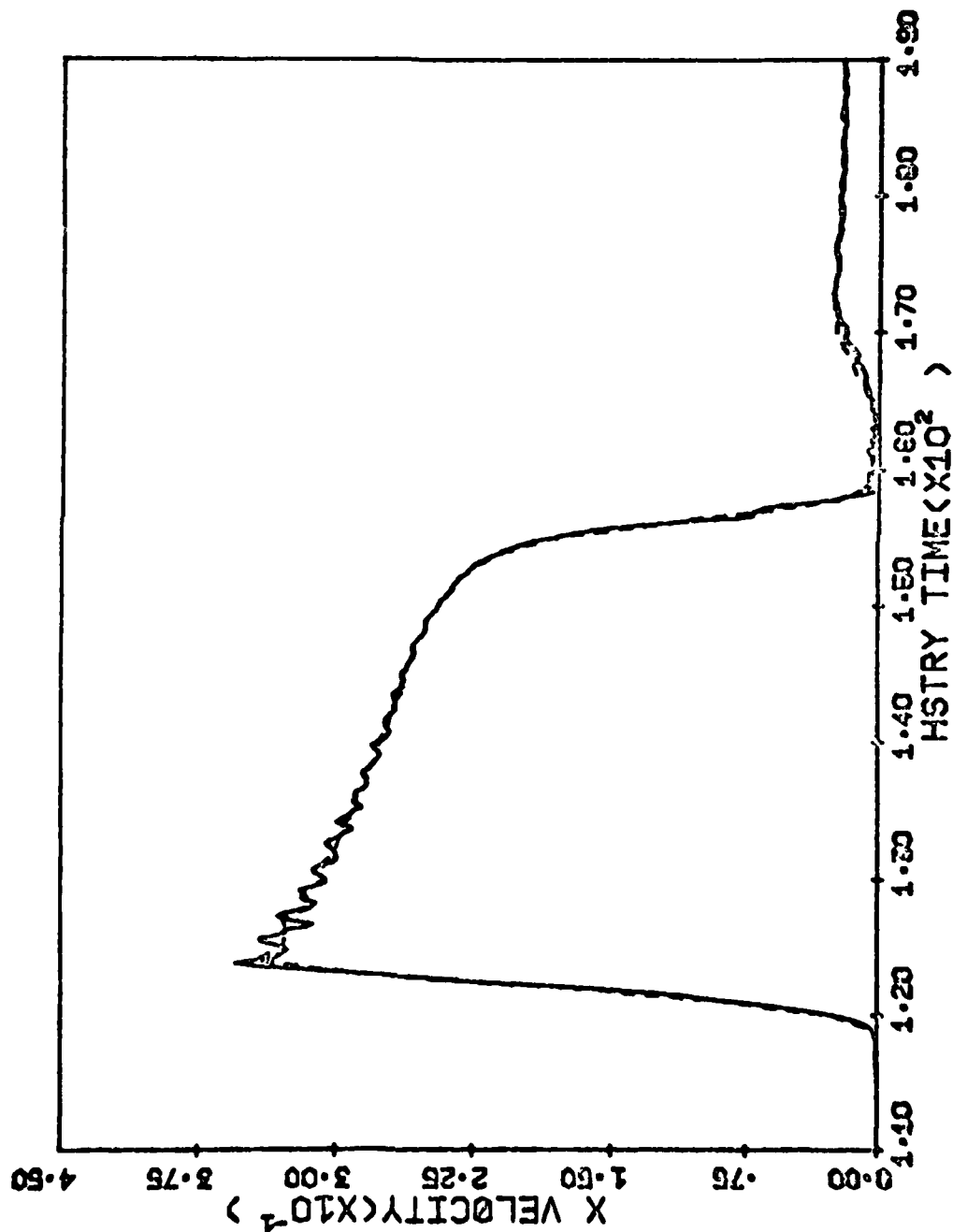


FIG. 12-3

EXAMPLE 23 1 1 79/01/28.

