AD-A011 134

EVALUATION OF EXISTING STRUCTURES

C. K. Wiehle

Stanford Research Institute

Prepared for:

Defense Civil Preparedness Agency

December 1974

Final Report December 1974

EVALUATION OF EXISTING STRUCTURES

CONTRACT DAHC20-71-C-0292
DCPA Work Unit 1154I

STANFORD RESEARCH INSTITUTE
Menlo Park, California 94025 · U.S.A.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (none)	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER AD A011 134
4. TITLE (and Subtitle) EVALUATION OF EXISTING STRUCTURES		5. TYPE OF REPORT & PERIOD COVERED Final Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) C. K. Wiehle		8. CONTRACT OR GRANT NUMBER(s) DAHC20-71-C-0292
9. PERFORMING ORGANIZATION NAME AND ADDRESS Stanford Research Institute Facilities and Housing Research Department		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS DCPA Work Unit 1154I
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Civil Preparedness Agency Washington, D.C. 20301		12. REPORT DATE December 1974 / 13. NO. OF PAGES 195
		15. SECURITY CLASS. (of this report) Unclassified
14. MONITORING AGENCY NAME & ADDRESS (if diff. from Controlling Office)		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this report)

Approved for public release and sale; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from report)

18. SUPPLEMENTARY NOTES

PRICES SUBJECT TO CHANGE

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Structures
Blast
Wall and Floor Analysis
Dynamic Analysis

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The objective of the overall research program is to develop an evaluation procedure applicable to existing NSS-type structures and private homes. Past efforts have been concerned with examining exterior walls; window glass; steel frame connections; applications to actual buildings; reinforced concrete floor systems, including restrained slabs; wood-joist floors; and the dynamic inelastic analysis of a steel ·.. building. Since this is the final report in this effort, a summary of the evaluation procedure for existing structures is presented in the

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19. KEY WORDS (Continued)

20 ABSTRACT (Continued)

report. Also included is the flow chart developed for a computer program to analyze a building subsystem; i.e., the dynamic response and collapse of all exterior and interior walls on one floor level of a building. An analysis made to determine the blast resistance of basement walls of Emergency Operating Centers is presented. Finally, the report contains a complete listing of all computer programs developed during the project for analyzing the dynamic response and collapse of wall and floor elements.

SUMMARY

Introduction

The objective of this investigation was to develop an evaluation procedure for determining the blast protection afforded by existing NSS-type structures and private residences. The procedure developed consists basically of (1) a method for determining the air blast loading on the structure and structural elements, (2) a method for determining the dynamic structural response up to collapse, and (3) a method for establishing the failure criterion for each structural member of interest. The analytical method used was to establish the resistance function for each wall or floor element by considering the approximate response mode and by assuming that the element was subjected to a uniformly distributed static load. The member was then transformed into an equivalent single-degree-of-freedom dynamic system, and the equation of motion solved on a computer using a numerical integration procedure.

Background

The primary interest from the inception of this study has been the development of an evaluation procedure for analyzing the dynamic response and collapse of the building system. However, the complexity of a comprehensive evaluation procedure for a building system necessitated that important building elements be treated first. Therefore, the initial effort in the program was directed primarily toward the development of mathematical models to analyze the dynamic response and collapse of various types of one-way action walls. Also, the behavior of window glass and steel-frame connections were examined as part of the initial study.

The analytical procedures were then extended to include two-way walls, and a probability approach was incorporated into the evaluation procedure. Next, mathematical models were developed to analyze dynamically loaded reinforced concrete floor systems of various types, and wood-joist floors. During the conduct of the research program, the evaluation procedure has been used to predict the collapse overpressure of a large number of exterior walls and floors over basement areas for existing NSS buildings. As part of this effort, the relative collapse strength of the exterior walls and frame of a multistory steel-frame building was examined.

A summary of the evaluation procedure for existing structures is contained in this final report. Also included is a flow chart for a computer program to analyze a building subsystem, and an analysis of the blast resistance of basement walls located in areaways of Emergency Operating Centers (EOCs). An appendix of the report contains a complete listing of all computer programs developed during the study to analyze dynamically loaded wall and floor elements.

Discussion

The computer programs for the evaluation of existing structures were developed as individual programs to analyze various types of wall and floor elements. Although this approach permitted the analysis of actual structures to be made sooner than would otherwise be possible, as well as being convenient for correlation of analytical models with experimental data, there was a need to develop a computer program to analyze the building system. As the next logical step in the development of an overall building program, a flow chart was developed for a building subsystem. The subsystem selected was all exterior and interior walls on one floor of a building. The report presents the flow chart for a computer program to analyze each wall on a room-by-room basis as the blast wave moves through the building.

The collapse strength of reinforced concrete basement walls was examined to determine the feasibility of retrofitting EOCs with doors to resist the 10-psi blast overpressure level. Although mathematical models have been developed for walls with window openings, the evaluation procedures were not extended to include walls with door openings. Since there was insufficient time to develop a generalized model for calculating the resistance and response of the wall configuration of interest, the following three-phase approach was used:

(1) A detailed yield-line analysis of several specific reinforced concrete walls with door openings was made to establish the static resistance over a limited range of wall widths.

(2) The computer program for analyzing the collapse of wall elements, developed by SRI for DCPA, was used to calculate the resistance of walls without door openings. The results were then compared with those obtained from the yield-line analysis for walls with door openings to determine the feasibility of using the existing computer program to simulate the dynamic response of walls with door openings. If the resistances for the two wall types was found to be not comparable, then it would be necessary to hand calculate a resistance function for each wall case.

(3) An existing finite element computer program was used to analyze the static behavior for a few cases of walls with door openings to determine if shear or stress concentrations could conceivably produce a wall failure not predictable by the other analyses.

The various analyses led to a few general conclusions concerning the collapse of blast loaded reinforced concrete basement walls with door openings and located in areaways. First, for reinforced concrete walls 8-in. thick or thicker, and not over 10-ft high, it is probable that the wall strength of the weakest code-designed wall is sufficient to resist a 10-psi blast loading if the horizontal distance from the edge of the door opening to the areaway support wall is less than approximately 20 in. (L_H = 84 in.).

Second, for 8-in.-thick walls with horizontal distance between door opening and areaway wall greater than 20 in., it will be necessary to strengthen the wall in the vicinity of the door opening so as to upgrade the wall to the 10-psi blast overpressure level.

Third, for reinforced concrete basement walls 12-in thick or thicker, the blast strength can be expected to be approximately equal to or greater than the 10-psi blast overpressure criterion for all wall conditions.

STANFORD RESEARCH INSTITUTE
Menlo Park, California 94025 · U.S.A.

Final Report *December 1974*

EVALUATION OF EXISTING STRUCTURES

By: C. K. WIEHLE
Facilities and Housing Research

Prepared for:

CIVIL DEFENSE PREPAREDNESS AGENCY
WASHINGTON, D.C. 20301

CONTRACT DAHC20-71-C-0292
DCPA Work Unit 1154I

SRI Project 1219-2

DCPA REVIEW NOTICE:

This report has been reviewed in the Civil Defense Preparedness Agency and approved for publication. Approval does not signify that the contents necessarily reflect the views and policies of the Civil Defense Preparedness Agency.

Approved for public release and sale; distribution unlimited.

ABSTRACT

The objective of the overall research program is to develop an evaluation procedure applicable to existing NSS-type structures and private homes. Past efforts have been concerned with examining exterior walls; window glass; steel frame connections; applications to actual buildings; reinforced concrete floor systems, including restrained slabs, wood-joist floors; and the dynamic inelastic analysis of a steel frame building. Since this is the final report in this effort, a summary of the evaluation procedure for existing structures is presented in the report. Also included is the flow chart developed for a computer program to analyze a building subsystem; i.e., the dynamic response and collapse of all exterior and interior walls on one floor level of a building. An analysis made to determine the blast resistance of basement walls of Emergency Operating Centers is presented. Finally, the report contains a complete listing of all computer programs developed during the project for analyzing the dynamic response and collapse of wall and floor elements.

CONTENTS

ILLUSTRATIONS

TABLE

I INTRODUCTION

Under contract to the Defense Civil Preparedness Agency, Stanford Research Institute is developing a procedure for the evaluation of existing structures subjected to nuclear air blast. The objective of the program is to develop an evaluation procedure applicable to existing NSS-type structures and private homes. This report covers the final phase of the program.

Background

The Defense Civil Preparedness Agency has a number of problem areas in which an evaluation procedure for existing structures can be applied. These include:

- Survival and injury predictions
- Debris prediction
- Damage assessment
- Selection of existing structures that provide the best protection
- Selection of existing structures that have a potential for modification to provide blast shelters.

Even with the availability of high-speed computers, it was apparent that the complexity of an overall building evaluation procedure to meet the needs of DCPA could lead to considerable unwarranted computational effort if care was not exercised in the selection of the methodology. Therefore, relatively simplified air blast loading and room-filling procedures, as well as simplified structural response analytical methods, have been used in the evaluation program.

Although the primary interest from the inception of the program has been in the behavior and collapse of the building system, the complexity of a comprehensive evaluation procedure necessitated the establishment of a priority for determining which structural element to investigate first. It is apparent that the collapse of the exterior walls of most buildings is important to the casualties produced. This is especially true for large multistory buildings where the collapse of the exterior and interior walls could result in a large number of casualties through ejection from the building, even if the floors and frame remained intact. Since one of the primary uses of a building evaluation procedure is to provide input for prediction of survival of people located in buildings subjected to nuclear blast, the initial research effort was directed towards the development of a method to determine the response and collapse of exterior wall elements.

Subsequent to the development of the wall evaluation procedure, the procedures for analyzing the collapse of floor systems were developed. Although there were insufficient funds in the program to develop a procedure for evaluating the collapse of structural frames, it was possible to use an available elastic and inelastic computer program to analyze the dynamic response of a steel frame building and estimate the frame collapse overpressure. During the final phase of the research, a computer flow diagram was developed for analyzing a building subsystem; i.e., for predicting the time sequence of collapse of all exterior and interior walls on one floor of a building on a room-by-room basis. However, the computer program could not be written within the level of effort of the contract.

Past reports in this program have been concerned with examining exterior walls (Ref. 1),* window glass (Ref. 2), steel-frame connections (Ref. 3), two-way action walls (Ref. 4), applications to NSS buildings

---

* References are listed after the appendix.

(Refs. 5 and 6), reinforced concrete floor systems (Refs. 7 and 8), and wood-joist floors, and frame analysis (Ref. 8).

Report Organization

Since this is the final report on the research effort, a summary of the evaluation procedure is presented in Section II. A flow chart for the analysis of all walls on one floor level of a building (a building subsystem) is presented in Section III "Building System Program." The analysis of basement walls located in areaways is given in Section IV. During the project for evaluation of existing structures, computer programs were developed for analyzing the dynamic collapse strength of three types of wall elements and five types of floor system elements. The listings for the eight programs are included in the Appendix.

Acknowledgements

The author gratefully acknowledges the assistance and guidance of G. N. Sisson and M. A. Pachuta of the Defense Civil Preparedness Agency during the conduct of this program. Also acknowledged are J. R. Rempel and J. E. Beck of SRI, and Dr. J. L. Bockholt, consultant to SRI, for their contributions to this effort.

II EVALUATION OF EXISTING STRUCTURES

Approach

The overall approach adopted in this study for the evaluation of existing structures subjected to nuclear air blast has been to formulate a procedure for examining the response of a structure over a range of incident overpressure levels to determine the overpressure at which collapse will occur. Basically, the procedure consists of (1) a method for determining the air blast loading on the structure and structural elements, (2) a method for determining the dynamic structural response up to collapse, and (3) a method for establishing the failure criterion for each structural member of interest. An iterative process is employed in which the structural response can be examined for various levels of incident overpressure and compared with a failure criterion to predict the overpressure level at which collapse of each member will occur.

Wall and Floor Evaluation Procedure

Introduction

The analytical method used in the research study was to establish the resistance function for each wall or floor element of interest by considering the approximate response mode and by assuming that the element was subjected to a uniformly distributed static load. The member was then transformed into an equivalent single-degree-of-freedom dynamic system by the use of the transformation factors for the load, resistance, and mass. The equation of motion was then solved on a computer using the numerical integration procedure described in Ref. 9. Although the approach has been to use established analytical procedures wherever possible, it has been necessary to modify and adapt current procedures, as well as develop new methods, for specific uses.

The method followed in developing the wall and floor evaluation procedure was to (1) develop a mathematical model for each element of interest, (2) prepare the computer program, and (3) verify the analytical predictions with the available published test information on the dynamic response and collapse of wall and floor elements.

Although the mathematical models were formulated by using established analytical procedures, as noted in the referenced reports on the evaluation procedure for existing structures, the available published test data were adequate for correlation with only some of the analytical models. However, for other cases* there was a lack of definitive experimental information that adequately described the load-response relationship up to collapse. Although all mathematical models could not be correlated sufficiently with appropriate experimental data, the use of probability functions in the procedures for predicting the incipient collapse overpressure of the elements makes the use of precise resistance functions less critical than would otherwise be the case.

For the evaluation of existing structures, failure implies collapse or disintegration of the structural element. Furthermore, the predicted collapse overpressures calculated are for the incipient collapse of the element, which is defined as that point in the response where the wall or floor can be considered as on the threshold of collapse. The incipient collapse overpressure is just sufficient in magnitude to cause a collapse of the element.

---

* For example, the inelastic response up to collapse of a two-way lightly reinforced concrete wall with a window opening and with vertical in-plane forces acting on the wall.

Wall Analysis

The three basic types of exterior walls considered in the evaluation procedure are unreinforced concrete or masonry unit walls without arching, unreinforced concrete or masonry unit walls with arching, and reinforced concrete walls. The details of the development of the wall evaluation procedures are presented in Ref. 1 for one-way action walls and Ref. 4 for two-way walls.

For unreinforced masonry unit walls without arching and for reinforced concrete walls, resistance functions were developed for the following type of wall support conditions:

- Two-way, simply supported on four edges
- Two-way, fixed on four edges
- Two-way, fixed on vertical edges; simply supported on horizontal edges
- Two-way, simply supported on vertical edges; fixed on horizontal edges
- One-way, simply supported on opposite edges
- One-way, fixed on opposite edges
- One-way propped cantilever
- One-way, cantilever.

For unreinforced walls with arching, resistance functions were developed for one- and two-way action walls with rigid supports.

Floor System Analysis

One of the interests of DCPA has been the possible use of basements of existing NSS structures as blast shelter areas, and therefore the research effort was primarily concerned with developing methods for predicting the collapse of various types of reinforced concrete floor systems.

However, the resistance function for wood-joist floors was also developed. The details of the floor system evaluation procedures are presented in Refs. 7 and 8.

The types of floor elements included in the evaluation procedure are as follows:

- One- and two-way reinforced concrete solid slabs
- Two-way restrained reinforced concrete solid slab
- Reinforced concrete support beam (including T-beam and joist)
- Structural steel support beam (including composite action)
- Reinforced concrete flat slab
- Reinforced concrete flat plate
- Wood-joist floor

Probability Considerations

The analysis of actual building elements subjected to nuclear air blast requires the assumption of values for many of the physical properties of the structure that are unknown and cannot be measured without an unwarranted amount of effort. Similarly, assumptions are also required in the determination of the parameters defining the load acting on the building element. Since precise values cannot usually be specified for many of the parameters that influence the collapse of actual structures, a probabilistic approach was formulated to provide a realistic evaluation of existing structures subjected to nuclear air blast (Ref. 4).

It is apparent that the determination of the incipient collapse overpressure for a given wall or floor depends on a number of variables, at least some of which must be considered to be randomly distributed. Although the probability distribution of these random variables may be determined fairly easily, at least as approximations, the extension of this step to determine the probability distribution of the resulting collapse overpressure is not so easy. Since it was not possible to obtain an exact distribution, it was decided to use Monte Carlo, or simulation, techniques to determine the probability distribution for the incipient collapse overpressure.

This technique uses a set of mathematically simulated wall or floor elements, each of which possesses the characteristics of some real wall or floor to determine an approximate distribution of the incipient collapse overpressure. This set of simulated walls or floors is prepared by selecting the parameters to be varied and determining the values of these parameters by randomly sampling their corresponding probability distribution functions. Each simulated wall or floor is then analyzed by using the deterministic equations developed previously. The results of these analyses provide a probability distribution of the incipient collapse overpressure. It should be noted that the collapse overpressure of a wall or floor element can also be calculated deterministically.

Air Blast Loading

An important factor in the evaluation of existing structures subjected to nuclear air blast is the determination of the pressure-time function on each structural element of interest. This is a complex problem, since, even before the blast wave interacts with the structure, the blast wave is influenced by many factors, such as weapon yield and location, weather conditions, terrain, surface type, and blast shielding. Even if it were assumed that the free-field, pressure-time relationship were known for a blast wave incident on the side of a building, the determination of the loading function on a wall or floor element is difficult because of the interaction processes. The primary difficulty arises because the structural element responds to the differential or net loading, which requires a knowledge of the loading on both the front and back surfaces.

For the evaluation of existing structures subjected to nuclear air blast, it was assumed that the blast wave before interacting with the structure was an ideal Mach waveform propagating radially outward over an ideal reflecting surface. It was also assumed that the duration of positive phase of the dynamic overpressure was equal to that of the side-on overpressure and that the negative phase could be neglected for structural response calculations. The method used to determine the pressure-time function of an exterior wall is presented in detail in Ref. 4, and involves the calculation of an exterior, interior, and net loading.

To calculate the average load-time history on the exterior wall, the conventional air blast loading scheme for a closed rectangular block is used (Ref. 10). For the front face of a building with window openings, the conventional scheme is modified by using the weighted average clearing distance presented in Ref. 11. To calculate the interior pressure build up resulting from the air blast entering the building through openings, the room-filling procedure presented in Ref. 12 is used. For each specific problem, the net wall loading is obtained by a simple summation of the exterior and interior pressure-time histories.