— SUBCYCLE N
 - - - - SUBCYCLE Y

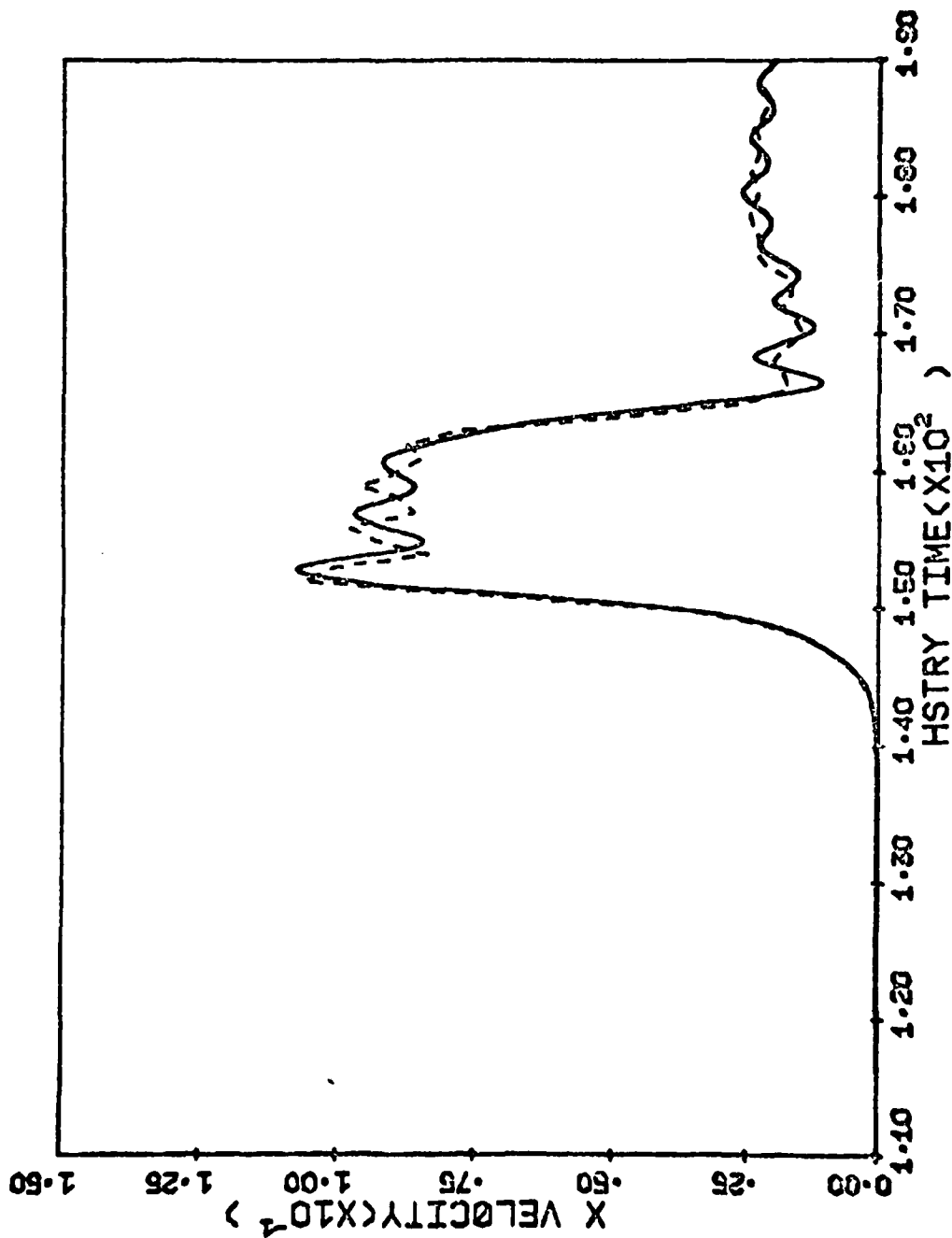


FIG. 12-4

EXAMPLE 4 1 1 79/01/26.

— SUBCYCLE N
 ---- SUBCYCLE Y

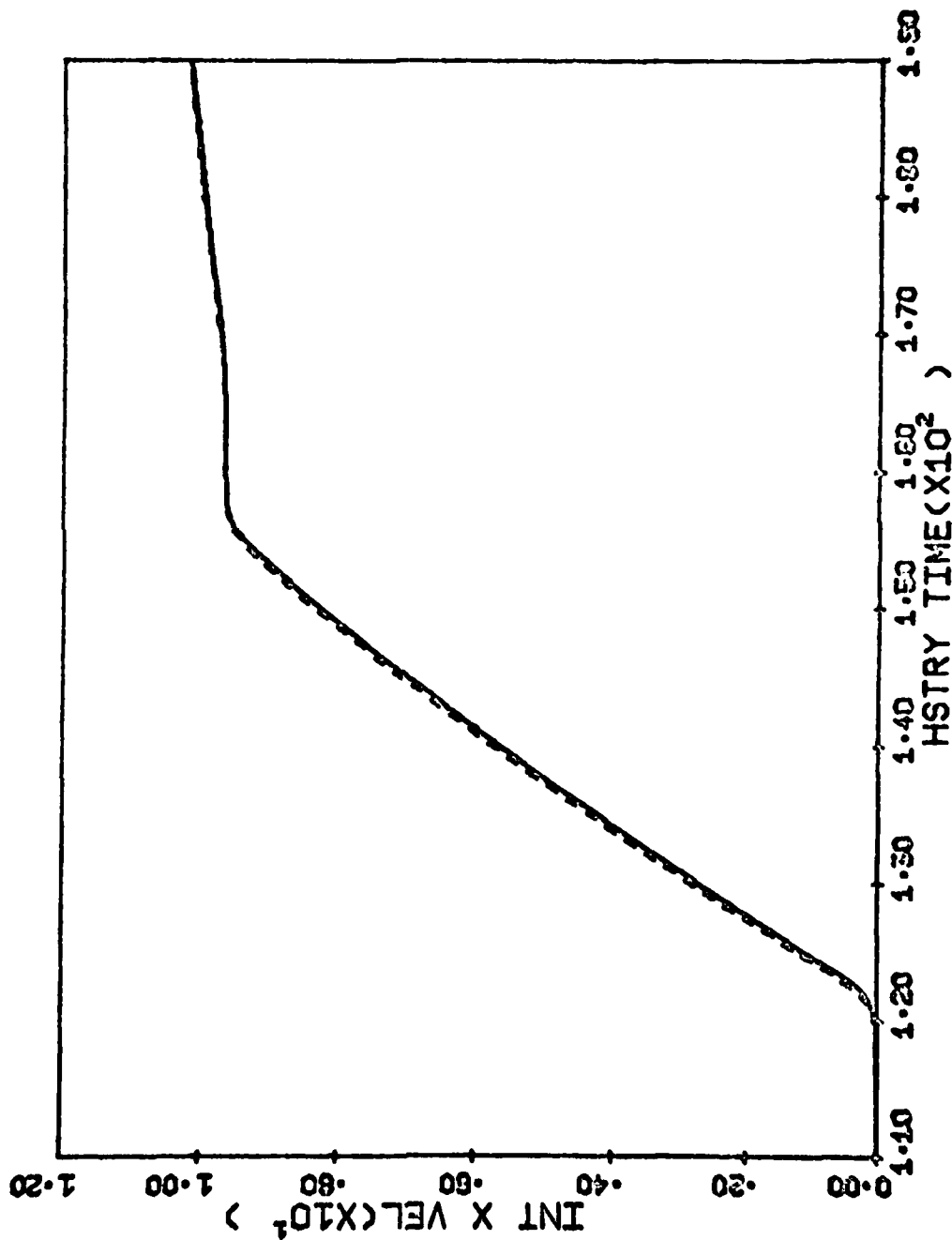


FIG. 12-5

EXAMPLE 23 1 1 79/01/26.

— SUECYCLE N
- - - SUECYCLE Y

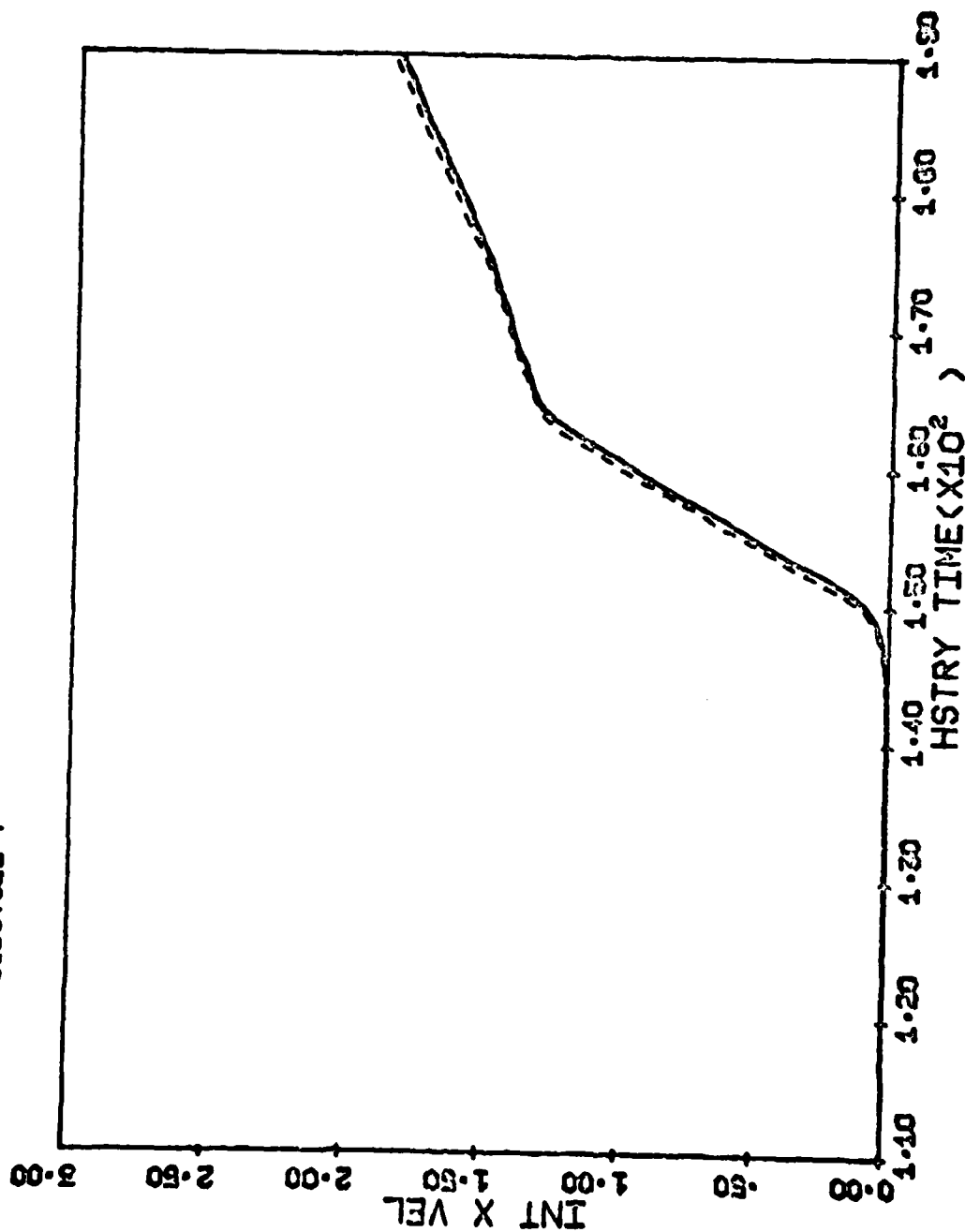


FIG. 12-6

APPENDIX A

NOTATIONS

The following is a list of notations that occur in more than one section of the report. All other notations are defined in the text as they are needed.

UC	Un-mapped soil island/structure configuration.
MC	Mapped soil island/structure configuration.
x,y,z	Global cartesian coordinates.
I,J,K	Element (node) numbers in x,y,z directions, respectfully.
ξ,η,ζ	Dimensionless local coordinates.
DA	Dynamic array: blank common.
DML	Data management library.
MDT	Major time increment.
MT	Major time: time at the end of a MDT.
SCM	Small core memory.
LCM	Large core memory.

APPENDIX B

REFERENCES

1. Methods of Computational Physics, edited by B. Adler et al, Vol. 3, Academic Press, New York, New York 1964.
2. R.D. Richtmyer, and K.W. Morton, "Difference Methods for Initial Value Problems", Interscience Publishers, New York, 1967.
3. J. Baylor, M.P. Bieniek and J.P. Wright, "TRANAL: A 3-D Finite Element Code for Transient Nonlinear Analysis", DNA 3501F, Weidlinger Associates, New York, New York, 1974.
4. J.P. Wright and J.L. Baylor, "Calculation of Three Test Problems by the 3-D TRANAL Code", DNA 3998F, Weidlinger Associates, New York, New York, 1976.
5. J.L. Baylor and J.P. Wright, "Three Dimensional Transient Nonlinear Analysis", in Finite Element Analysis of Transient Nonlinear Structural Behavior, A.S.M.E., AMD-Vol. 14, Winter Annual Meeting, Houston, Texas, pp. 179-191, December 1975.
6. F.L. DiMaggio and I.S. Sandler, "Material Model for Granular Soils", Journal of Engineering Mechanics Division, A.S.C.E., pp. 935-950, June 1971.
7. I.S. Sandler and D. Rubin, "A Modular Subroutine for the Cap Model", DNA 3875F, Weidlinger Associates, New York, New York, 1976.
8. J.P. Wright, "Mixed Time Integration Schemes", N.A.S.A. Symposium on Future Trends in Computerized Structural Analysis and Synthesis, Washington, D.C., October 1978; Also published in Journal of Computers and Structures, Vol. 10, pp. 235-238, 1979.
9. J.P. Wright, "Explicit Integration, Rayleigh's Principle and Numerical Stability", submitted for publication in the International Journal for Numerical Methods in Engineering, 1979.
10. S.H. Pang and J. Isenberg, "Structure-Medium Interaction and Structural Response Calculations for the Essex Phase 3 Event", Report No. WES-TR-N-78-3, Weidlinger Associates, Menlo Park, California, May 1978.
11. F.S. Wong and J. Isenberg, "Dynamic Structural Analysis of MAP Shelters", Report No. DNA4631Z, Weidlinger Associates, Menlo Park, California, June 1978.
12. H.S. Levine, J. Isenberg and J.P. Wright, "Three-Dimensional Soil-Island Calculation of the HARDPAN I-3 Embedded Structure Subjected to a Simulated Nuclear Explosion", SAMSO Technical Report, Weidlinger Associates, (Unpublished).