In addition to the ideal air blast loading, the evaluation procedure for exterior walls includes the following loading schemes:

- Triangular load
- Rectangular load
- URS shock tunnel load
- Arbitrary load.

For the dynamic analysis of floor systems subjected to nuclear air blast, two load-time functions were included in the evaluation procedure. The first load type was equal to the free-field blast overpressure, except with a rise time equal to the travel time of the wave front across the floor panel. The second load type was equal to the room-filling pressure resulting from the interaction of an ideal air blast wave with a structure with window openings. In addition, the floor evaluation procedure includes an arbitrary load shape.

It should be noted that although the net load-time function resulting from a nuclear air blast is calculated so as to analyze the dynamic response of a wall or floor element, a description of the net load-time function is not too meaningful for comparing collapse predictions for elements of various structures. Therefore, the predicted collapse overpressures given in this study are the peak incident overpressures of the free-field blast wave that results in collapse of the element.

Applications

As part of an integrated program to develop a survey procedure for all nuclear weapon effects, Research Triangle Institute (RTI) made an initial on-site field survey during November 1970 of five NSS buildings in Detroit, Michigan. The survey was conducted primarily to obtain a complete structural description of buildings that would be adequate for predicting building damage and casualties. The results of the field survey were recorded on forms and included sketches and photographs. A complete copy of this information, together with the building plans, was provided to SRI for analysis of the buildings. The results of the dynamic analysis of the exterior walls of the five Detroit buildings are presented in Ref. 5.

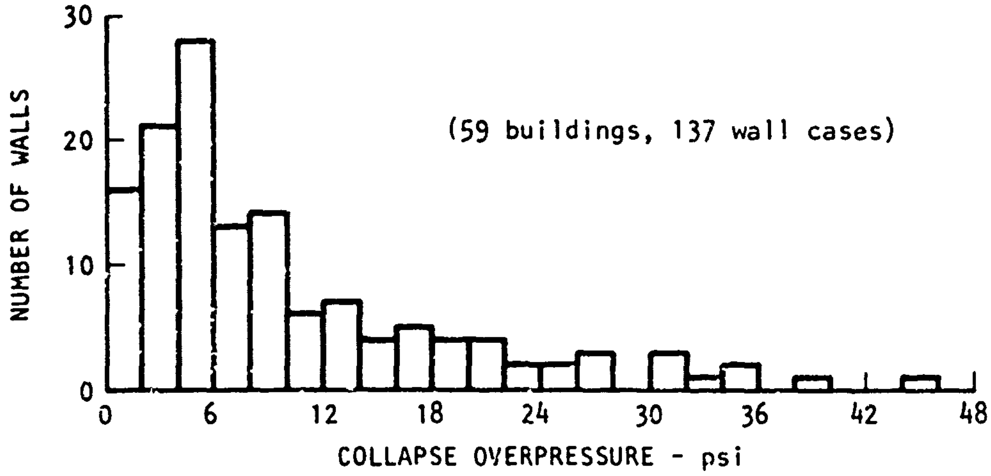
To provide additional input information for the development of the all-effects survey, RTI made a second on-site field survey in July 1971 of five buildings in the vicinity of Greensboro, North Carolina. As in the analysis of the Detroit buildings presented in Ref. 5, two dynamic analyses were made of each of the Greensboro buildings. The first analysis was made using the data obtained during the RTI on-site survey.

A second analysis of the same building was then made independently using data obtained from the actual building plans. This procedure provided a check on the adequacy of the survey technique and the proposed field survey data form, and emphasized areas of possible improvement. The results of the dynamic analysis of the exterior walls of the five Greensboro-High Point buildings are presented in Ref. 6.

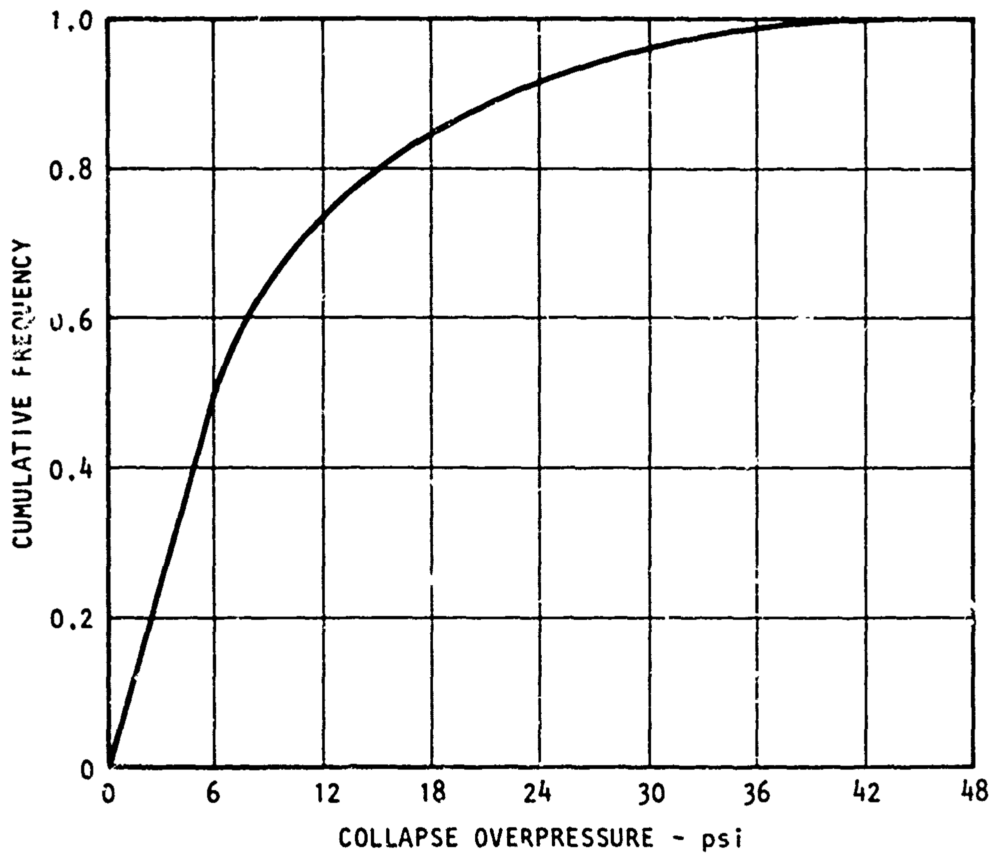
In addition to the two research studies to develop an all-effects shelter survey procedure, RTI also collected data on a national sample of NSS buildings for the Engineering Directorate of DCPA (Ref. 13). Of the 219 NSS buildings comprising the national sample, the SRI evaluation procedure was used to predict the collapse overpressure of the exterior walls for 50 of the buildings and of floors over basement areas for 36 of the buildings. Since the results of the dynamic analyses of the walls and floors of actual buildings are of interest to the evaluation of existing structures program, a short summary of the findings are presented in this report.

Walls

The collapse predictions for the exterior walls of NSS buildings required the dynamic analysis of 137 wall cases. These walls represent 59 NSS buildings, which can be categorized as 15 load-bearing wall buildings, 23 structural steel frame buildings, and 21 reinforced concrete frame buildings. Figure 1 shows a histogram and cumulative frequency distribution of the mean collapse overpressure for the 137-wall population. The data indicate that for the 59 sample buildings, 50 percent of the exterior walls are predicted to have a mean collapse overpressure of 6 psi or less, and 90 percent are predicted to have a mean collapse overpressure of 22 psi or less.



(a) Histogram for all wall cases



(b) Cumulative frequency distribution for all wall cases

FIGURE 1 HISTOGRAM AND CUMULATIVE FREQUENCY DISTRIBUTION OF THE MEAN COLLAPSE OVERPRESSURE FOR THE EXTERIOR WALLS OF 59 BUILDINGS

The effect of the type of frame on the collapse strength of exterior walls is indicated by the cumulative frequency distributions for the wall collapse overpressures shown in Figure 2 for the three major building frame categories. Although the data are considered as insufficient to establish quantitatively the effect of frame type on wall collapse overpressure level, the trends in the data are apparent. The mean values of the collapse overpressures for walls are about 4.5 psi for load-bearing wall buildings, 6 psi for reinforced concrete frame buildings, and 10 psi for steel frame buildings.

Floors

The collapse predictions for the floor systems over basement areas of NSS buildings required the dynamic analysis of 82 floor cases, which represent 36 buildings. Figure 3 shows a histogram and cumulative frequency distribution of the collapse overpressures for all floors. As noted on the figure, the collapse overpressure for floors over basement areas ranged from about 2 to 55 psi, with 50 percent of the floors predicted to collapse at 7 psi or less and 90 percent predicted to collapse at 18 psi or less.

Frame Analysis

A continuing concern in evaluating the collapse overpressure of existing buildings has been the relative blast strength of the exterior walls and frames of multistory buildings. To predict the collapse overpressure of the exterior walls for the existing NSS buildings discussed in the previous subsection, it was assumed that the structural frame did not collapse at a lower overpressure than that predicted for the exterior walls. For weak-walled buildings, such an assumption is reasonable. In fact, it is often assumed for the analysis of blast loaded frame buildings that the exterior walls can be considered as frangible,

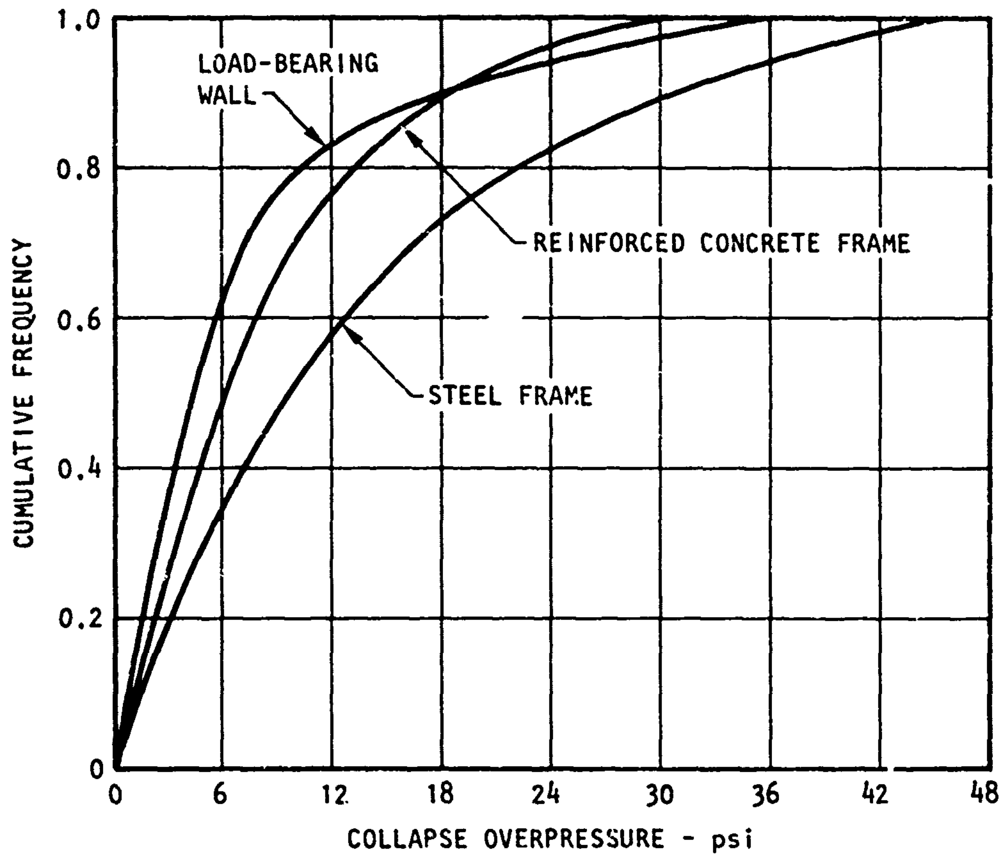
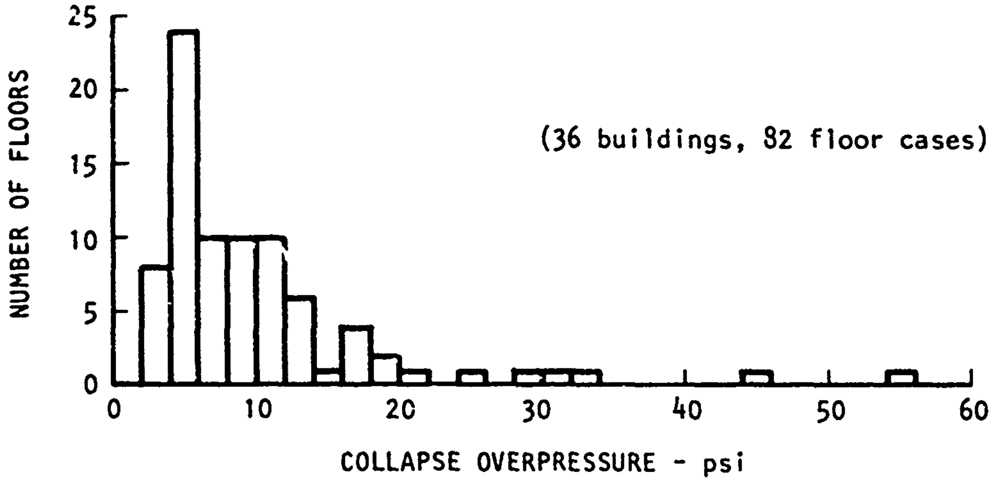
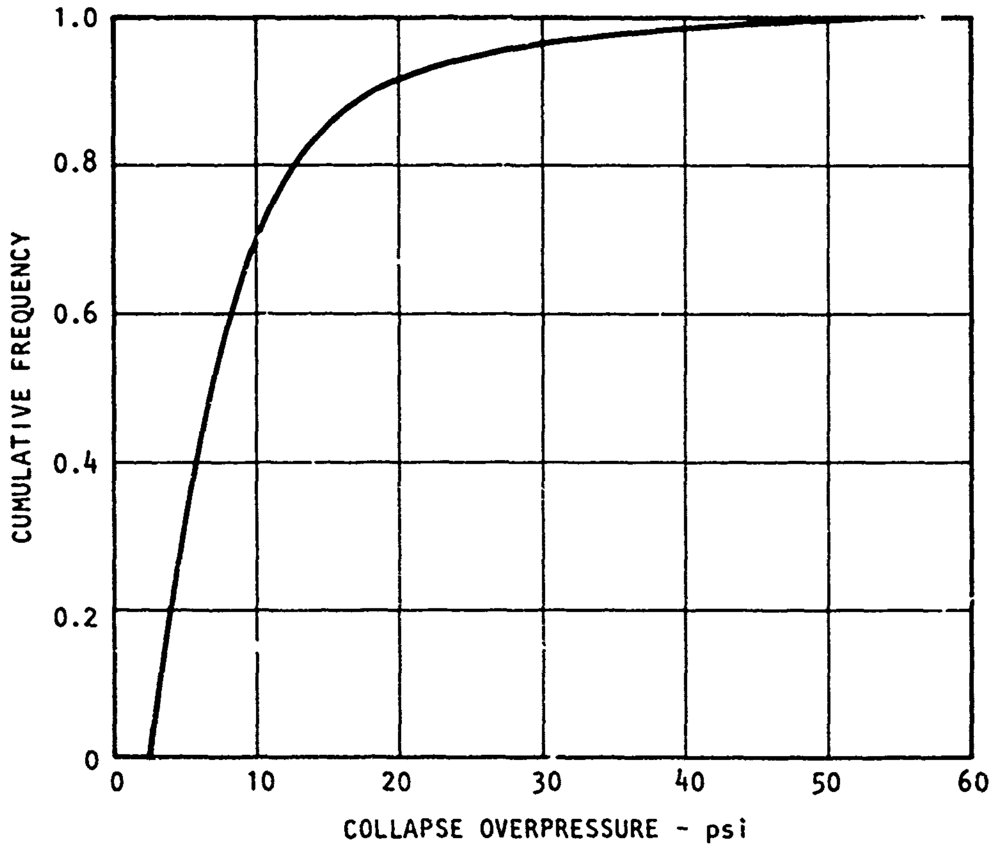


FIGURE 2 COMPARISON OF THE CUMULATIVE FREQUENCY DISTRIBUTIONS OF THE MEAN COLLAPSE OVERPRESSURE FOR EXTERIOR WALLS BY THE TYPE OF BUILDING FRAME



(a) Histogram for all floor cases



(b) Cumulative frequency distribution for all floor cases

FIGURE 3 HISTOGRAM AND CUMULATIVE FREQUENCY DISTRIBUTION OF THE MEAN COLLAPSE OVERPRESSURE FOR THE FLOORS OVER BASEMENT AREAS OF 36 BUILDINGS and therefore, that the wall loading transferred to the frame can be approximated by an impulse loading. However, for many of the actual buildings analyzed, the strength of the exterior walls under blast loading was sufficiently high to make it doubtful that the frame could survive at the overpressure level required to collapse the walls. For example, Figure 1 shows that 50 percent of the walls of the NSS buildings analyzed were predicted to collapse at an overpressure level greater than 6 psi. The strength of the exterior walls is important in calculating the collapse of the frame, since, for a given overpressure level, the blast loading on the total wall area can be much more severe than the blast loading on the frame alone plus an impulse loading from a frangible-type wall.

To investigate the relative strength of the exterior walls and frame of a building would require a comprehensive computer program that includes inelastic response under dynamic loading as well as realistic frame collapse mechanisms. Since such a program was not available, a computer program for analyzing the elastic and inelastic dynamic response of two-dimensional structural frames was used (Ref. 14). Although the program does not include frame collapse mechanisms, it was felt that the results would provide a basis for estimating the possible collapse strength of a building frame relative to the strength of the exterior walls.

The building selected for analysis was the North Carolina National Bank, Greensboro, North Carolina. The building has a structural steel frame, and consists of eight stories, with a height of about 110 ft, and plan dimensions of 50 ft by 115 ft. A complete description of the bank building and the results of the analysis of the exterior walls are given in Ref. 6. The exterior walls on the upper stories consist of a 4-in.-thick brick veneer, which is continuous over the frame members, and an 8-in.-thick terra cotta backing wythe, which is inset in the frame and parged to the brick veneer.

Three different types of frame analyses were performed: (1) an elastic analysis to determine the strength of the inset walls on the ends of the building acting as shear walls, (2) elastic frame analyses at various overpressure levels, and (3) inelastic frame analyses at various overpressure levels. The exterior walls of the building were previously found to have an incipient collapse overpressure (50 percent probability) of 15.7 psi (Ref. 6), and therefore the blast loading for the frame analyses are calculated for a box-type building with nonfailing exterior walls with window openings.

The results of the analyses provide an estimate of the collapse strength of the structural steel frame of the bank building under blast loading, even though the computer program used cannot predict frame collapse. The results of the first analysis, for the shear wall building, indicated that the cracking of the exterior walls acting as shear walls occurs at an incident overpressure level of less than 2 psi, since the moment ratios (computed moment/yield moment) for the shear walls are above 35 for a 2-psi incident overpressure level. Therefore, it was assumed for the other two types of analyses that the inset end walls acting as shear walls contributed negligible resistance to the frame, and that the analysis of the frame acting alone should adequately model the building behavior under lateral load.

An elastic analysis of the frames for 16 psi, which approximated the incipient collapse overpressure of the exterior walls, indicated a maximum stress ratio (computed stress/stress at yield) of about 20. Since the elastic analysis is much simpler than the inelastic analysis, the frames were then analyzed for elastic behavior at 5-, 4-, and 3-psi overpressures to obtain an estimate of the frame strength. The results of the elastic analyses indicated that the strength of the frames was in the range of the lower overpressures examined, and therefore the inelastic frame analyses were run at 3-, 4-, and 5-psi overpressure levels.

The inelastic analyses indicated maximum ductility ratios at 3 psi of 13.4 for the beams and 20.6 for the columns, and maximum moment ratios of about 1.64 for the beams and columns. At the 4-psi overpressure level, the maximum ductility ratios were 29.5 in the beams and 42.2 in the columns, and the maximum moment ratios were in excess of 2. A simplified hand calculation indicated that the P-Δ effect, which is not included in the computer program, would increase some of the moment ratios by over 50 percent. The calculated lateral deflection of the top story of the building was about 21 ft for the 3-psi overpressure level, and 47 ft for the 4-psi level. If it is assumed that the frame would collapse at a ductility ratio of about 50, then the estimated collapse overpressure is between 3-, and 4-psi incident overpressure level. The actual blast strength could be much less, since the effect of the axial column load (P-Δ effect) and frame collapse mechanisms, such as column buckling or instability, are not accounted for in the analytical procedure.

It should be noted that the frame of the North Carolina National Bank building appears to be constructed of relatively light structural shapes that may not necessarily be typical of most NSS structures. In any event, however, the analysis indicated that the blast resistance of the frame of the building was much less than (possibly only one-fourth) that of the exterior walls. This, of course, is an important consideration in predicting either building damage or casualties.

Building System Computer Program

Since the inception of the evaluation project, the intention has been to develop a procedure for the analysis of a building system that would be applicable to various requirements of DCPA. For predicting damage to NSS structures in this program, it was assumed that each wall analyzed could be treated as though it were the "front face" of the building with an ideal blast wave advancing at normal incidence to it.

The time-sequence of collapse of various building elements, or the effect of the engulfment of the building by the blast wave, is not directly accounted for in the current computer programs. For example, to use the computer codes for predicting the collapse of all exterior walls of a building (i.e., front, side, and back) for the blast approaching from one direction it is necessary to use engineering judgment in providing realistic input data. Such a procedure was used to correlate analytical predictions with nuclear field tests of brick load-bearing-wall houses (Ref. 4).

In order to systematize the building evaluation procedure, a flow diagram was prepared during the current effort that outlines the computer analysis of all wall elements on one story of a building. The results are presented in Section III.

III BUILDING SYSTEM PROGRAM

The purpose of this phase of the research was to examine a method for systematizing the collapse predictions for blast loaded buildings. Since the inception of the existing structures evaluation project, the intention has been to develop a procedure for the analysis of a building system that would be applicable to various requirements of DCPA, such as damage assessment, survival and injury predictions, and debris predictions. Because of the complex nature of analyzing the response and collapse of buildings under dynamic loading, as well as the difficulty of calculating precise blast loadings on each element in a complex building geometry, the approach has been to establish a sound technical basis for the analysis of each building element. It has been necessary to derive realistic mathematical response models before computer codes could be prepared for the various structural elements of interest. Although the original intent was to develop subroutines for each element for the eventual incorporation into a single computer program, the need to analyze existing buildings preceded the completion of a building system program. Instead, relatively complete computer programs, as opposed to building element subroutines, were prepared for each building element; i.e., for each element the computer code consists of a main routine, a subroutine to calculate the resistance function, a subroutine to calculate transformation factors, subroutines to calculate the exterior and interior blast pressures and net load on the element, and a subroutine for calculating the probability of collapse. The development of these individual element programs diverted some effort from the development of a building system program; however, the individual programs permitted analyses of existing buildings to be performed much sooner than would otherwise have been possible. Also, the availability of individual element programs was convenient for the correlation of experimental data with analytical models.

Essentially, the building element computer programs were developed as research tools for use in developing realistic analytical prediction models, and for performing limited analyses of buildings rather than for performing a large number of analyses of existing buildings. However, as originally intended, it has become apparent that a computer program for analyzing a building system, or at least a building subsystem, would be useful. Therefore, during this phase of the research effort, the feasibility of incorporating the previously developed computer programs for wall analysis into a program for the analysis of a subsystem of the overall building system was examined. Specifically, a relatively detailed flow diagram was prepared that outlines the procedure for analyzing all exterior and interior walls on one story of a building. During this phase, the computer flow chart was prepared, but the computer program was not written.

A subsystem analysis approach was chosen as the most expedient and logical next step in the development of an overall building evaluation procedure. Figure 4 shows a macroscopic organizational flow chart of the proposed program to be used. The subsystem is one floor level of a building that can be oriented at any angle to the blast wave front. For a given free-field overpressure level, the net loading on each wall element will be computed and the wall response calculated on a room-by-room basis as the blast wave moves through the building.

At the present time the evaluation procedure has the capability of calculating the exterior pressure-time environment resulting from an interacting blast wave, and can compute the resulting interior pressure

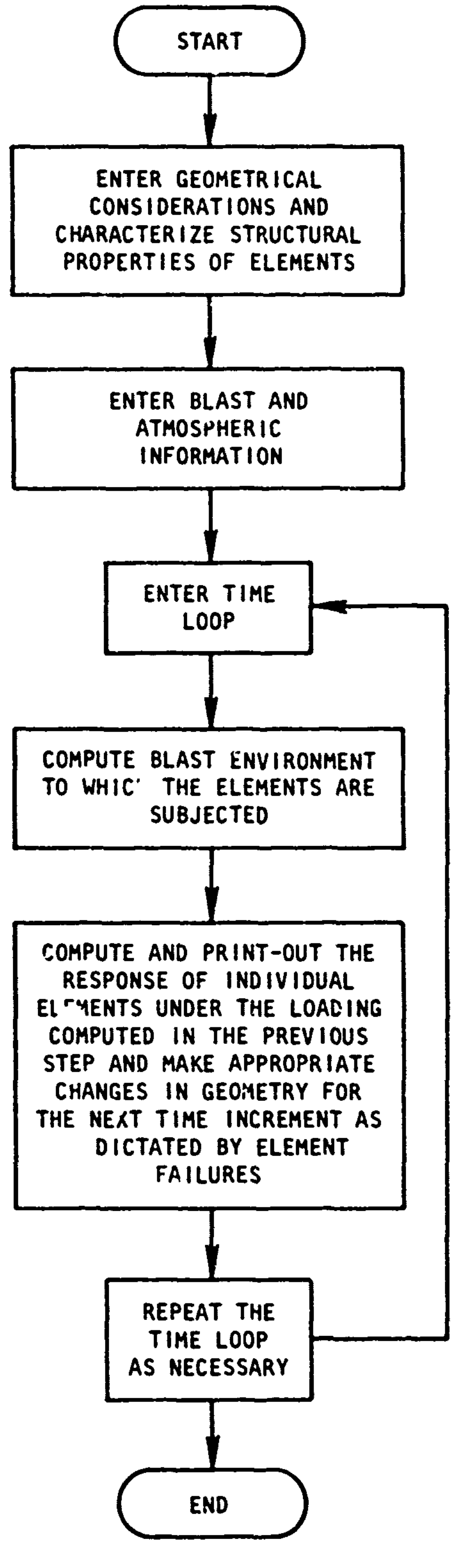


FIGURE 4 BUILDING SUBSYSTEM PROGRAM MACROSCOPIC ORGANIZATIONAL FLOW CHART

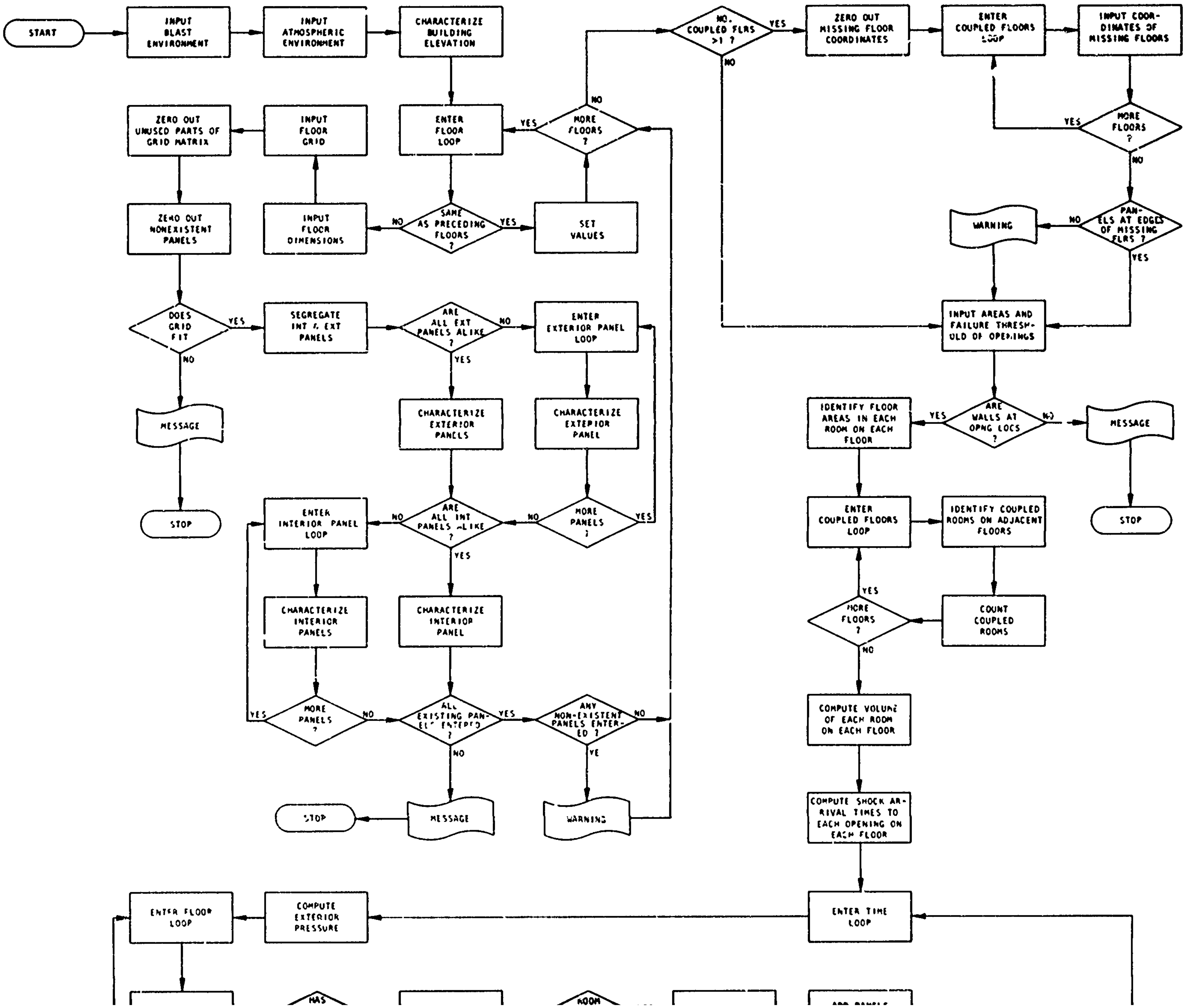
build-up to a single room. For the subsystem program, it will be necessary only to extend the capability to calculate the interior pressure in a multiroom complex by the method presented in Ref. 12. Also, as discussed previously, the mathematical models for predicting the response and collapse of walls are available. The remaining task consists of combining the loading and response models into a single subsystem program that includes the geometry of the floor of a building.

Figure 5 shows a detailed organizational flow chart of the proposed subsystem program. Table 1 is a list of abbreviations used in the flow chart.

Table I

ABBREVIATIONS USED
IN FLOW CHART

Abbreviation	Word Represented
EXT	EXTERIOR
INT	INTERIOR
OPNG	OPENING
FLR	FLOOR
PR	PAIR
ARVD	ARRIVED
RM	ROOM
NRST	NEAREST


START
INPUT BLAST ENVIRONMENT
INPUT ATMOSPHERIC ENVIRONMENT
CHARACTERIZE BUILDING ELEVATION
ENTER FLOOR LOOP
SAME AS PRECEDING FLOORS ?
YES
NO
SET VALUES
MORE FLOORS ?
INPUT FLOOR DIMENSIONS
INPUT FLOOR GRID
ZERO OUT UNUSED PARTS OF GRID MATRIX
ZERO OUT NONEXISTENT PANELS
DOES GRID FIT
MESSAGE
STOP
SEGREGATE INT & EXT PANELS
ARE ALL EXT PANELS ALIKE ?
ENTER EXTERIOR PANEL LOOP
CHARACTERIZE EXTERIOR PANELS
CHARACTERIZE EXTERIOR PANEL
MORE PANELS ?
ARE ALL INT PANELS ALIKE ?
ENTER INTERIOR PANEL LOOP
CHARACTERIZE INTERIOR PANELS
CHARACTERIZE INTERIOR PANEL
ALL EXISTING PANELS ENTERED ?
ANY NON-EXISTENT PANELS ENTERED ?
WARNING
NO. COUPLED FLRS >1 ?
ZERO OUT MISSING FLOOR COORDINATES
ENTER COUPLED FLOORS LOOP
INPUT COORDINATES OF MISSING FLOORS
PANELS AT EDGES OF MISSING FLRS ?
INPUT AREAS AND FAILURE THRESHOLD OF OPENINGS
ARE WALLS AT OPNG LOCS ?
IDENTIFY FLOOR AREAS IN EACH ROOM ON EACH FLOOR
IDENTIFY COUPLED ROOMS ON ADJACENT FLOORS
COUNT COUPLED ROOMS
COMPUTE VOLUME OF EACH ROOM ON EACH FLOOR
COMPUTE SHOCK ARRIVAL TIMES TO EACH OPENING ON EACH FLOOR
ENTER TIME LOOP
COMPUTE EXTERIOR PRESSURE
ENTER FLOOR LOOP
HAS
ROOM

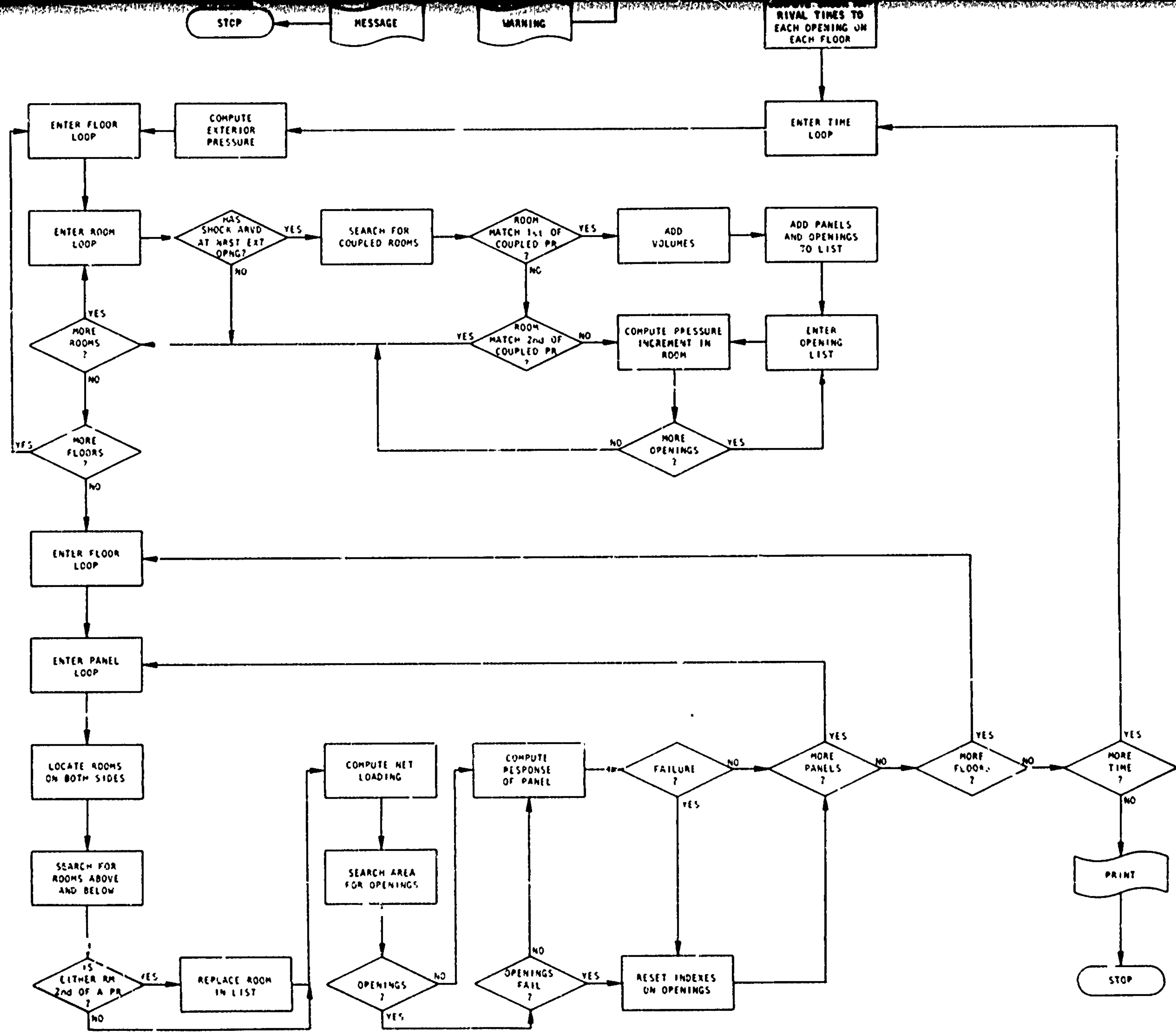


FIGURE 5 BUILDING SUBSYSTEM ORGANIZATIONAL FLOW CHART

IV ANALYSIS OF BASEMENT WALLS

The objective of this phase of the effort was to examine the blast resistance of exposed, reinforced concrete basement walls with door openings to determine the feasibility of retrofitting EOCs with blast doors. The primary purpose was to determine if reinforced concrete basement walls located in areaways of existing buildings could resist 10-psi blast overpressure.