13. J. Isenberg, D.K. Vaughan and I. Sandler, "Nonlinear Soil-Structure Interaction", EPRI Report No. NP-945, Weidlinger Associates, Menlo Park, California, December 1978.
14. R. Smilowitz and J.P. Wright, "TRANAL Calculations of MIGHTY EPIC Test Structures C-Y-12, C-Y-13 and C-Y-14", forthcoming DNA report, Weidlinger Associates, New York (Unpublished).
15. J.P. Wright, "A Portable FORTRAN Data Manager", in Data Handling Techniques in Civil Engineering, A.S.C.E. National Structural Engineering Convention, New Orleans, Louisiana, April 1975.
16. I. Nelson, "Analysis of Soil-Structure Interaction on Site Defense (SD) Type Semi-Buried Structures. Part I: Development of Soil Island Procedures and Associated Free-Field Calculations", HND-TR-75-20-ED-CS, Weidlinger Associates, New York, New York, 1975.
17. I. Nelson and J. Isenberg, "Soil Island Approach to Structure-Media Interaction", Second International Conference on Numerical Methods in Geomechanics, Virginia Polytechnic Institute and State University, June 1976.
18. J. Isenberg and D.K. Vaughan, "Three Dimensional Nonlinear Analysis of Soil-Structure Interaction in a Nuclear Power Plant Containment Structure", Second International Conference on Microzonation, San Francisco, November 1978.
19. I.S. Sandler, F.L. DiMaggio and G.Y. Baladi, "Generalized Cap Model for Geological Materials", J. Geotech. Engrg. Div., A.S.C.E., Vol. 102, No. GT7, July 1976.
20. I.S. Sandler, and F.L. DiMaggio, "Anisotropy in Elastic-Plastic Models of Geological Materials", Weidlinger Associates, New York, New York, (Presented at the Defense Nuclear Agency Meeting of Material Properties for Ground Motion Calculations), Park City, Utah, November 1973.
21. F.L. DiMaggio, and I.S. Sandler, "The Effect of Strain Rate on the Constitutive Equations for Rocks", Report No. DNA2801T, Weidlinger Associates, New York, New York, October 1971.
22. D. Rubin, and I.S. Sandler, "Development of a High Pressure Cap Model for Use in Computations of Ground Shock from Subsurface Explosions". Weidlinger Associates, New York, New York, December 1976.
23. R.W. Clough, "Comparison of Three Dimensional Finite Elements", Proceedings of the Symposium on Applications of Finite Element Methods in Civil Engineering, Vanderbilt University, Nashville Tennessee, pp. 1-26, November 1969.
24. L.D. Bertholf and S.E. Benzley, "TOODY II, A Computer Program for Two-Dimensional Wave Propagation", SC-RR-68-41, Sandia Laboratories, Albuquerque, New Mexico, November 1968.
25. R.E. Kronauer, "Oscillations", in Handbook of Applied Mathematics, edited by C.E. Pearson, Van Nostrand Reinhold Company, New York, pp. 711-760, 1974.

26. H. Fujii, "Finite Element Schemes: Stability and Convergence", in Advances in Computational Methods in Structural Mechanics and Design, UAH Press, Hunstville, Alabama, pp. 201-218, 1972.
27. G.E. Forsythe and W.R. Wasow, Finite Difference Methods for Partial Differential Equations, John Wiley and Sons, Inc., New York, 1960.
28. H.L. Brode, "Height of Burst Effects at High Overpressures", DASA 2506, The Rand Corporation, Santa Monica, California, July 1970.
29. M.J. Archuleta, Displaying Complex Three Dimensional Objects, Lawrence Livermore Laboratory, L-73 Livermore, California.
30. W.M. Newman and R.F. Sproull, Principles of Interactive Computer Graphics, McGraw Hill Book Company, pp. 537-574, 1973.

APPENDIX C

FILES AND DATA STRUCTURES

All data files are maintained on external (to SCM) storage devices. The FORTRAN file tape number, by which a data file is referenced, is the default tape number.

C-1 Data Structure

The data structure for the MC is based upon the zone partition of the elements. Data associated with an element-related quantity (e.g. stress) are stored zone by zone one record per zone. Since nodes on zone boundaries (zone interface nodes) are, in general, associated with more than one zone, data related to a nodal quantity (e.g. velocity) are broken down into zone corner, edge, face, and interior data sets. A corner data set is stored as one record and an edge (face) data set is stored edge by edge (face by face) one record per edge (face). A node interior data set is stored zone by zone one record per zone.

C-2 Random Data Files

Because element and nodal data records are not referenced sequentially when processing on a zone basis, element and nodal data files are stored as random files. On the CDC 7600 all random files reside in large core memory (LCM).

The record structure of the coordinate file (tape 15), the force file (tape 20) and the velocity file (tape 40) is the same. The first record is zone corner data. It is followed in turn by zone x-edge, y-edge, z-edge, x-face, y-face, z-face, and interior data records (see Table C-1). The force file is an exception in that it does not have interior data records.

The stress file (tape 30) and the element information file (tape 32) are stored one record per zone in the order of increasing zone number. These files are random because of subcycling.

The random boundary condition file (tape 6) consists of N equal length records. The number of records N is a function of the time content of the boundary velocity input and the integration time step size. This file is handled in a circular fashion.

C-3 Sequential Data Files

There are four primary sequential files: applied boundary velocity file (tape 5), computational results output file (tape 3), general output file (tape 2) and SCM common file (tape 50). Several temporary sequential files also exist: processor input file (tape 1), region corner coordinate file (tape 11), region x-edge coordinate directive file (tape 12), y-edge file (tape 13), and z-edge file (tape 14). Each processor requires a different tape 1 input file. In addition to the main input file (tape 1), various alternate input files may be required by a given processor.

During program execution all sequential files are stored on disk.

Restart Files

Only two files are required for restart purposes: a LCM image file (tape 60) and the SCM common file (tape 50). It should be noted, however, that the applied velocity boundary condition file (tape 5) must be accessible to the CLEAN UP processor and each time processor. Also, the results output file (tape 3) for the CLEAN UP processor and the first execution of PROSSOR must be the same file.

The LCM image file contains a complete copy of the user's LCM data space, i.e., tape 6, tape 15, tape 20, tape 30, tape 32, and tape 40. The SCM common file contains a copy of active blank common, all labeled common necessary to restart, and two restart error-processing parameters.