Approach

In past studies, the collapse strength of blast loaded walls of existing buildings has been determined for various types and configurations of walls. Although mathematical models have been developed for walls with window openings, the procedures were not extended to include walls with door openings. Since there was insufficient time and funds to develop a generalized model and computer program for calculating the resistance and collapse of the wall configuration of interest, the following three-phase approach was used:

(1) A detailed yield-line analysis of several specific reinforced concrete walls with door openings was made to establish the static resistance over a limited range of wall widths.

(2) The computer programs developed for the building evaluation procedure for DCPA for analyzing the collapse of wall elements was used to generate resistance functions for walls without door openings. The results were then compared with those obtained from the yield-line analysis for walls with door openings to determine the feasibility of using the existing computer programs to simulate the dynamic response and

collapse of walls with door openings. If the resistances for the two wall types was found to be not comparable, then it would be necessary to hand calculate a resistance function for each wall case.

(3) An existing finite element computer program was used to analyze the static behavior for a few wall cases with door openings to determine if shear or stress concentrations could conceivably produce a wall failure not predictable by the other analyses.

Wall Design

The basement wall considered in this study was located in an open areaway such that the wall and door are fully exposed to the air blast effects. For simplicity, a standard door opening of 3 ft 8 in. wide by 6 ft 8 in. high was adopted; this is a two-unit-of-exit-width door opening as specified in Ref. 15. It was also assumed that the door was closed for all analyses and that it did not fail. The general layout of the basement wall analyzed is shown in Figure 6. As noted in Figure 6, it was assumed that the wall was bounded at the top and bottom by the first story and basement floors, and on the sides by the vertical areaway walls. The basement wall was continuous at the areaway wall intersection, and no interior walls abutted the basement wall in the vicinity of the areaway. The soil backfill adjacent to the areaway and basement walls extended to the first story level.

Since specific wall details were not provided, it was assumed that the basement walls were designed according to the 1963 ACI code (Ref. 16). Pertinent requirements of the code applicable to basement walls are:

- Area of horizontal reinforcing steel is not less than 0.0025 times the area of the reinforced section of the wall.
- Area of vertical reinforcing steel is not less than 0.0015 times the area of the reinforced section of the wall.

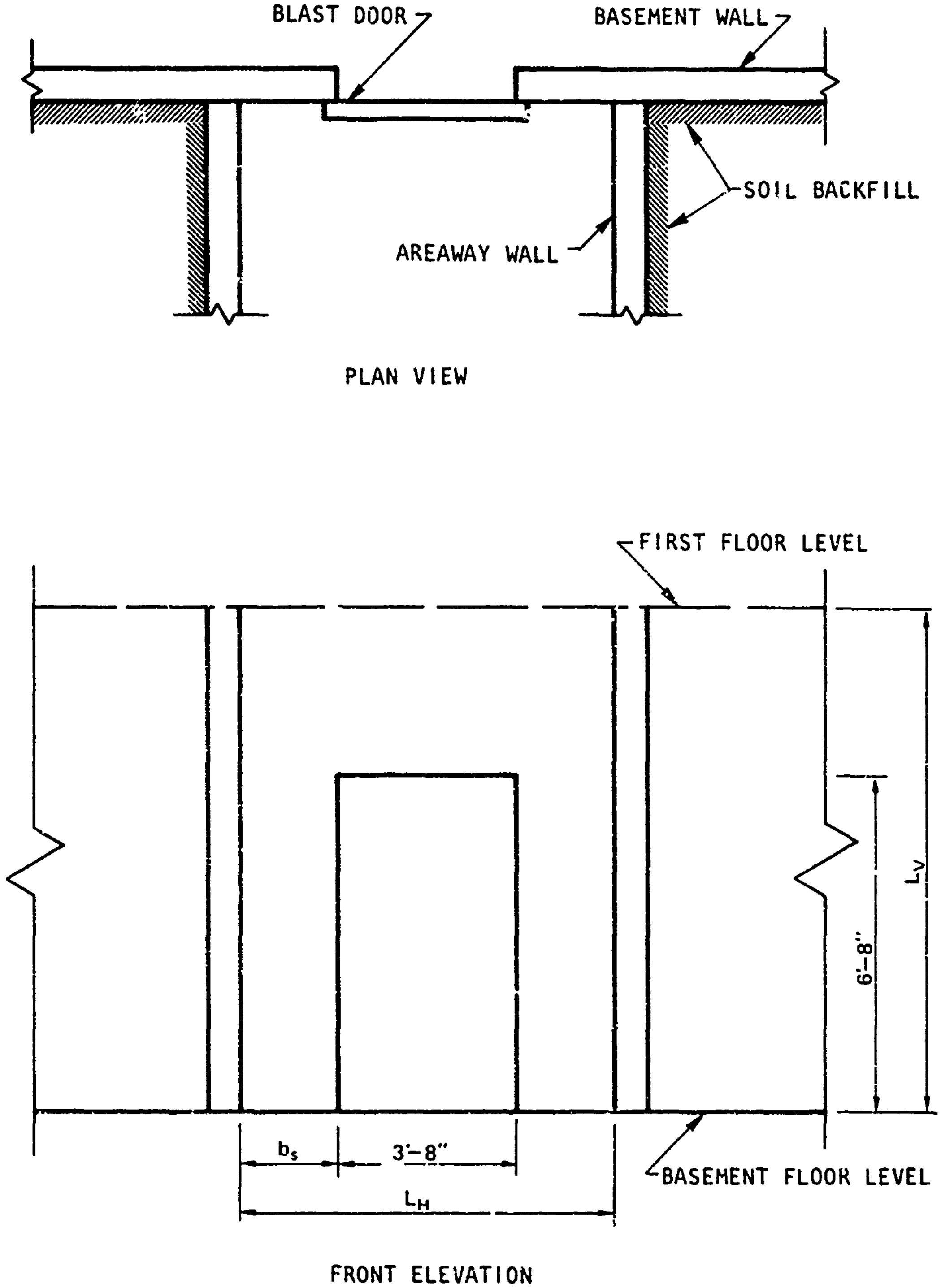


FIGURE 6 PLAN VIEW AND FRONT ELEVATION OF BASEMENT WALL USED IN THE ANALYSIS

- In addition to the above, two No. 5 bars are required around the door opening, and extending a distance of 24 in. beyond the opening.
- Minimum bar size is No. 3 at 18 in., center-to-center.
- Basement wall is assumed to be anchored to the floors and areaway walls with reinforcement equal to that in the wall.
- Minimum basement wall thickness is 8 in.

In addition, since the most efficient use of the reinforcing steel for the basement walls with soil backfill would dictate that the reinforcement be placed near the inside face of the wall, it was assumed that the reinforcement for the basement wall located in the areaway was also near the inside face.

Since it was assumed that the basement wall in the areaway was identical to that with soil backfill, the strength of a basement wall with minimum code reinforcement and with soil backfill was checked for adequacy as follows:

The lateral static soil pressure against the wall is

$p_h = K_o h \rho$,

where K_o = lateral soil coefficient (assumed as 0.30 for well-drained soil)

h = soil depth

ρ = unit weight of soil.

For height of wall, $L_v = 10.0$ ft,

$p_h = (.3)(10)(100) = 300$ psf (bottom of wall).

The maximum applied moment for a one-way wall simply supported at the top and bottom and with a triangular load function is

$M = 0.1283\ P_T L_v$,

where P_T = total applied load.

Therefore, the applied moment is equal to

$$M = (.1283)(300 \times \frac{10}{2})(10) = 1925 \text{ ft-lb/ft} .$$

For a reinforced concrete section with tensile reinforcement only, the ultimate bending moment of the section is

$$M_u = \varphi \left[A_s f_y (d - \frac{a}{2}) \right]$$

where $a = \dfrac{A_s f_y}{0.85 f'_c b}$.

For a reinforced concrete basement wall with minimum thickness, $t_w = 8$ in., the area of vertical reinforcement is

$$A_s = (8)(12)(.0015) = 0.144 \text{ sq in./ft of wall,}$$

$$A_s = \text{No. 3 @ 9 in.,}$$

and

$$d = 8 - (\frac{3}{4} + \frac{1}{2} \times \frac{3}{8}) = 7.06 \text{ in.}$$

For $f'_c = 3000$ psi and $f_y = 33{,}000$ psi

$$a = \frac{(.144)(33{,}000)}{(.85)(3000)(12)} = 0.1553,$$

and with a coefficient of flexure $\varphi = 0.90$, the ultimate bending moment is therefore

$$M_u = (.90) \left[(.144)(33{,}000)(7.06 - \frac{.1553}{2}) \right] = 29{,}862 \text{ in.-lb/ft}$$

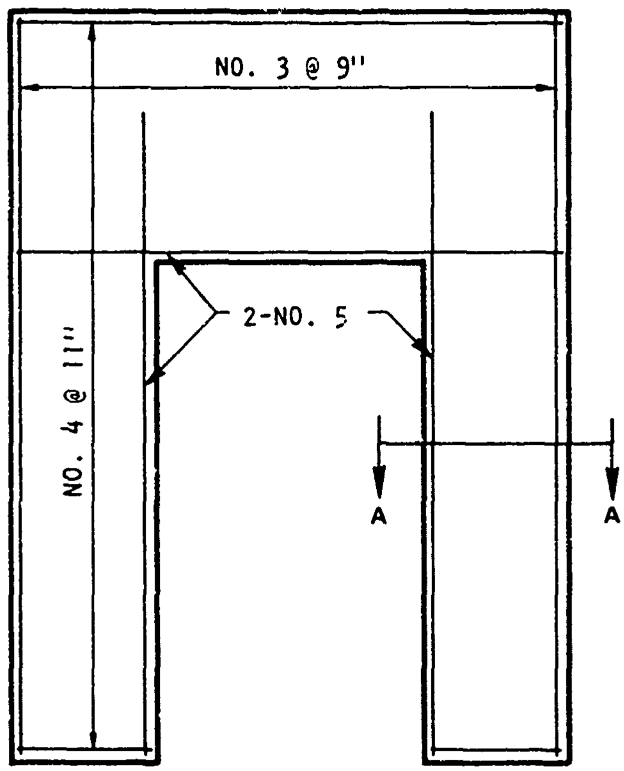
or

$$M_u = 2489 \text{ ft-lb/ft} .$$

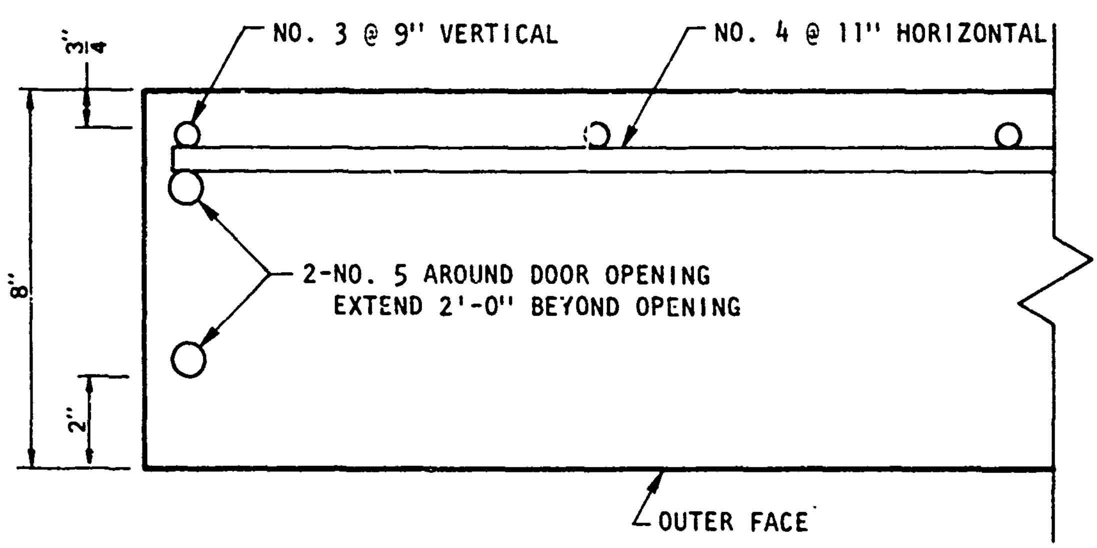
Since $M_u > M$, a fully buried, 10-ft-high by 8-in.-thick reinforced concrete basement wall with minimum code reinforcement is adequate to resist the pressure from a well-drained soil. Figure 7 illustrates the reinforcing steel details assumed for the 8-in.-thick basement wall used in the analysis.

Analysis

Various types of analyses were performed because a mathematical model that adequately represented the dynamic behavior and collapse of



ELEVATION OF BASEMENT WALL



SECTION A-A

FIGURE 7 REINFORCING STEEL DETAILS FOR 8-INCH THICK CONCRETE BASEMENT WALL USED IN ANALYSIS basement walls with door openings had not been developed previously. The object was to use the available analytical tools to estimate with a good degree of confidence the collapse strength of basement walls in areaways without actually expending the time and effort required to develop a realistic mathematical model and writing the computer code.

Wall With Door Opening

The work-energy method from the yield-line theory for reinforced concrete slabs was used to calculate the flexural resistance for reinforced concrete walls with door openings. The method is outlined in Ref. 4, and will not be repeated here. The purpose of performing a limited number of yield-line analyses was to compare the resistance of walls with door openings with that of walls without door openings. If the resistance values for the two wall types were found to be approximately the same, then the available dynamic computer programs for wall elements could be used to provide interim collapse predictions for a variety of wall cases. However, if the resistances for the two wall types were found to be different, then the resistance values calculated for the walls with door openings could be used to perform a limited number of dynamic analyses.

The reinforcing steel details used for the analysis are shown in Figure 7. Since the calculation of the yield-line moments is a relatively tedious hand calculation requiring trial and error solutions, a minimum number of wall cases was considered. Therefore, only an 8-in.-thick wall with a height of 120 in. was treated; walls were analyzed with widths, L_y, of 92 in., 116 in., and 140 in. The calculated yield lines and resistance values for the three walls are shown in Figure 8.

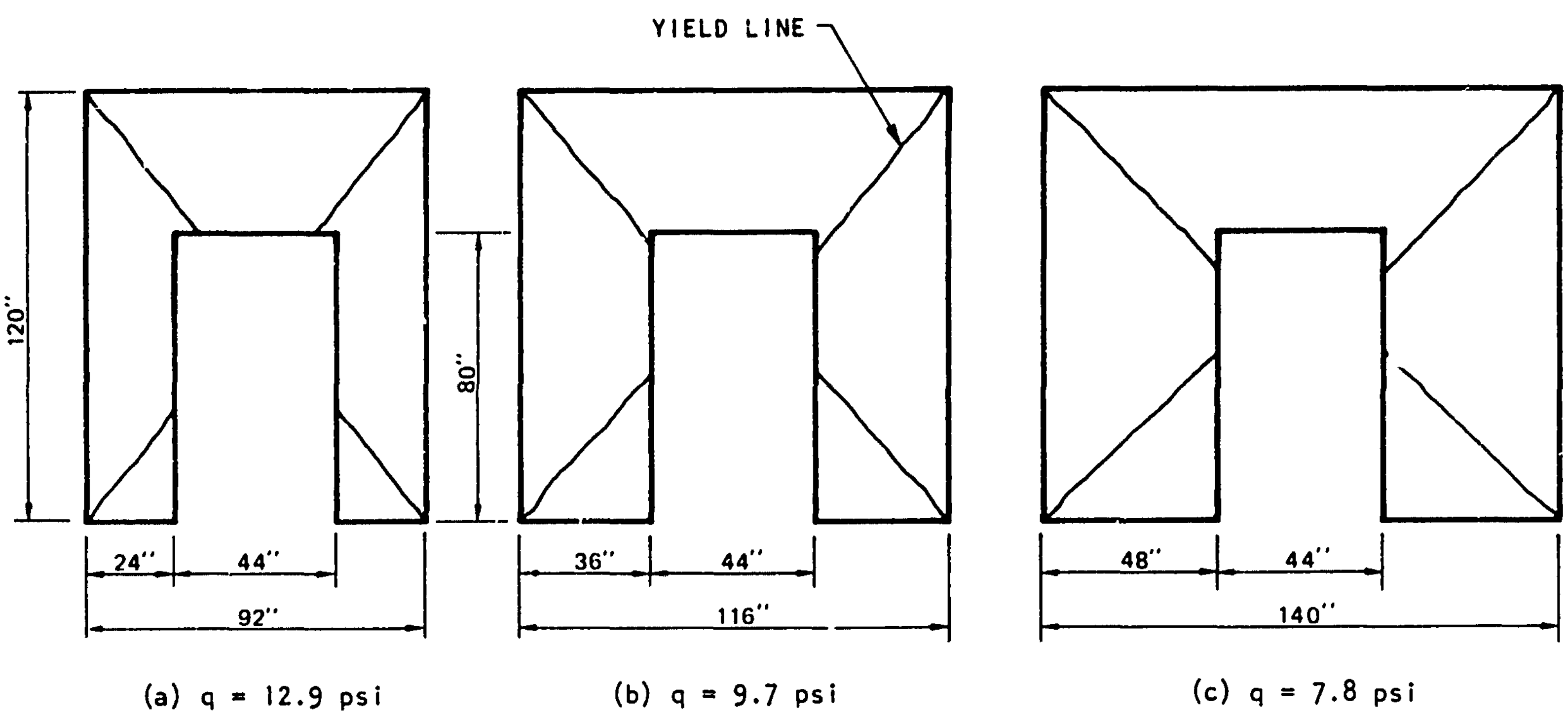


FIGURE 8 YIELD LINES AND RESISTANCE VALUES FOR REINFORCED CONCRETE WALLS WITH DOOR OPENING

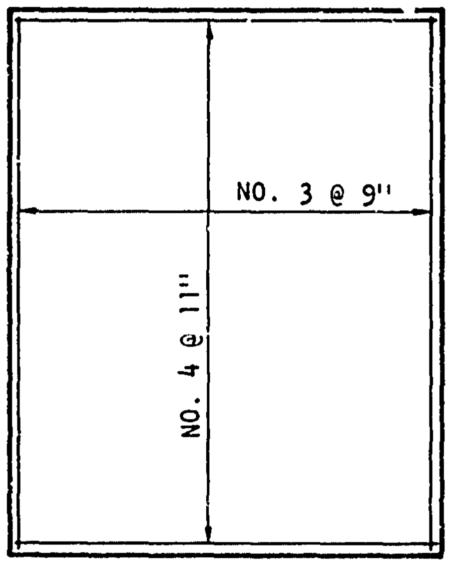
Wall Without Door Opening

A series of computer runs was made using the SRI programs developed previously for two-way action reinforced concrete walls (Ref. 4). The vertical load, P_v, in the plane of the wall was assumed to be zero; i.e., the wall was considered as a panel wall that did not carry any loads from the floor levels above. The horizontal and vertical reinforcement for the walls without door openings was the same as that for walls with door openings, except that for walls without door openings the two No. 5 bars around the opening shown in Figure 7 were deleted, as shown in Figure 9.

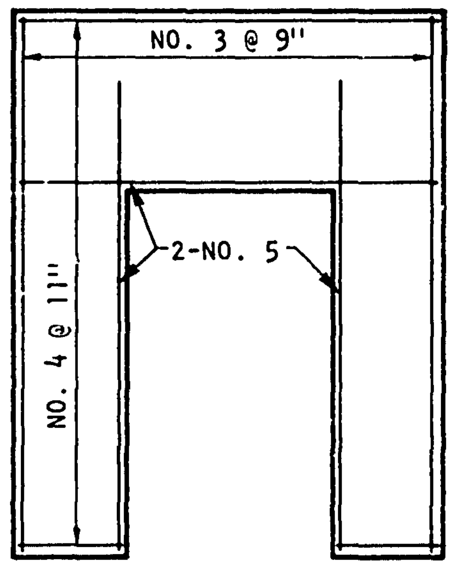
The values of the resistance for walls without door openings are plotted in Figure 10, where they are compared with the resistances calculated by the yield-line theory for walls with door openings. From the figure, it is apparent that the maximum resistance, q, for the two wall cases is approximately equal for the range of wall widths considered. Based on this limited study, it was assumed for the purpose of performing preliminary dynamic collapse predictions for reinforced concrete basement walls that the flexural resistances of wall with and without a door opening, and reinforced as shown in Figure 9, were equal.

To estimate the collapse of blast loaded basement walls with door openings, dynamic analyses were performed for a range of wall widths, two wall thicknesses, and two wall heights. The results of these analyses are shown in Figure 11. It should be noted that, since it was assumed that the basement walls were panel walls with $P_v = 0$, and did not arch, the curves can be considered as lower bound predictions for each wall type shown.

The collapse criterion adopted for walls in the evaluation procedure was based on the collapse of the wall in flexure; as discussed in Ref. 4, collapse is predicted as a result of excessive steel strain, instability, or an excessive ductility ratio. From studies made during the development



(a) Solid Wall Without Door Opening



(b) Wall With 3'-8" x 6'-8" Door Opening

FIGURE 9 WALLS ASSUMED AS EQUIVALENT FOR ANALYSIS PURPOSES

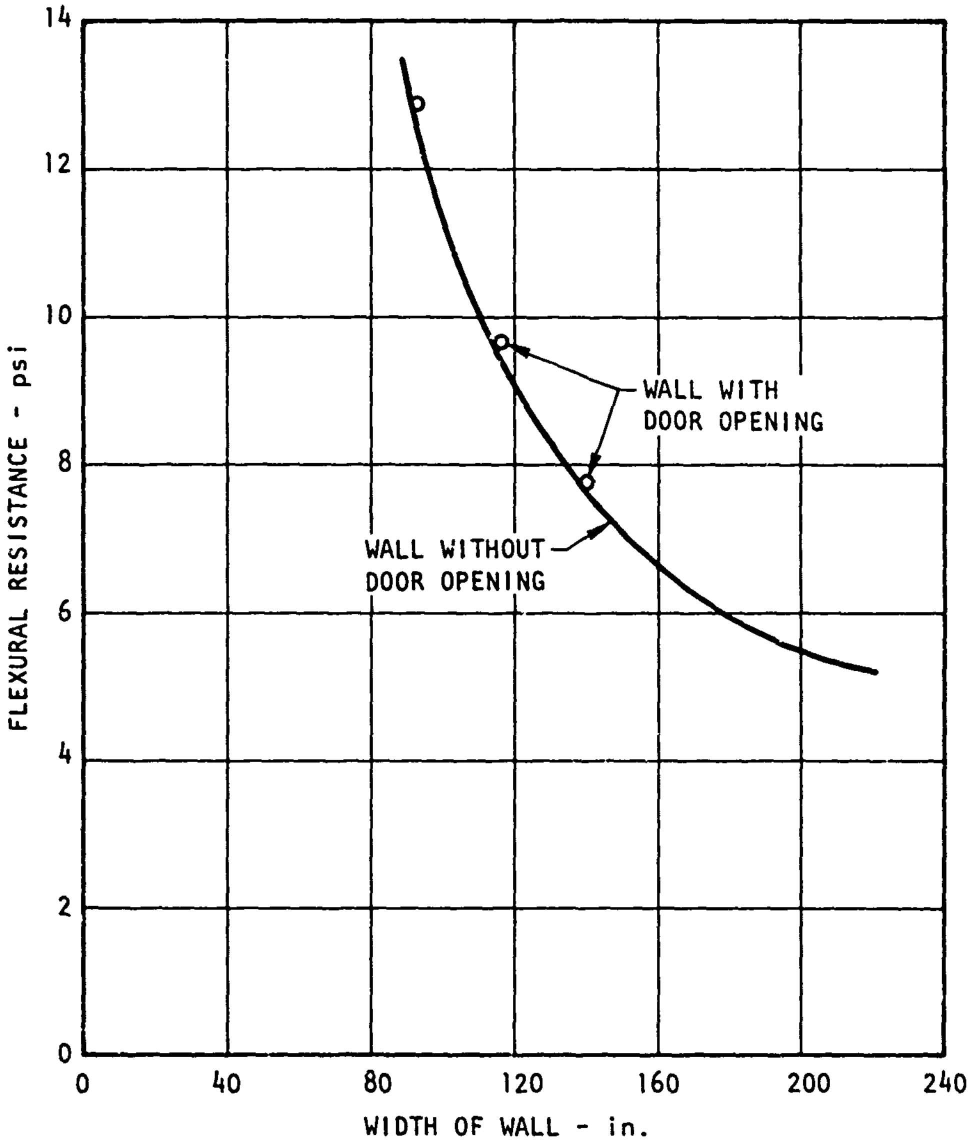


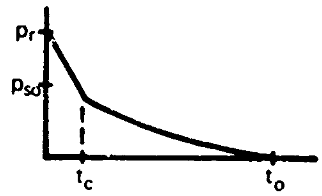
FIGURE 10 COMPARISON OF RESISTANCE VALUES FOR WALLS WITH AND WITHOUT A DOOR OPENING

WALL PARAMETERS

L_V = 96 in. and 120 in.
L_H = 92 in. to 240 in.
t_w = 8 in. and 12 in.
f'_{dc} = 3,750 psi
f_{dy} = 44,000 psi
p = 0.0025 A_c (horizontal)
p = 0.0015 A_c (vertical)
γ = 145 pcf
A_w = 0
Door opening: 3'-8" x 6'-8"
Support case: fixed four edges

LOAD PARAMETERS

W = 1 Mt
P_v = 0
S = 6.7 ft



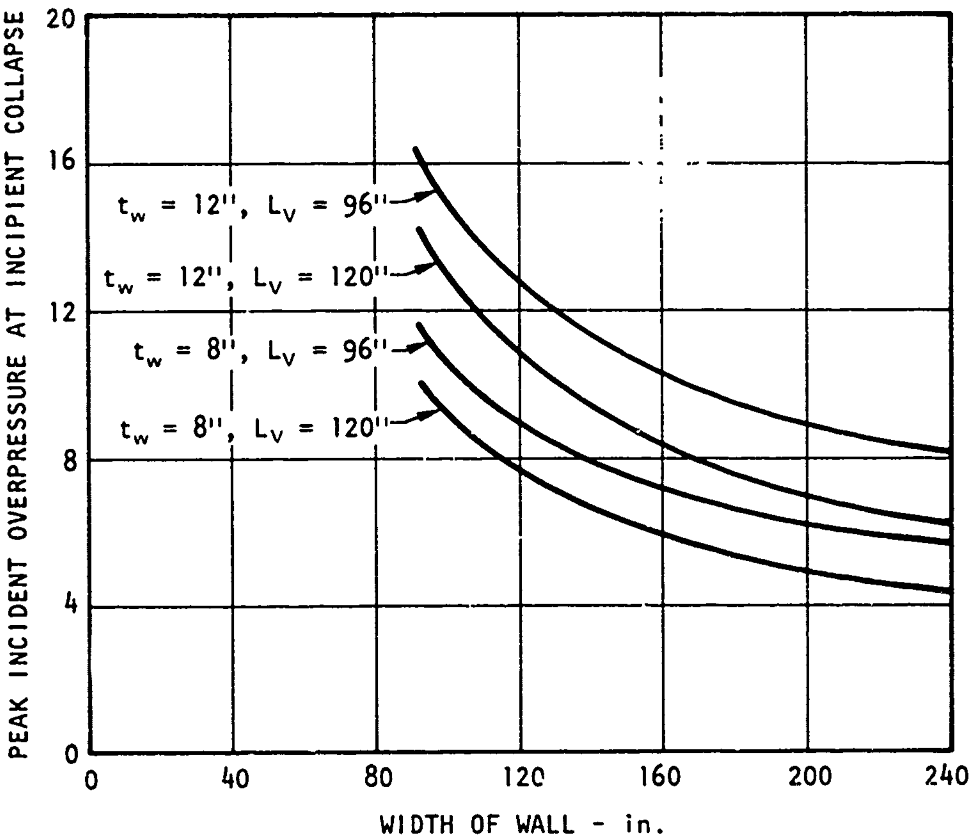


FIGURE 11 PEAK INCIDENT OVERPRESSURE AT INCIPIENT COLLAPSE VERSUS WALL WIDTH

Two-Way Reinforced Concrete Basement Wall In Areaway of the original wall evaluation procedure, it was determined that for lightly reinforced members with the usual type of supports, a shear failure would not be expected to occur, and if it did, it would not be expected to result in collapse of the wall. However, for the basement walls considered in this study, there were two factors that could influence the collapse mechanism assumed in the original procedure. First, the door opening could produce both higher local shears and stress concentrations than found in solid walls. Second, the reaction of the basement wall at the construction joint between the basement wall and the areaway support wall is opposite in direction to that usually encountered. That is, the reaction places the joint between the basement and areaway walls in tension; the areaway wall provides lateral support to the basement wall only through development of tensile forces in the reinforcing steel continuous through the joint shown in Figure 12a. This, of course, is opposite to the usual case where the lateral load on the wall forces the member to bear directly against its support as shown in Figure 12b.

There are two important implications as a result of the type of lateral support provided by the areaway walls that could influence the collapse predictions of the basement walls shown in Figure 11. First, the reinforcing steel between the basement and areaway walls could fail in tension, which would result in one-way wall action between floor levels rather than two-way action as assumed in the analysis. Second, under the lateral blast load tne basement wall cracks along all supports at small elastic deflections as a result of the negative moment developed. As illustrated in Figure 13a, the reinforcing steel is near the inside surface of the basement wall and the effective depth of the steel for resisting this negative moment is measured from the inside wall surface; for the assumed wall this distance, d, would be only 1-3/8 in. Because of the cracks at the support, the thickness of the concrete available for resisting the shear force is only equal to d, and therefore a shear

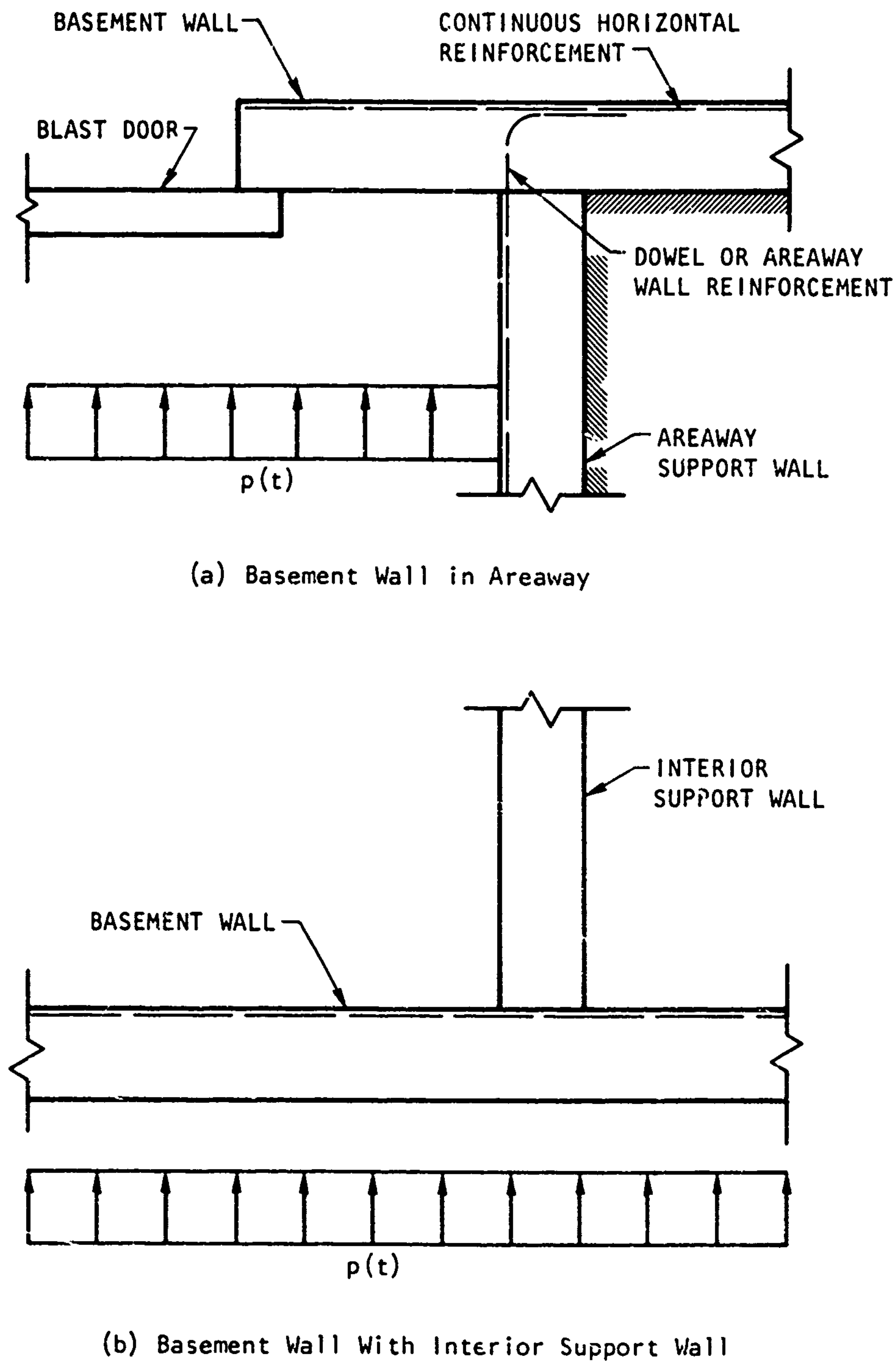
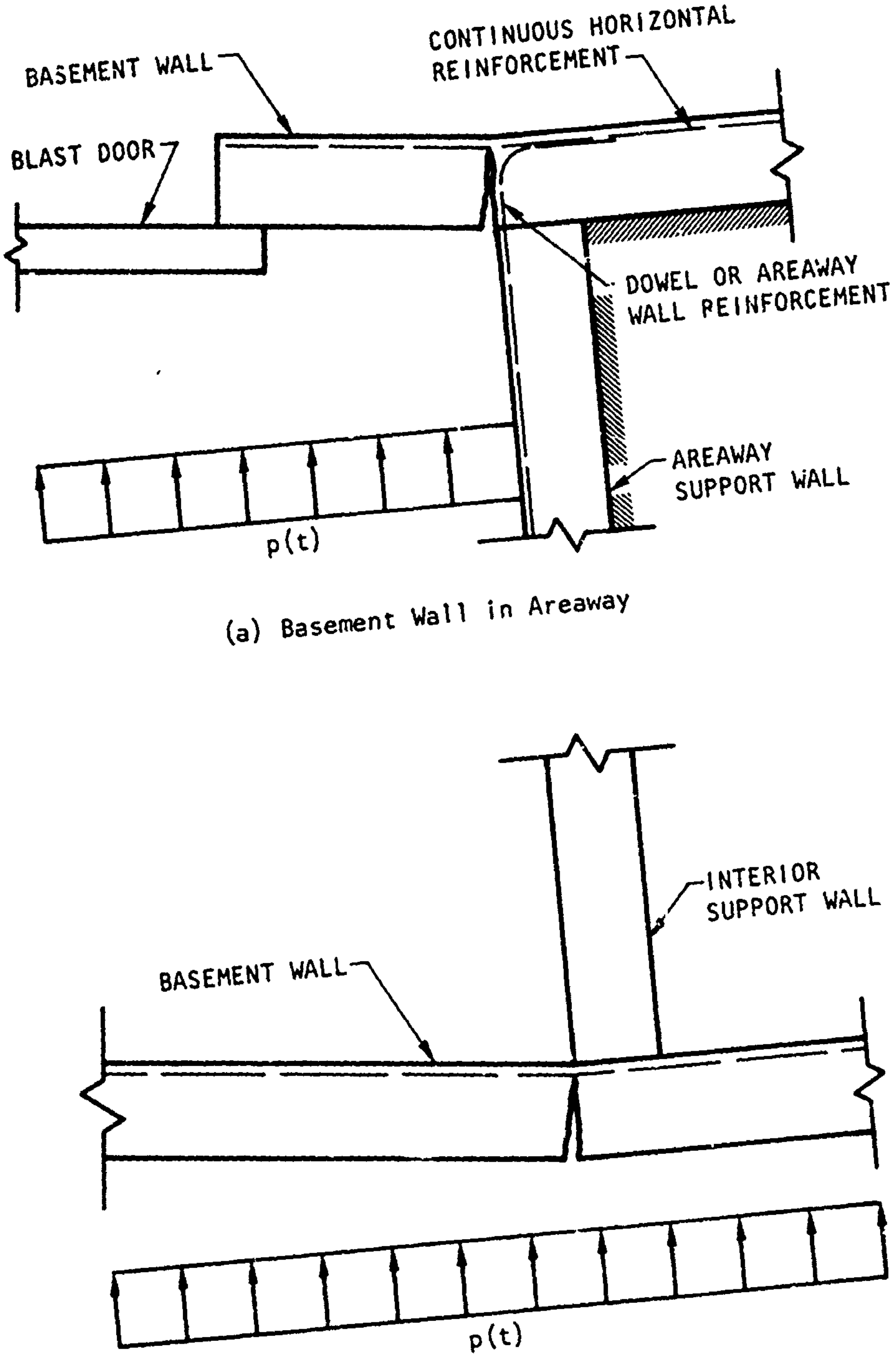