TABLE C-1

RECORD STRUCTURE OF NODE DATA FILES

Number of Records	Contents of Records
1	Zone Corner Data
* NXZ x (NYZ+1) x (NZZ+1)	Zone x-Edge Data
NYZ x (NXZ+1) x (NZZ+1)	Zone y-Edge Data
NZZ x (NXZ+1) x (NYZ+1)	Zone z-Edge Data
(NXZ+1) x NYZ x NZZ	Zone x-Normal Face Data
(NYZ+1) x NXZ x NZZ	Zone y-Normal Face Data
(NZZ+1) x NXZ x NYZ	Zone z-Normal Face Data
NXZ x NYZ x NZZ	Zone Interior Data

* $N \propto Z$ = Number of Zones in α Direction ($\alpha = X, Y, Z$)

TABLE C-2

APPLIED BOUNDARY VELOCITY FILE (Tape 5)

Record Number	Number of Words	Record Description
1	8	Boundary Velocity Title
2	9	Upstream boundary radius Downstream boundary radius Distance between horizontal base data stations Number of horizontal base data stations (NHS) Number of vertical data stations (NVS) Boundary velocity time increment Boundary velocity starting time Number of boundary velocity data records (NDR) Boundary velocity termination time
3	NVS	Vertical data station depths
4 thru 3 + NDR	1 + Data Length*	Boundary velocity data records: Time Upstream vertical data stations velocities (2 x (NVS-1) words) Horizontal base data stations velocities (2 x NHS words) Downstream vertical data stations velocities (2 x (NVS-1) words)

* Data Length = $2 \times (2 \times (NVS-1) + NHS)$

- Notes:
1. For each boundary data station the horizontal and vertical components of velocity are given, respectively.
 2. Both upstream and downstream vertical data stations velocities are stored in the order of increasing depth.
 3. The horizontal base data stations velocities are stored in the order of increasing range.
 4. The bottom upstream vertical data station is the first horizontal base data station.
 5. The bottom downstream vertical data station is the last horizontal base data station.

APPENDIX D

CDC UPDATE AND LIBGEN FILES

D-1 UPDATE Files

All TRANAL routines are maintained as an UPDATE program sequential library file (TRNLPL). Each routine is identified by a DECK directive. For all routines the DECK name and the routine name are the same. The processor driver programs and all standard UPDATE directives to the TRNLPL file are stored on the UPDATE file TRNLDIR.

Routines that are inherently CDC computer facility dependent (e.g. random file IO transfer routines) are not considered part of the TRANAL system of routines. These facility-dependent routines are maintained as a separate UPDATE program library file (TPL1).

A given CDC computer facility may offer certain execution time operational conveniences (e.g. dynamic length allocation of blank common). The routines that implement these operational conveniences are put on the TPL1 file. The corresponding UPDATE directives to the TRNLPL file are stored on the UPDATE file TDIR1.

All PL (program library) designated UPDATE files use the default master control character (*); however, the DIR (directive) files use dollar (\$), see Section 12 for example. The directives on the DIR type UPDATE files are asterisk (*) prefixed directives.

D-2 Library Files

A compiled binary object code image of each PL type UPDATE file is kept on separate LIB designated files in system library format. For example, the library image of TRNLPL is TRNLLIB and the image of TPL1 is TLIB1. These LIB files are recognized by the loader as system compatible libraries. Thus, the loader can search these files directly for unsatisfied external references.

APPENDIX E

DATA MANAGEMENT LIBRARYE-1 Introduction

The data management library (DML) is a name-oriented family of routines which facilitates (a) the manipulation of data arrays in blank common, and (b) the transmission of data records between external storage and blank common. Blank common is called the dynamic array (DA). The DML uses the control arrays NAME, I POINT, N TYPE, and MODE which reside in the labeled common block /DTA MGR/.

Only arrays are stored in the DA. The dimensions of an array are located in front of the array with the slowest varying dimension next to the first array element. The Hollerith name, the DA pointer, the dimensionality, and the general characteristics of each array are stored in the NAME, I POINT, N TYPE, and MODE arrays, respectively. The DA pointer is the relative address in blank common of the first element of an array. The general characteristics of an array is described by a Hollerith word such as 10HHOLLERITH, 10HIO BLOCK, 10HINT OR REL, etc. Scalars are not stored in the DA. Instead, a scalar's value is stored in I POINT, its Hollerith name is placed in NAME, and the corresponding values of N TYPE and MODE are set to zero.

E-2 Library Routines

The name (entry points), calling sequence, input, output or action, and error checks are given for each DML routine. Throughout the list of the DML routines the following variables are used:

H NAME - a Hollerith array, IO block, or scalar name.
 OPTION - one of the following I/O options: 10HREAD, 10HWRITE, 10HREWRITE.
 NO UNIT - an I/O unit number.
 ARRAY - a data array or IO block.
 LENGTH - length of ARRAY.
 I NAME - location of H NAME in the DML control array NAME.
 J NAME - the DA pointer of array with name H NAME.
 NBR NAM - the number of data array names in an IO block.
 NBR REC - the number of record numbers in an IO block.

An IO block is a one dimensional array of length $3 + \text{NBR NAM} + \text{NBR REC}$. IO blocks are used to facilitate the input/output transfer of random file data records which are associated with named DA arrays. A typical IO block array is made up of the following words:

random file Hollerith unit name,
 NBR NAM,
 data array Hollerith names (NBR NAM words),
 NBR REC, and
 record numbers (NBR REC words).

The following is an alphabetical list of the DML routines.

10 BLK: Performs random file record input/output transfers under control of an IO block.

Calling sequence: CALL 10 BLK (H NAME, OPTION)

Input : H NAME - IO block name.
 OPTION - input/output option.
 Action : Random file data records input/output transfer.
 Errors : 1. IO block is not defined.
 2. A data array is not defined.
 3. NBR NAM = 1, NBR REC > 1, and data array is one dimensional.
 4. NBR NAM = 1 and NBR REC is not equal to slowest varying dimension of data array.

IO RANM: Performs random file record input/output transfer.
 Calling sequence: CALL IO RANM (OPTION, NO UNIT, ARRAY, LENGTH, NO RCRD)

Input : OPTION - input/output option.
 NO UNIT - random file unit number.
 ARRAY - record location.
 LENGTH - record length.
 NO RCRD - record number.
 Action : Random file data record input/output transfer.
 Error : OPTION is invalid.

IO SEQL: Performs sequential file record input/output transfer.
 Calling sequence: CALL IO SEQL (OPTION, NO UNIT, ARRAY, LENGTH)

Input : OPTION - input/output option.
 NO UNIT - sequential file unit number.
 ARRAY - record location.
 LENGTH - record length.
 Action : Sequential file data record input/output transfer.
 Error : OPTION is invalid.

LD MODE: Loads the general characteristics of an array into the DML control array MODE.

Calling sequence: CALL LD MODE (H NAME, H MODE)
 Input : H NAME - array name.
 H MODE - Hollerith array characteristic.
 Action : H MODE is loaded into the MODE array element corresponding to H NAME.
 Error : H NAME is not defined.

LD SCLR (LD UNIT): Loads the name and value of a scalar into the DML control arrays NAME and I POINT, respectively.

Calling sequence: CALL LD SCLR (H NAME, VALUE)
 Input : H NAME - scalar name.
 VALUE - value of scalar.
 Action : H NAME is loaded into array NAME, VALUE is loaded into array I POINT, and the corresponding elements of arrays N TYPE and MODE are set to zero.
 Errors : 1. H NAME already exist.
 2. NAME array was overrun.

LOC DTA (I GT INT, LOC BLK): Locates an array in the DA.

Calling sequence: J NAME = LOC DTA (H NAME)

Input : H NAME - array name.

Output : J NAME - array DA pointer.

Error : H NAME is not defined.

LOC NAM: Locates a name in the DML control array NAME.

Calling sequence: I NAME = LOC NAM (H NAME)

Input : H NAME - array or scalar name.

Output : I NAME - location of H NAME in array NAME.

Default: If H NAME is not defined, I NAME is set to zero.

MSG XIT: Prints an error message in the dayfile and on system output (tape 2), writes traceback information on system output, and returns control to the system.

Calling sequence: CALL MSG XIT (FORMAT, ITEM 1, ITEM 2)

Input : FORMAT - error message format.

ITEM 1 - error message output entity (optional).

ITEM 2 - error message output entity (optional).

Action : An error message is printed in the dayfile and on tape 2, traceback information is written on system output, and execution is terminated.

NAM LNG: Determines the length of an array.

Calling sequence: LENGTH = NAM LNG (I NAME)

Input : I NAME - location of array name in the NAME array.

Output : LENGTH - length array.

NAM MOD: Determines the general characteristics of an array.

Calling sequence: MODE H = NAM MOD (I NAME)

Input : I NAME - location of array name in the NAME array.

Output : MODE H - Hollerith array characteristic.

NAM PTR: Determiniens the DA pointer of an array.

Calling sequence: J NAME = NAM PTR (I NAME)

Input : I NAME - location of arra_ name in the NAME array.

Output : J NAME - array DA pointer.

NAM TYP: Determines the dimensionality of an array.

Calling sequence: NBR DIM = NAM TYP (I NAME)

Input : I NAME - location of array name in the NAME array.

Output : NBR DIM - number of array dimensions.

OPN FIL: Opens a random file.

Calling sequence: CALL OPN FIL (NO UNIT, H NAME, LENGTH)

Input : NO UNIT - random file unit number.

H NAME - index array name.

LENGTH - number of records in file.

Action : The index array is reserved in the DA (if necessary) and the file is opened.

RES DTA: DA space for an array is reserved.

Calling sequence: CALL RES DTA (H NAME, LD1, LD2, LD3, LD4, LD5)

Input : H NAME - array name.

LD1 - fastest varying array dimension.

LD2 }
thru } - array dimensions (optional).
LD5 }

Action : H NAME, the array DA pointer, the number of dimensions in the CALL, and IOHINT OR REL are placed in the DML control arrays NAME, I POINT, N TYPE and MODE, respectively. The dimensions in the CALL are placed in the DA just ahead of the space reserved for the array.

Errors : 1. H NAME already exist.
2. No dimension information is given in CALL.
3. NAME array was overrun.
4. array length is negative.
5. the DA was overrun.

RES IOB: DA space for an IO block array is reserved.

Calling sequence: CALL RES IOB (H NAME, UNT NAM, NBR NAM, NBR REC)

Input : H NAME - IO block array name.

UNT NAM - Hollerith unit name.

NBR NAM - number of data array names.

NBR REC - number of record numbers.

Action : The IO block is reserved by RES DTA as a one dimensional array with dimension 3 + NBR NAM + NBR REC. UNT NAM, NBR NAM, and NBR REC are placed in the IO block array.

Error : NBR NAM > 1 and NBR REC ≠ NBR NAM.

RUB OUT: All information associated with an array (or scalar) is removed from the DML control arrays and the array is deleted from the DA.

Calling sequence: CALL RUB OUT (H NAME)

Input : H NAME - array (or scalar) name.

Action : All data associated with H NAME is removed from the DML control arrays and the DA. The control arrays and the DA are closed up, and the array DA pointers in NAME are reset.

Error : H NAME is not defined.

STR IOL: Loads data array names into an IO block.

Calling sequence: CALL STR IOL (H NAME, NM1, NM2, NM3, NM4,
NM5, NM6, NM7, NM8)

Input : H NAME - IO block name.

NM1 - data array Hollerith name.

NM2 }
thru } - data array Hollerith names (optional).
NM8 }

Action : The data array names in the CALL are loaded into the IO block array H NAME.

Error : The number of data array names in the CALL is not equal to NBR NAM specified in the IO block.

STR IOR: Loads random file record numbers into an IO block.

Calling sequence: CALL STR IOR (H NAME, NR1, NR2, NR3, NR4,
NR5, NR6, NR7, NR8)

Input : H NAME - IO block name.
 NR1 - record number.
 NR2 }
 thru } - record numbers (optional).
 NR8 }
 Action : The record numbers in the CALL are loaded into the IO block
 array H NAME.
 Error : The number of record numbers in the CALL is not equal to
 to NBR REC specified in the IO block.

In conjunction with the DML routines listed above, a set of utility
 routines are also available. These routines are useful for (a) debugging,
 (b) data consistency checking, (c) data examination, and (d) I/O and data
 manipulation. An alphabetical list of these routines are given below.

CHK MGR: The DML control arrays are checked for consistency.

DMP ALL: The active part of the DA is output.

DMP ETA: Outputs a named array according to an integer or a real format.

DMP HOL: Outputs a Hollerith array.

DMP IOB: Outputs an IO block array.

DMP MGR: The DML control arrays are output.

DMP OCT: A named array is output in an octal format.

EXT DTA: The slowest varying dimension of a named array is increased
 and the array's DA data space is extended accordingly.

REC SQC: Copies records from one sequential file to another.

RUB DTG: A group of named arrays are removed from the DA and their
 DML control array information is eliminated.

WRT ARY: Outputs an array according to a specified format.