FIGURE 12 PLAN VIEW OF CONSTRUCTION JOINT AT INTERSECTION OF BASEMENT AND SUPPORT WALLS



(a) Basement Wall in Areaway

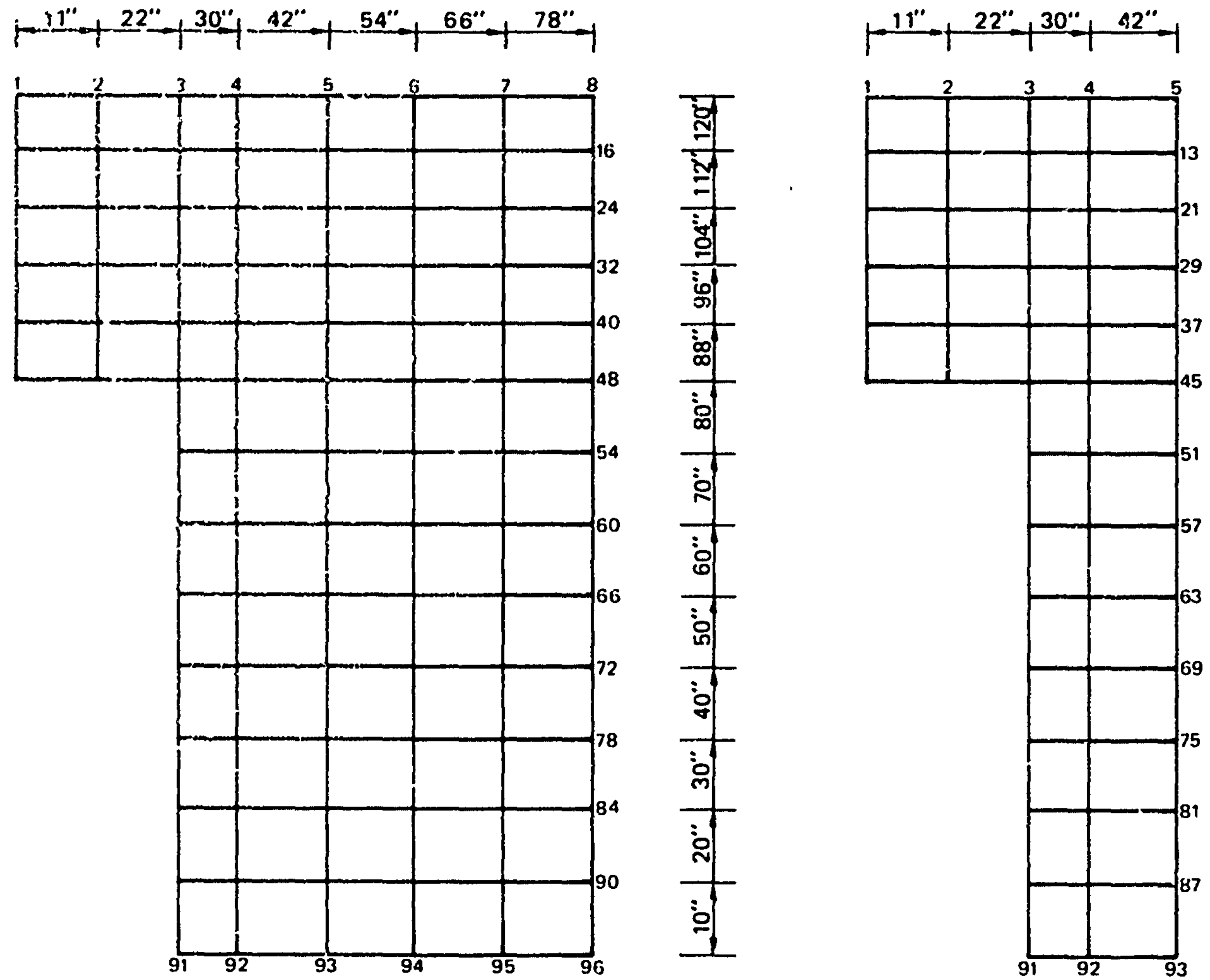
(b) Basement Wall With Interior Support Wall

FIGURE 13 PLAN VIEW SHOWING CRACK AT CONSTRUCTION JOINT AT THE INTERSECTION OF BASEMENT AND SUPPORT WALLS failure could occur early in the wall response. For the type of walls considered when developing the evaluation procedure, the reaction was assumed to act against the lateral interior support wall and a shear failure was not considered as a wall collapse mechanism. That is, a shear failure at the lateral support wall would precipitate a tensile membrane mode of response in the basement wall, rather than a wall collapse, and the reactive forces would be transferred to the support by tensile action of the continuous reinforcement; this is illustrated in Figure 13b. For a basement wall located in an areaway, a shear failure in the wall at the construction joint between the basement and areaway walls would result in rupturing of the concrete and tearing out of the continuous reinforcing steel in the basement wall; the small concrete cover over the reinforcement shown in Figure 13a could not be expected to resist the reactive forces of the basement wall.

Since the mathematical models developed for the evaluation procedures could not be used to investigate the details of localized internal stresses and reactions for the complex door opening wall geometry, an available static finite element computer program was used to estimate probable failure modes.

Finite Element Wall Program

Although the available finite element program is a powerful analytical tool, it is limited to static, elastic structural systems. Therefore, the primary value of the results for this study was to provide a basis for estimating possible collapse mechanisms; the results were of only limited quantitative value. The two basic wall configurations analyzed are illustrated in Figure 14; because of symmetry, it was only necessary to consider one-half the wall. The wall model consisted of an assemblage of plate elements, and only the support nodes are numbered on the figure. The useful output information included the deflections, internal loads,



(a) One-Half Wall, L_H = 156 in.

(b) One-Half Wall, L_H = 84 in.

FIGURE 14 WALL WITH DOOR OPENING SHOWING FINITE PLATE ELEMENTS AND NODES and extreme fiber stresses in the plates, and the reactions at each support node. For this exercise, only an 8-in.-thick wall with a 120-in. height was considered. The walls were analyzed for a static lateral load of 10 psi applied uniformly to the wall and door; the door load was distributed to the wall nodes adjacent to the opening.

The primary reason for conducting the finite element analysis was to provide more detailed information about the reactions at the construction joint formed by the intersection of the basement and areaway walls than was available from the other analyses. For the wall with L_u = 156 in., the following values for the basement wall reactions at the basement/areaway wall joint were obtained for the nodes shown in Figure 11:

Node	Reaction, lb
16	1,904
24	2,915
32	3,801
40	-3,165
48	11,785
54	6,125
60	6,079
66	6,027
72	5,684
78	5,081
84	1,194
90	2,680

The maximum reaction predicted is 11,785 lb at node 48; next is 6,125 lb at node 54. Since the actual wall had continuous support rather than point support at each node, it is appropriate to average the reaction between two adjacent nodes. Therefore, the maximum average applied shear along the basement/areaway wall joint is

$$V = \frac{11,785 + 6,125}{(2)(10)} = 896 \text{ lb/in.}$$

To determine whether this magnitude of applied shear would result in a shear failure in the basement wall, it was necessary to calculate the shear resistance of the wall. From Ref. 7, the unit shear resistance at the support of a reinforced concrete member is

$$(v_c)_s = \frac{2.28\sqrt{f'_c}}{1-2d/L} + \frac{3000p}{1-d/L} \le \frac{3.5\sqrt{f'_c}}{1-2d/L} \quad .$$

For the basement wall, the horizontal reinforcement is

$$A_s = 0.240 \text{ sq in./ft},$$

and d = 1.38 in. for a racked concrete section, therefore

$$p = \frac{0.240}{(12)(1.38)} = 0.01449.$$

Substituting these quantities into the above equation, the unit shear is

$$(v_c)_s = \frac{2.28\sqrt{3000}}{1-\frac{(2)(1.38)}{1.56}} + \frac{(3000)(.01449)}{1-\frac{1.38}{156}} = 171 \text{ psi}.$$

Since the total shear resistance at the support would be

$$(V_c)_s = (v_c)_s bd,$$

then

$$(V_c)_s = (171)(1)(1.38) = 236 \text{ lb/in.}$$

This value is, of course, much less than the shear force of 896 lb/in. resulting from a uniform static load of 10 psi. For estimating purposes only, if it is assumed that the dynamic shear resistance is 25 percent greater than the static, and that a dynamic load factor (DLF) of 1.15 is appropriate for the load type, then a rough estimate of the level of the blast load that would result in a shear failure in the wall would be

$$p_{3,?} \approx \frac{(236)(1.25)(10)}{(1.15)(896)} \approx 2.9 \text{ psi}.$$

For the basement wall with L_u = 84 in., two wall cases were considered; (1) simply supported on four edges, and (2) fixed on four edges. The values of the reactions along the basement/areaway wall joint for the simply supported wall with a 10-psi static load were as follows:

Node	Reaction, lb
13	1,520
21	2,318
29	3,087
37	-3,839
45	7,764
51	3,924
57	3,820
63	3,976
69	4,027
75	3,904
81	3,531
87	2,467

Using the same method as before for two adjacent nodes, the predicted maximum average shear along the basement/areaway wall joint is

$$V = \frac{7764 + 3924}{(2)(10)} = 584 \text{ lb/in.}$$

Again, a rough estimate of the blast overpressure that would result in a shear failure in the basement wall would be

$$p_{so} \approx \frac{(236)(1.25)(10)}{(1.15)(584)} \approx 4.4 \text{ psi.}$$

Based on the above rough estimates for 8-in.-thick reinforced concrete walls with door openings, it could be concluded that a shear failure will occur at blast overpressures less than 5 psi if the horizontal distance from the edge of the door opening to the areaway wall is greater than about 18 in., i.e., for $L_w \geq 80$ in. However, since this estimate is based on a cracked concrete section, it is of interest to examine the effect on the strength of the wall of the concrete cracking along the supports.

As the exposed horizontal distance between the edge of the door opening and the areaway support wall is increased, the probability of a shear failure occurring in the basement wall is also increased. For example, if the horizontal distance is equal to the wall thickness,

t_w = 8 in. (L_H = 60 in.), then the full thickness of the concrete wall section is effective in resisting the applied shear force because the modulus of rupture of the concrete has not been exceeded and the concrete section is uncracked. Since the shear resistance for this case is much greater than the applied shear, the wall would not be expected to experience a shear failure. As the width of the wall is increased to 84 in., the values of the reactions and moments along the basement/areaway joint for an 8-in.-thick concrete wall fixed on four edges and with a 10-psi static load are as follows:

Node	Reaction, lb	Moment in.-lb
13	-26	1,149
21	1,187	10,813
29	1,738	20,043
37	2,726	37,845
45	5,031	55,746
51	3,786	49,166
57	4,076	52,550
63	4,267	54,022
69	4,329	52,165
75	4,103	45,468
81	3,125	31,006
87	167	7,190

The maximum moment predicted for two adjacent nodes is 52,550 in.-lb for node 57; next is 54,022 in.-lb for node 63. Therefore, the maximum average moment along the basement/areaway wall joint is

$$M = \frac{52,550 + 54,022}{(2)(10)} = 5329 \text{ in.-lb/in.}$$

To estimate if the wall cracks under the applied moment, it is necessary to calculate the resisting moment for the uncracked wall section. For a linear relationship between stress and strain across the section of the wall, the maximum resisting moment is equal to

$$M_u = \frac{f_r b t_x^2}{6} .$$

For

$$f_r = 8\sqrt{f'_c} ,$$

$$M_u = \frac{(8)\sqrt{3750}(1)(8)^2}{6} = 5226 \text{ in.-lb/in.} ,$$

which is approximately equal to the applied moment. Therefore, an 8-in.-thick concrete wall with a horizontal distance between the edge of the door and the areaway support wall of about 20 in. (L_u = 84 in.) would be expected to crack and experience a shear failure along the basement/areaway wall joint at a blast overpressure level somewhat less than 10 psi. This, of course, indicates a much greater blast strength for the uncracked wall case than was estimated above for the cracked wall case.

One-Way Reinforced Concrete Wall (Without Arching)

As discussed in the previous subsection, under dynamic load the initial shear failure in an 8-in.-thick, two-way reinforced concrete basement wall located in an areaway would occur at the points of maximum shear along the joint between the basement and areaway walls at relatively small wall deflections. The shear failure would result in the initiation of one-way wall action (i.e., the lateral support of the areaway wall would be lost) at a time shortly after arrival of the blast wave. However, as noted in the above tabulations for the nodes shown in Figure 14, the shear forces developed as a result of a 10-psi uniform static load decreased in magnitude from a maximum in the center portion of the wall to a minimum near the top and bottom supports; in particular, the shear forces at the nodes above the level of the door opening are much less than the maximum shear values. It is therefore reasonable to assume that under blast loading the one-way action wall without arching will have an effective span somewhat less than the total height of the basement wall. Therefore, for this study, the effective wall height for

for one-way action was assumed equal to the height of the door opening. The collapse overpressure was obtained for walls with the following properties and load conditions:

L_V = 80 in.

L_H = 92 to 360 in.

t_w = 8 in.

f'_{dc} = 3,750 psi

f_{dy} = 44,000 psi

p = 0.0015 A_c (vertical)

γ = 145 pcf

P_v = 0

Support case: one-way propped cantilever

S = 6.7 ft

W = 1 Mt

The collapse of these basement walls is predicted to occur at a blast overpressure of approximately 4.5 psi for all wall widths from L_H = 92 in to 360 in. For comparison, the collapse overpressure was also obtained for a 12-in.-thick reinforced concrete basement wall with the same properties as for the above 8-in. wall. The predicted collapse overpressure for the 12-in.-thick wall was found to be 9.2 psi for the same range of wall widths.

One-Way Concrete Wall (With Arching)

For a frame structure, it is possible that one-way arching, rather than one-way flexure, may occur in the basement wall subsequent to a shear failure at the basement/areaway wall construction joint. Therefore, calculations were performed to determine the blast strength of one-way arching walls. Since arching walls develop considerable more resistance than similar nonarching walls, it was felt to be more meaningful to use the full height of arching walls in the analysis, rather than the height of the door opening as was done for nonarching walls. The collapse overpressure was obtained for a wall with the following properties and load conditions:

L_{ℓ} = 120 in.

L_w = 92 to 360 in.

t_w = 8 in.

f'_c = 3,750 psi

γ = 145 pcf

Support case: one-way arching

S = 6.7 ft.

W = 1 Mt

The collapse of these basement walls is predicted to increase from a blast overpressure level of 6.9 psi for L_w = 92 in. to 10.1 psi for L_w = 360 in. The results of the analyses for both the one-way concrete walls with arching and without arching are shown in Figure 15.

Summary and Discussion

The primary purpose of this effort was to examine the dynamic response of conventional reinforced concrete basement walls located in areaways, and determine if such walls can resist a 10-psi blast overpressure. Since an adequate analytical model for predicting the collapse of basement walls with door openings was not available, it was necessary to perform several types of analyses so as to make a realistic estimate of the collapse strength of the walls. To provide uniformity for the various analyses, a standard basement wall with door opening was designed in accordance with the 1963 ACI code for reinforced concrete.

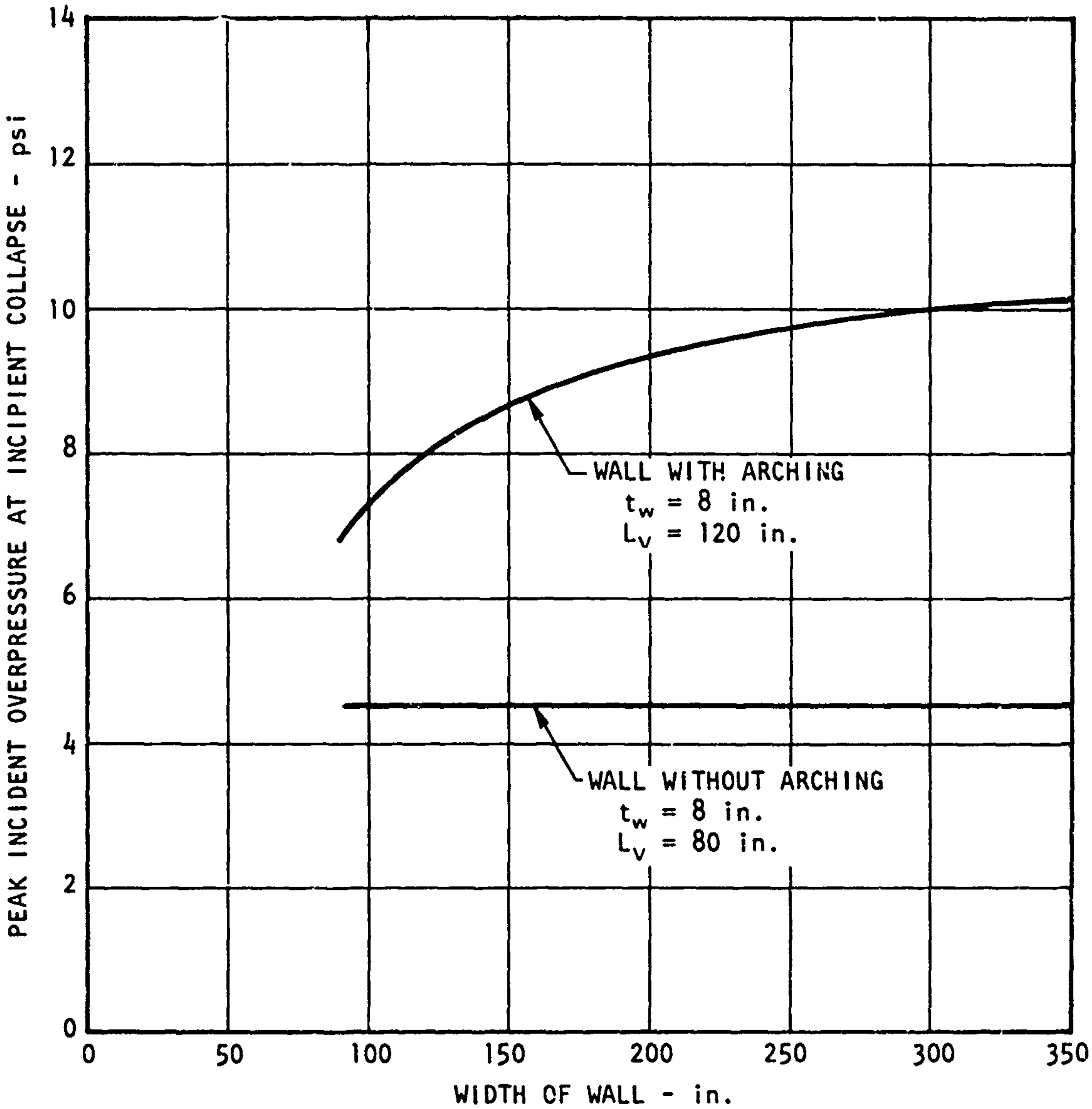


FIGURE 15 PEAK INCIDENT OVERPRESSURE AT INCIPIENT COLLAPSE VERSUS WALL WIDTH

One-Way Reinforced Concrete Basement Wall Without Arching
One-Way Concrete Basement Wall With Arching

An initial static analysis was made to compare the flexural resistances of two-way reinforced concrete walls with and without door openings. The results of this analysis (Figure 10) indicated that the flexural resistances of the two walls shown in Figure 9 were approximately equivalent. It was therefore warranted to use an available computer program to perform dynamic analyses of walls without door openings, and then to use these results for estimating the collapse overpressure of basement walls with door openings. The predicted collapse overpressures for two-way walls without door openings are shown in Figure 11 for 8- and 12-in.-thick reinforced concrete walls with wall heights of 96 and 120 in.

However, although the results of this analysis appeared reasonable, there was some co· ern because the available analytical model could not provide sufficient detailed information on the effect of the complex door opening geometry on the response of the wall. Therefore, an available static finite element computer program was used to analyze an 8-in.-thick two-way wall. The results for a 10-psi uniform lateral static load indicated that a shear failure was probable at the construction joint between the basement and areaway walls shown in Figure 12a. From the analysis of two walls with different widths, it was concluded that a shear failure would occur at relatively low overpressure levels if the areaway wall was located greater than abo 20-in. horizontal distance from the edge of the door opening. However, a shear failure in the basement wall at the basement/areaway wall joint does not necessarily result in collapse of the basement wall, since the wall may still resist the applied blast forces through one-way flexural or arching action between top and bottom supports subsequent to the shear failure and loss of side supports.

To determine the effect of a shear failure at the basement/areaway wall joint on the collapse strength of two-way walls, dynamic analyses were performed for one-way basement walls, both with and without arching. The results of the analyses are indicated in Figure 15, where it can be noted that 8-in.-thick reinforced concrete basement walls without arching are predicted to collapse at less than 5-psi blast overpressure. Although the predicted collapse overpressure for arching walls is much greater than for nonarching walls, for most of the wall widths examined the strength of arching walls is less than the 10-psi blast overpressure criterion. It should be mentioned that these results apply to the minimum thickness reinforced concrete basement wall, which has the minimum area of steel reinforcement permitted by the 1963 ACI building code for reinforced concrete.

Conclusions

From the various analyses performed, a few general conclusions can be made concerning the collapse of blast loaded reinforced concrete basement walls with door openings and located in areaways. First, for reinforced concrete walls 8-in. thick or thicker, and not over 10-ft high, it is probable that the wall strength of the weakest code-designed wall is sufficient to resist a 10-psi blast loading if the horizontal distance from the edge of the door opening to the areaway support wall is less than approximately 20 in. (L_w = 84 in.).

Second, for 8-in.-thick walls with horizontal distance between door opening and areaway wall greater than about 20 in., it will be necessary to strengthen the wall in the vicinity of the door opening so as to upgrade the wall to the 10-psi blast overpressure level.

Third, for reinforced concrete basement walls 12-in. thick or thicker, the blast strength can be expected to be approximately equal to or greater than the 10-psi blast overpressure criterion for all wall conditions.

Appendix A

LISTINGS OF COMPUTER PROGRAMS

Appendix A

LISTINGS OF COMPUTER PROGRAMS

Introduction

This appendix contains a printout of the listing for each program developed for DCPA for analyzing the dynamic response and collapse of walls and floor systems of existing buildings.

The programs were coded in FORTRAN and run on United Computing Systems, Inc., commercial time-sharing CDC 6400 computer (System UCS-VI); running on other systems may require minor modifications to the programs. For convenience and ease of use during the research effort, as well as by others later, the programs were written in an interactive or conversational mode.

To reduce the size of the computer central memory required, and thereby reduce the cost of running the programs, the Link Mode or chaining technique was used for the larger programs. Chaining has the advantage of reducing the overall cost of running programs, but a slightly more complicated technique is required to compile the programs in preparation for execution. Half of the programs were developed as chained programs.

Also included in this appendix are short summaries describing the function of each of the following eight programs:

- UNREINF, Unreinforced masonry wall without arching, see page 63
- ARCHING, Unreinforced masonry wall with arching, see page 81
- RCWALL, Reinforced concrete wall,* see page 95

---

* Link Mode or chained program.

- RCSLAB, Reinforced concrete slab,* see page 115
- RESTRAN, Restrained reinforced concrete slab, see page 133
- RCBEAM, Reinforced concrete support beam,* see page 147
- STBEAM, Steel support bean,* see page 165
- FLAT, Flat slab or flat plate, see page 181.

Following the summaries are the listings of the programs.

Summary of Computer Programs

Program UNREINF

Analyzes one-way and two-way unreinforced masonry walls (exterior or interior) without arching for a given load, or solves for incipient collapse load. Window openings may be included. Load types include: idealized blast loading (front, side, or rear face) with or without room filling; triangular load; rectangular load; URS tunnel loading; arbitrary load shape. Modulus of rupture and clearing distance may be randomly varied (normal distribution).

Subroutines:		
	Main Routine	COEF
	FORCE	TRANS
	FILL	WINDOW
	RESIST	RANDOM

Program ARCHING

Analyzes one-way and two-way unreinforced masonry walls (exterior or interior) with arching for a given load, or solves for incipient collapse load. Window openings may be included. Load types include: idealized blast loading (front, side, or rear face) with or without room filling; triangular load; rectangular load; URS tunnel loading; arbitrary load shape. Ultimate compressive strength, modulus of elasticity, and clearing distance may be randomly varied (normal distribution).

---

* Link Mode or chained program.

Subroutines:	Main Routine	RESIST
	FORCE	WINDOW
	FILL	RANDOM

Program RCWALL

Analyzes one-way and two-way reinforced concrete walls (exterior or interior) for a given load, or solves for incipient collapse load. Window openings may be included. Load types include: idealized blast loading (front, side, or rear face) with or without room filling; arbitrary load shape. Dynamic yield strength of reinforcement steel and clearing distance may be randomly varied (normal distribution).

Subroutines:	RCWALL1	RESIST
	WINDOW	MOMENT
	RCWALL2	COEF
	FORCE	TRANS
	FILL	RANDOM

Program RCSLAB

Analyzes one-way and two-way reinforced concrete floor slabs for a given load, or solves for incipient collapse load. Tensile membrane resistance may be included. Dynamic reactions may be output to a data file for use in analyzing support beams. Load types include: idealized blast loading (top face) with rise time equal to time required for blast wave to travel across short span; room filling pressure resulting from idealized blast loading; arbitrary load shape. Dynamic yield strength of reinforcement steel and clearing distance (for room filling load) may be randomly varied (normal distribution).

Subroutines:	RCSLAB1	RESIST
	COEF	MOMENT
	RCSLAB2	TRANS
	FORCE	RANDOM
	FILL	

Program RESTRAN

Analyzes two-way reinforced concrete floor slabs with edges restrained against lateral movement for a given load, or solves for incipient collapse load. Both compressive and tensile membrane behavior are included. Load types include: idealized blast loading (top face) with rise time equal to time required for blast wave to travel across short span; room filling pressure resulting from idealized blast loading; arbitrary load shape. Yield strength of reinforcement steel, concrete compressive strength, and clearing distance (for room filling load) may be randomly varied (normal distribution).

Subroutines:	Main Routine	MOMENT
	FORCE	TRANS
	FILL	RANDOM
	RESIST	

Program RCBEAM

Analyzes reinforced concrete beams (rectangular or T-beam) for a given load, or solves for incipient collapse load. Tensile membrane resistance may be included. Load types include: dynamic reactions from slab analysis (see RCSLAB); idealized blast loading acting on beam and area of slab supported by the beam with rise time equal to time required for blast wave to travel length of the beam; room filling pressure resulting from idealized blast loading acting on beam and area of slab supported by the beam; arbitrary load shape. Dynamic yield strength of reinforcement steel and clearing distance (for room filling load) may be randomly varied (normal distribution).

Subroutines:	RCBEAM1	RESIST
	RCBEAM2	MOMENT
	FORCE	COEF
	FILL	TRANS
		RANDOM

Program STBEAM

Analyzes structural steel beam (wide flange may include bottom steel cover plate and/or composite action with slab) for a given load, or solves for incipient collapse load. Load types include: dynamic reactions from slab analysis (see RCSLAB); idealized blast loading acting on beam and area of slab supported by the beam with rise time equal to time required for blast wave to travel length of the beam; room filling pressure resulting from idealized blast loading acting on beam and area of slab supported by the beam; arbitrary load shape. Dynamic yield strength of structural steel, dynamic yield strength of reinforcement steel (composite beam), and clearing distance (for room filling load) may be randomly varied (normal distribution).

Subroutines:	STBEAM1	RESIST
	STBEAM2	COEF
	FORCE	TRANS
	FILL	RANDOM

Program FLAT

Analyzes reinforced concrete flat slab floor system or flat plate floor system for a given load, or solves for incipient collapse load. Tensile membrane resistance may be included. Load types include: idealized blast load (top face) with rise time equal to the time required for blast wave to travel across span (slab assumed to be square); room filling pressure resulting from idealized blast loading; arbitrary load shape. Dynamic yield strength of reinforcement steel and clearing distance (for room filling load) may be randomly varied (normal distribution).

Subroutines:	Main Routine	RESIST
	FORCE	MOMENT
	FILL	RANDOM

UNREINF

Unreinforced Masonry Wall Without Arching

PROGRAM UNREINF

```
PRØGRAM JIMRØC(INPUT,ØUTPUT,TAPE1=ØUTPUT)
C:  THIS RØUTINE IS THE CØNTRØLLING RØUTINE FØR THE PRØGRAM USED
C:  IN THE ANALYSIS ØF ØNE-WAY ØR TWØ-WAY ACTIØN WALLS.
C
      CØMMØN Y(100),YU,YFAIL,Q,QU,AREA,ZMASS,ZKLM,VH1,VH2,VV1,VV2
      CØMMØN KWALL,KINC,KRF,KRAND,I,ICASE,FU,VFAIL,FR,FPM,EM,FDY
      CØMMØN FDC,D(4),LDTYPE,PEXT,PF,PS3,PDØ,PC,TC,TØ,P3,TIME,L,S
      CØMMØN /RAND/  TIMEC,IWALL
      DIMENSIØN A(100),V(100),T(100),VV(100),VH(100),
+       PEX(100),PIN(100),PN(100)
C
C:  READ TITLE AND CØNTRØL PARAMETERS
5     PRINT 67
      READ 68,TITLE
      PRINT 95
      READ,KWALL KINC,LDTYPE,KRF,KRAND
      DELAY=0
      VFAIL=1E10
      CALL RESIST(1)
      CALL FØRCE(1)
      IF(KRF.EQ.0)GØTØ 12
      CALL FILL(PINT,1)
12    IF(KWALL.EQ.0)GØTØ 14
      PRINT 96
      READ,DELAY
      DELAY=DELAY/1000.0
14    IF(KRAND.NE.1)GØTØ 35
      CALL FØRCE(4)
      CALL RANDØM(1)
34    CALL RANDØM(2)
35    CALL RESIST(3)
C
C: MINIMUM, MAXIMUM, AND STARTING VALUES ARE DETERMINED FØR CASES
C: WHERE THE LØAD CAUSING INCIPIENT CØLLAPSE IS TØ BE FØUND
13    IF(KINC.EQ.0)GØTØ 23
       PF=QU
       PFMAX=0
       PFMIN=PF/2.0
      GØTØ 20
16     PF=(PFMIN+PFMAX)/2.0
20    CALL FØRCE(2)
23    IF(KRF.EQ.0)GØTØ 24
      CALL FILL(PINT,2)
C
C: INITIALIZE VALUES FØR BETA METHØD (BETA = 1/6) AND CØMPUTE VALU
C: FØR FIRST TIME INTERVAL ASSUMING WALL INITIALLY AT REST
24     I=1
       TIME=0
      T(1)=0$ V(1)=0$ Y(1)=0
       DELTA=0.001
      IF(KWALL.EQ.0)GØTØ 29
27    IF(TIME.GE.(DELAY-0.00001))GØTØ 28
      TIME=TIME+DELTA
      CALL FILL(PINT,3)
      GØTØ 27
28    PIN(1)=PINT
      TPNET=AREA*PINT
      T(1)=TIME
      GØTØ 30
29    CALL FØRCE(3)
      PEX(I)=PEXT
      TPNET=AREA*PEXT
       PN(1)=PEXT
30    CALL RESIST(2)
       A(1)=TPNET/(ZMASS*ZKLM)
       VV(1)=VV1*TPNET
       VH(1)=VH1*TPNET
C
C: PRØCEDURE FØR ALL SUBSEQUENT TIME INTERVALS
1      I=I+1
      IF(I.LT.101)GØTØ 11
      PRINT 98,TIME
98    FØRMAT(/,*I=101:  TIME =*,F6.3,*:  WALL ASSUMED TØ NØT FAIL*)
      GØTØ 6
```