DISTRIBUTION LIST

DEPARTMENT OF DEFENSE

Assistant to the Secretary of Defense
Atomic Energy
ATTN: Executive Assistant

Defense Advanced Rsch. Proj. Agency
ATTN: TIO

Defense Intelligence Agency
ATTN: RDS-3A
ATTN: DB-4C2, B. Morris
ATTN: DB-4C1

Defense Nuclear Agency
ATTN: STSP
ATTN: DDST
ATTN: SPSS
2 cy ATTN: SPAS
4 cy ATTN: TITL

Defense Technical Information Center
12 cy ATTN: DD

Field Command
Defense Nuclear Agency
ATTN: FCTMOF
ATTN: FCPR

Field Command
Defense Nuclear Agency
Livermore Division
ATTN: FCPR

Joint Strategic Tgt. Planning Staff
ATTN: JSTPS/JLA, R. Haag
ATTN: JLTW-2
ATTN: NRI-STINFO, Library

Undersecretary of Defense for Rsch. & Engrg.
ATTN: Strategic & Space Systems (OS)

DEPARTMENT OF THE ARMY

RMD Advanced Technology Center
Department of the Army
ATTN: ATC-T
ATTN: 1CRDABH-X

Chief of Engineers
Department of the Army
ATTN: DAEN-RDM
ATTN: DAEN-MCE-D

Construction Engineering Rsch. Lab.
Department of the Army
ATTN: CERL-SOI-L

Deputy Chief of Staff for Rsch. Dev. & Acq.
Department of the Army
ATTN: DAMA-CSS-N

DEPARTMENT OF THE ARMY (Continued)

Harry Diamond Laboratories
Department of the Army
ATTN: DELHD-I-TL
ATTN: DELHD-N-P

U.S. Army Armament Material Readiness Command
ATTN: MA Library

U.S. Army Ballistic Research Labs
ATTN: DRDAR-BLT, W. Taylor
ATTN: DRDAR-BLE, J. Keefer

U.S. Army Communications Command
ATTN: Technical Reference Division

U.S. Army Engineer Center
ATTN: ATZA

U.S. Army Engineer Div., Huntsville
ATTN: HNDED-SR

U.S. Army Engineer Div., Ohio River
ATTN: ORDAS-L

U.S. Army Engr. Waterways Exper. Station
ATTN: Library
ATTN: WESSD, G. Jackson
ATTN: WESSA, W. Flathau
ATTN: WESSE, L. Ingram
ATTN: J. Strange
ATTN: WESSS, J. Ballard
ATTN: F. Brown

U.S. Army Material & Mechanics Rsch. Ctr.
ATTN: Technical Library

U.S. Army Materiel Dev. & Readiness Cmd.
ATTN: DRXAM-TL

U.S. Army Missile Command
ATTN: DRDMI-XS
ATTN: RSIC

U.S. Army Mobility Equip. R&D Cmd.
ATTN: DRDME-WC

U.S. Army Nuclear & Chemical Agency
ATTN: Library

U.S. Army War College
ATTN: Library

U.S. Military Academy
ATTN: R. La Frenz

DEPARTMENT OF THE NAVY

David Taylor Naval Ship R & D Ctr.
ATTN: Code L42-3
ATTN: Code 177, E. Palmer
ATTN: Code 1700, W. Murray

DEPARTMENT OF THE NAVY (Continued)

Naval Construction Battalion Center
Civil Engineering Laboratory
ATTN: Code L51, J. Crawford
ATTN: Code L51, R. Odello
ATTN: Code L51, S. Takahashi
ATTN: Code L51, W. Shaw

Naval Electronic Systems Command
ATTN: PME 117-21

Naval Explosive Ord. Disposal Fac.
ATTN: Code 504, J. Petrousky

Naval Facilities Engineering Command
ATTN: Code 09M22C

Naval Ocean Systems Center
ATTN: Code 4471

Naval Postgraduate School
ATTN: Code 1424, Library
ATTN: Code 0142, Library

Naval Research Laboratory
ATTN: Code 8403, R. Belsham
ATTN: Code 2627

Naval Sea Systems Command
ATTN: SEA-09G53

Naval Ship Engineering Center
ATTN: Code 09G3

Naval Surface Weapons Center
ATTN: Code R14
ATTN: Code R10
ATTN: Code U401, M. Kleinerman
ATTN: Code F31

Naval Surface Weapons Center
ATTN: Technical Library & Info. Services Br.

Naval War College
ATTN: Code E-11, Technical Service

Naval Weapons Center
ATTN: Code 3263, J. Bowen
ATTN: Code 266, C. Austin
ATTN: Code 233

Naval Weapons Evaluation Facility
ATTN: Code 10
ATTN: R. Hughes

Office of Naval Research
ATTN: Code 715
ATTN: Code 474, N. Perrone
ATTN: Code 463, J. Heacock

Strategic Systems Project Office
Department of the Navy
ATTN: NSP-43

DEPARTMENT OF THE AIR FORCE

Aerospace Defense Command
Department of the Air Force
ATTN: XPX

DEPARTMENT OF THE AIR FORCE (Continued)

Air Force Armament Laboratory
ATTN: DLYV, J. Collins

Air Force Geophysics Laboratory
ATTN: LWV, K. Thompson

Air Force Institute of Technology, Air University
ATTN: Commander
ATTN: Library

Air Force Office of Scientific Research
ATTN: NA, B. Wolfson

Air Force Weapons Laboratory
Air Force Systems Command
ATTN: DED
ATTN: DES-G, S. Melzer
ATTN: DE, M. Plamondon
ATTN: SUL
ATTN: DES-C, R. Henny

Ballistic Missile Office
Air Force Systems Command
ATTN: DEB

Ballistic Missile Office
Air Force Systems Command
ATTN: MNNH
ATTN: MMH

Deputy Chief of Staff
Research, Development, & Acq.
Department of the Air Force
ATTN: AFRUQSM
ATTN: R. Steere

Foreign Technology Division
Air Force Systems Command
ATTN: NIIS, Library

Headquarters Space Division
Air Force Systems Command
ATTN: DYS

Rome Air Development Center
Air Force Systems Command
ATTN: TSLD
ATTN: Commander
ATTN: RBES, R. Mair

Strategic Air Command
Department of the Air Force
ATTN: NRI-STINFO, Library

United States Air Force Academy
ATTN: DFCEM, W. Fluhr

DEPARTMENT OF ENERGY

Department of Energy
Albuquerque Operations Office
ATTN: CTID

Department of Energy
ATTN: Document Control for OMA/RD&T

DEPARTMENT OF ENERGY CONTRACTORS

Lawrence Livermore Laboratory

ATTN: Document Control for L-90, D. Norris
ATTN: Document Control for L-96, L. Woodruff
ATTN: Document Control for Technical
Information Dept. Library
ATTN: Document Control for L-200, J. Cortez
ATTN: Document Control for L-90, R. Dong

Los Alamos Scientific Laboratory

ATTN: Document Control for G. Spillman
ATTN: Document Control for M/S632, T. Dowler
ATTN: Document Control for RMS 364

Oak Ridge National Laboratory

ATTN: Document Control for Central Research
Library
ATTN: Civil Def. Res. Proj.

Sandia Laboratories

Livermore Laboratory

ATTN: Document Control for Library &
Security Classification Div.

Sandia Laboratories

ATTN: Document Control for L. Vortman
ATTN: Document Control for W. Herrmann
ATTN: Document Control for A. Chabai

OTHER GOVERNMENT AGENCIES

Department of the Interior

Bureau of Mines

ATTN: Technical Library

Department of the Interior

U.S. Geological Survey

ATTN: D. Roddy

NASA

Ames Research Center

ATTN: R. Jackson

U.S. Nuclear Regulatory Commission

ATTN: R. Whipp for Div. of Security for L. Shao

DEPARTMENT OF DEFENSE CONTRACTORS

Acurex Corp.

ATTN: J. Stockton

Aerospace Corp.

ATTN: L. Selzer
ATTN: P. Mathur

2 cy ATTN: Technical Information Services

Agbabian Associates

ATTN: M. Agbabian
ATTN: C. Bagge

Analytic Services, Inc.

ATTN: G. Hesselbacher

Applied Theory, Inc.

2 cy ATTN: J. Trulio

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

Artec Associates, Inc.

ATTN: S. Gill

AVCO Research & Systems Group

ATTN: W. Broding
ATTN: Library, A8:0

BDM Corp.

ATTN: A. Lavagnino
ATTN: Corporate Library
ATTN: T. Neighbors

BDM Corp.

ATTN: R. Hensley

Bell Telephone Labs

ATTN: J. White

Boeing Co.

ATTN: R. Holmes
ATTN: Aerospace Library
ATTN: J. Wooster
ATTN: R. Dyrda

California Institute of Technology

ATTN: T. Ahrens

California Research & Technology, Inc.

ATTN: K. Kreyenhagen
ATTN: S. Shuster
ATTN: Library

California Research & Technology, Inc.

ATTN: D. Orphal

Calspan Corp.

ATTN: Library

Center for Planning & Rsch., Inc.

ATTN: R. Shnider

Civil Systems, Inc.

ATTN: J. Bratton

University of Denver, Colorado Seminary

ATTN: Sec. Officer for J. Wisotski

EG&G Washington Analytical Services Center, Inc.

ATTN: Director
ATTN: Library

Electric Power Research Institute

ATTN: G. Sliter

Electromechanical Sys. of New Mexico, Inc.

ATTN: R. Shunk

Eric H. Wang

Civil Engineering Rsch. Fac.

ATTN: N. Baum
ATTN: D. Calhoun

Franklin Institute

ATTN: Z. Zudans

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

Gard, Inc.
ATTN: G. Neidhardt

General Dynamics Corp.
ATTN: K. Anderson

General Electric Co.
ATTN: M. Bortner

General Electric Co.
ATTN: A. Ross

General Electric Company-TEMPO
ATTN: DASIAC

General Research Corp.
ATTN: B. Alexander

H-Tech Labs, Inc.
ATTN: B. Hartenbaum

Honeywell, Inc.
ATTN: T. Helvig

IIT Research Institute
ATTN: Documents Library
ATTN: A. Longinow

University of Illinois, Consulting Services
ATTN: N. Newmark
ATTN: W. Hall
ATTN: J. Haltiwanger

Institute for Defense Analyses
ATTN: Director
ATTN: Classified Library

J. H. Wiggins Co., Inc.
ATTN: J. Collins

Kaman Avidyne
ATTN: N. Hobbs
ATTN: Library
ATTN: E. Criscione

Kaman Sciences Corp.
ATTN: D. Sachs
ATTN: Library

Karagozian and Case
ATTN: J. Karagozian

Lockheed Missiles & Space Co., Inc.
ATTN: TIC, Library

Lockheed Missiles and Space Co., Inc.
ATTN: T. Geers

Management Science Associates
ATTN: K. Kaplan

Martin Marietta Corp.
ATTN: A. Cowan
ATTN: G. Fotieo

Martin Marietta Corp.
ATTN: J. Donathan

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

University of Massachusetts
ATTN: W. Nath

McDonnell Douglas Corp.
ATTN: R. Halprin

Merritt CASES, Inc.
ATTN: J. Merritt
ATTN: Library

Meteorology Research, Inc.
ATTN: W. Green

Mitre Corp.
ATTN: Director

University of New Mexico
ATTN: G. Triandafalidis

University of Oklahoma
ATTN: J. Thompson

Pacifica Technology
ATTN: G. Kent
ATTN: R. Bjork
ATTN: R. Allen

Pacific-Sierra Research Corp.
ATTN: H. Brode

Physics International Co.
ATTN: F. Sauer
ATTN: E. Moore
ATTN: Technical Library

University of Pittsburgh
ATTN: M. Willims, Jr.

R&D Associates
ATTN: R. Port
ATTN: C. MacDonald
ATTN: P. Rausch
ATTN: J. Lewis
ATTN: Technical Information Center

R&D Associates
ATTN: H. Cooper

Rand Corp.
ATTN: A. Laupa
ATTN: Library
ATTN: C. Mow

Science Applications, Inc.
ATTN: Technical Library

Science Applications, Inc.
ATTN: S. Oston

Science Applications, Inc.
ATTN: R. Hoffmann
ATTN: D. Maxwell
ATTN: D. Bernstein

Science Applications, Inc.
ATTN: B. Chambers III
ATTN: W. Layson
ATTN: G. Binninger

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

Southwest Research Institute

ATTN: W. Baker

ATTN: A. Wenzel

SRI International

ATTN: G. Abrahamson

ATTN: W. Wilkinson

Systems, Science & Software, Inc.

ATTN: Library

ATTN: T. Riney

ATTN: D. Grine

ATTN: T. McKinley

Teledyne Brown Engineering

ATTN: J. Ravenscraft

Terra Tek, Inc.

ATTN: Library

Tetra Tech, Inc.

ATTN: L. Hwang

ATTN: Library

Texas A&M University System

ATTN: H. Coyle

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

TRW Defense & Space Sys. Group

ATTN: Technical Information Center

ATTN: P. Bhutta

ATTN: A. Feldman

2 cy ATTN: P. Dai

TRW Defense & Space Sys. Group

ATTN: E. Wong

ATTN: G. Hulcher

ATTN: F. Pieper

Weidlinger Assoc., Consulting Engineers

ATTN: J. McCormick

ATTN: M. Baron

ATTN: J. Baylor

ATTN: J. Wright

ATTN: C. Chung

Weidlinger Assoc., Consulting Engineers

ATTN: J. Isenberg

Westinghouse Electric Corp.

ATTN: W. Volz

Dist-6