PROGRAM UNREINF (CONTINUED)

```
 11   TIME=TIME+DELTA
       T(I)=TIME
       A(I)=A(I-1)
       CALL FORCE(3)
       PEX(I)=PEXT
      IF(KWALL.EQ.1)GOTO 10
      IF(KRF.EQ.0)GOTO 3
      CALL FILL(PINT,3)
       PIN(I)=PINT
      TPNET=AREA*(PEXT-PINT)
       GOTO 2
 3     TPNET=AREA*PEXT
      GOTO 2
 10   CALL FILL(PINT,3)
      PIN(I)=PINT
      TPNET=AREA*PINT
 2    PN(I)=TPNET/AREA
      DO 8 JJ=1,10
       Y(I)=Y(I-1)+DELTA*V(I-1)+DELTA*DELTA*(A(I-1)/3.+A(I)/6.)
       CALL RESIST(2)
       QT=Q*AREA
 4     ANEW=(TPNET-QT)/(ZMASS*ZKL4)
       ADELTA=ANEW-A(I)
       A(I)=ANEW
 IF(ANEW.EQ.0)PRINT,*1985*,TIME,TPNET,QT,ZMASS,ZKL4,Y(I),A(I-1)
      IF(ABS(ADELTA/(ANEW+1.0E-10)).LT.0.01)GOTO 9
 8     CONTINUE
       A(I)=ANEW-ADELTA/2.0
       WRITE(1,80)TIME,PF,A(I),Y(I)
 9     CONTINUE
       Y(I)=Y(I-1)+DELTA*V(I-1)+DELTA*DELTA*(A(I-1)/3.+A(I)/6.)
       V(I)=V(I-1)+DELTA*(A(I)+A(I-1))/2.0
       VV(I)=VV1*TPNET+VV2*QT
       VH(I)=VH1*TPNET+VH2*QT
      IF(VV(I).GT.VFAIL)GOTO 7
C
C: CHECK FOR MAXIMUM DEFLECTION OR FAILURE OF WALL
C:   IF MAXIMUM DEFLECTION REACHED, WALL DID NOT FAIL
      IF(Y(I).LE.Y(I-1).AND.PN(I).LE.PN(I-1))GOTO 6
      IF(Y(I).LT.0)GOTO 6
      IF(TIME-DELAY.GE.0.010)DELTA=0.002
      IF(Y(I).LT.YJ)GOTO 1
      IF(TIME-DELAY.GE.0.020)DELTA=0.005
      IF(TIME-DELAY.GE.0.100)DELTA=0.010
      IF(TIME-DELAY.GE.0.500)DELTA=0.050
C   IF FAILURE DEFLECTION REACHED, WALL FAILED
      IF(Y(I).GE.YFAIL)GOTO 7
       GOTO 1
C
C: INTERVAL HALVING PROCEDURE TO DETERMINE LOAD CAUSING INCIPIENT
C: COLLAPSE FOR CASES WHERE DESIRED
C:   WALL DID NOT FAIL - SET PFMIN TO PF
 6     CONTINUE
      IF(KRAND.EQ.1)GOTO 36
      IF(KINC.EQ.0)GOTO 18
 36   PFMIN=PF
       IF(PFMAX.GT.0)GOTO 17
       PF=2.0*PF
       GOTO 20
C:   WALL FAILED - SET PFMAX TO PF
 7     CONTINUE
       TIMEC=TIME
      IF(KRAND.EQ.1)GOTO 37
      IF(KINC.EQ.0)GOTO 18
 37   PFMAX=PF
C:   CHECK TO SEE IF LOAD RANGE IS WITHIN DESIRED ACCURACY
 17    IF((PFMAX-PFMIN)/PFMIN.GT.0.01)GOTO 16
      IF(KRAND.NE.1)GOTO 18
      CALL RANDOM(3)
      GOTO 34
C
C: OUTPUT DATA INCLUDES THE MAXIMUM DEFLECTION AND TIME OF
C: OCCURANCE FOR A NON-FAILING WALL OR THE TIME AND VELOCITY
C: AT COLLAPSE FOR A FAILING WALL.  OPTIONAL OUTPUT IS THE
```

PROGRAM UNREINF (CONTINUED)

```
C: ENTIRE BEHAVIOR TIME-HISTORY OF THE WALL.
C
C: OUTPUT LOAD DATA
 18     CALL FORCE(4)
C
C: OUTPUT FINAL RESULTS
        IF(Y(I).LT.YFAIL)WRITE(1,70)Y(I),T(I)
        IF(Y(I).GE.YFAIL)WRITE(1,71)T(I),V(I)
C
C: CHECK TO SEE IF ENTIRE TIME-HISTORY OF WALL IS DESIRED
        WRITE(1,72)
 READ,M
        IF(M.EQ.0)GOTO 25
       IF(KWALL.EQ.1)GOTO 32
       IF(KRF.EQ.0)GOTO 26
        WRITE(1,75)(T(J),PEX(J),PIN(J),PN(J),Y(J),VV(J),VH(J),J=1,I)
        GOTO 25
 26     WRITE(1,76)(T(J),PEX(J),A(J),V(J),Y(J),VV(J),VH(J),J=1,I)
       GOTO 25
 32    WRITE(1,76)(T(J),PIN(J),A(J),V(J),Y(J),VV(J),VH(J),J=1,I)
 25     WRITE(1,77)
        GOTO 5
C
 67 FORMAT(/*INPUT TITLE*,+)
 68 FORMAT(A59)
 70 FORMAT(/*WALL DID NOT FAIL - MAX. DEFLECTION OF*F6.2
+     * IN. REACHED AT*F7.3,* SEC*)
 71 FORMAT(/*WALL FAILED AT*,F7.3,* SEC (FINAL VELOCITY =*
+    F7.2* IN./SEC)*)
 72 FORMAT(/*IS TIME HISTORY OF WALL DESIRED (YES=1, NO=0)*,+)
 75 FORMAT(/,15X,*PRESSURE ON WALL*/*  TIME    EXTERIOR  *
+ *INTERIOR     NET      DISPLACEMENT     VV       VH*/
+    (F6.3,3F10.3,F12.4,F11.0,F8.0))
 76 FORMAT(/*  TIME  PRESSURE  ACCELERATION  VELOCITY  *
+    *DISPLACEMENT     VV       VH*/
+    (F6.3,F9.3,F12.1,F12.2,F12.4,F11.0,F8.0))
 77     FORMAT(///,7(*-----------*))
 90 FORMAT(/*ACCELERATION NOT CONVERGING AT TIME =*,F6.3,
+            * SEC (PF =*,F7.3,* PSI)*/*     A(I) SET EQUAL TO*,
+            F8.1,* (AVG OF LAST 2 ITERATIONS)*/*     Y(I) =*,
+            F8.4,* IN.*)
 95 FORMAT(/*INPUT KWALL(0=EXT,1=INT),KINC,LOTYPE,KRF,KRAND*,
+    *(1=RANDOM)*)
 96 FORMAT(/*INPUT DELAY TIME (MSEC) TO INITIAL LOADING OF*,
+    * INTERIOR WALL*,+)
C
 999    STOP
       END
 SUBROUTINE FORCE(IENTRY)
C  THIS SUBROUTINE INPUTS THE LOAD PARAMETERS AND DETERMINES
C  THE LOAD AT A GIVEN TIME FOR THE FOLLOWING LOAD TYPES:
C      1. IDEALIZED BLAST LOAD (FRONT OR SIDE FACE)
C
 COMMON Y(100),YU,YFAIL,Q,QU,AREA,ZMASS,ZKLM,VH1,VH2,VV1,VV2
 COMMON KWALL,KINC,KRF,KRAND,I,ICASE,FU,VFAIL,FR,FPM,EM,FDY
 COMMON FDC,D(4),LDTYPE,P,PR,PSO,PDO,PC,TC,TO,PO,TIME,LL,S
C
 GOTO(100,200,300,4),IENTRY
C
C  INPUT LOAD PARAMETERS
 100 IF(KRAND.EQ.0)GOTO 102
 W=1000 $ PO=14.7 $ CO=1120.0 $ LOC=1
 RETURN
 102 PRINT 600
 READ,W,PO,CO,LOC,S
 IF(LOC.EQ.1)GOTO 105
 PRINT 605
 READ,ZLEN,CD
 105 IF(KINC.EQ.1)RETURN
 PRINT 630
 READ,PSO
 PR=2.0*PSO*(7.0*PO+4.0*PSO)/(7.0*PO+PSO)
 GOTO 215
C
```

PROGRAM UNREINF (CONTINUED)

```
C  CALCULATE LOAD PROPERTIES FOR GIVEN PEAK PRESSURE
 200 GOTO(205,210),LOC
 205 PS0=(PR-14.0*P0+SQRT(196.0*P0*P0+196.0*P0*PR+PR*PR))/16.0
 GOTO 215
 210 PS0=PR
 215 PD0=2.5*PS0*PS0/(7.0*P0+PS0)
 U=C0*SQRT(1.0+(6.0*PS0)/(7.0*P0))
 TO=W**0.3333/(2.2399+0.1886*PS0)
 GOTO(220,[illegible].OC
 220 TC=3.0*[illegible]
 PC=PS0*(1-TC/TO)*EXP(-TC/TO)+PD0*(1-TC/TO)**2*EXP(-2*TC/TO)
 CD=1.0
 RETURN
 225 TA=ZLEN/U
 TA2=TA/2.0
 TA2TO=TA2/TO
 PA=PS0*(1-TA2TO)*EXP(-TA2TO)+CD*PD0*(1-TA2TO)**2*EXP(-2*TA2TO)
 RETURN
C
C  CALCULATE LOAD
 300 GOTO(305,310),LOC
 305 TTO=TIME/TO
 IF(TIME.GT.TC)GOTO 320
 P=PC+(TC-TIME)*(PR-PC)/TC
 RETURN
 310 TTO=(TIME-TA2)/TO
 IF(TIME.GT.TA)GOTO 320
 P=PA*TIME/TA
 RETURN
 320 IF(TTO.GE.1.0)GOTO 330
 P=PS0*(1-TTO)*EXP(-TTO)+CD*PD0*(1-TTO)**2*EXP(-2*TTO)
 RETURN
 330 P=0
 RETURN
C
C  PRINT LOAD DATA
 4 IF(KINC.EQ.0)GOTO 400
 PRINT 640,LDTYPE
 GOTO 410
 400 PRINT 645,LDTYPE
 410 CONTINUE
 415 GOTO(420,425),LOC
 420 PRINT 650
 GOTO 430
 425 PRINT 655
 430 PRINT 660,W,P0,C0
 IF(KRAND.NE.0)RETURN
 GOTO(435,440),LOC
 435 PRINT 665,S,TC,PR
 GOTO 445
 440 PRINT 670,ZLEN,TA,PA
 445 PRINT 675,U,TO,CD,PS0,PD0
 RETURN
C
 600 FORMAT(/*INPUT W,P0,C0,LOC,S*,*)
 605 FORMAT(/*INPUT L,CD*,*)
 630 FORMAT(/*INPUT PS0*,*)
 640 FORMAT(/*LOAD CAUSING INCIPIENT FAILURE IS AS FOLLOWS:*,
+    /,5X,*LOAD TYPE NUMBER*,I2)
 645 FORMAT(/*PROPERTIES OF LOAD ACTING ON WALL ARE AS FOLLOWS:*,
+    /,5X,*LOAD TYPE NUMBER*,I2)
 650 FORMAT(8X,*(FRONT FACE)*)
 655 FORMAT(8X,*(SIDE FACE)*)
 660 FORMAT(10X,*W =*,F8.1,* KT     P0 =*,F6.2,* PSI      C0 =*,
+    F7.1,* FPS*)
 665 FORMAT(10X,*S =*,F6.1,* FT      TC =*,F6.3,* SEC      PR =*,
+    F7.3,* PSI*)
 670 FORMAT(10X,*L =*,F6.1,* FT      TA =*,F6.3,* SEC      PA =*,
+    F7.3,* PSI*)
 675 FORMAT(10X,*U =*,F7.1,* FPS    TO =*,F6.3,* SEC      CD =*,
+    F5.1,/,8X,*PS0 =*,F7.3,* PSI   PD0 =*,F7.3,* PSI*)
 END
```

PROGRAM UNREINF (CONTINUED)

```
SUBROUTINE FILL(P3,IENTRY)
C: COMPUTES AVERAGE AIR PRESSURE IN ROOM DUE TO BLAST WAVE
C: INCIDENT HEAD-ON UPON FRONT WALL.
C
COMMON Y(100),YU,YFAIL,Q,QU,AREA,ZMASS,ZKLM,VH1,VH2,VV1,VV2
COMMON KWALL,KINC,KRF,KRAND,II,ICASE,FU,VFAIL,FR,FPM,EM,FDY
COMMON FOC,D(4),LDTYPE,PEXT,PR,PS0,PD0,PC,TC,TO,P0,TIME,L,S
DIMENSION AA(8,2),NN(8)
LOGICAL L1,L2,L3
GOTO(10,13,11),IENTRY
10 PRINT 700
RHO0=0.076 $ L1=.FALSE.
READ,NWIN,V3
AT=0$ AFRONT=0$ ASIDE=0
DO 18 I=1,NWIN
PRINT 710,I
READ,AA(I,1),NN(I),AA(I,2)
AA(I,2)=AA(I,2)/1000.0
AT=AT+AA(I,1)
M=NN(I)$ GOTO(12,14,14),M
12 AFRONT=AFRONT+AA(I,1)
GOTO 18
14 ASIDE=ASIDE+AA(I,1)
18 CONTINUE
AFRONT=AFRONT/AT$ ASIDE=ASIDE/AT
700 FORMAT(/*INPUT NWIN AND ROOM VOLUME (CF)*,/)
710 FORMAT(/*INPUT AREA (SQ FT),LOCATION CODE & DELAY(MSEC)*
+ * FOR WINDOW*,I2,/)
G=1.4 $ G2=1./G $ G3=1.-G2 $ G4=2./G3 $ G5=G+1.
G6=2.*G/G5 $ G7=(G-1.)/G5
PP2=.1912
C=SQRT(G*P0*32.*144./RHO0)
TAU=2.*(V3**(1./3.))/C
NSTEP=4
DT=TAU/NSTEP
RETURN
C
13 P30=P0
TT=0.$ T0=0.
RHO30=RHO0
L2=.FALSE. $ L3=.FALSE.
RETURN
C
11 IF(L1)GOTO 52
IF(L2.A.L3)GOTO 9
52 DDT=(TIME-T0)*0.5
ISTOP=2
53 IF(DDT.LT.DT)GOTO 51
50 DDT=0.5*DDT
ISTOP=2*ISTOP
GO TO 53
51 CONTINUE
DO 99 I=1,ISTOP
TT=T0+I*DDT
IF(TT.GT.TO)GO TO 99
DM=0. $ WW=0. $ NW=0
DO 500 K=1,NWIN
M=NN(K) $ DLY=AA(K,2)+0.000001
IF(DLY.GE.TT)GO TO 500
GOTO(15,16,16),M
15 CDF=1.0
IF(TT-TC)20,20,21
20  P11=(TC-TT)*(PR-PC)/TC+PC
P11=P11+P0
GO TO 30
16 CDF=-0.4
21  R=TT/TO $ RR=1.-R
PD=PD0*RR*RR*EXP(-2.*R)
PS=PS0*RR*EXP(-R)
P11=PS+CDF*PD
P11=P11+P0
30  RHO1=RHO0*((P11/P0)**G2)
IF(P11-P30)36,36,37
36  JSIGN=-1
```

PROGRAM UNREINF (CONTINUED)

```
20680 L2=.TRUE.
20770 303  P2=P11
20780 RHØ2=((P2/P33)**G2)*RHØ33
20790 X=P33/RHØ33
20800 GØ TØ 38
20810 37  JSIGN=+1
20820 306 P2=PP2*P11
20830 RHØ2=((P2/P11)**G2)*RHØ1
20840 X=P11/RHØ1
20850 38  U22=G4*(X-P2/RHØ2)*32.*144.
20860 IF(U22)40,39,39
20870 40  PRINT,*U22 NEGATIVE*,U22
20880 STØP
20890 39  U2=SQRT(U22)*JSIGN
20900 DDM=U2*RHØ2*AA(K,1)*DDT
20910 DM=DM+DDM
20920 WW=WW+P11*DDM/(G3*RHØ1)
20930 500 CØNTINUE
20940 P30=P30+(G-1.)*WW/V3
20950 RHØ30=RHØ30+DM/V3
20960 99  CØNTINUE
20970 TØ=TT
20980 P3=P30-P3
20982 IF(TIME.GE.TC)L3=.TRUE.
20983 RETURN
20984 9 R=TIME/TO $ RR=1.0-R
20985 PD=PD0*RR*RR*EXP(-2.0*R)
20986 PS=PS0*RR*EXP(-R)
20987 P3=PS+PD*(AFRØNT-0.4*ASIDE)
20990 999 RETURN
21020 END
30000 SUBRØUTINE RESIST (IENTRY)
30010C
30020C:  THIS SUBRØUTINE DETERMINES THE RESISTANCE FUNCTIØN FØR AN
30030C:  UNREINFØRCED MASØNRY WALL WITH ØR WITHØUT ØPENINGS.  CASES
30040C:  1-4 ARE TWØ-WAY WALLS AND CASES 5-7 ARE ØNE-WAY WALLS
30050C
30060 CØMMØN Y(100),YU,YFAIL,Q,QU,AREA,ZMASS,ZKLM,VH1,VH2,VV1,VV2
30070 CØMMØN KWALL,KINC,KRF,KRAND,I,ICASE,FU,VFAIL,FR,FPM,EM,FDY
30080 CØMMØN FDC,D(4),LDTYPE,PEXT,PF,PS0,PD0,PC,TC,TO,P0,TIME,L,S
30085 CØMMØN /RAND/ TIMEC,IWALL
30090 REAL K1,K2,K3,KUD,N,IC,ICTØT,IG,KK1,KK2,KK3,MM,MPR,MPRSQ
30100 REAL KS,KF,KEP,KEQ,MU,MU
30110C
30120 GØTØ(5,500,262),IENTRY
30130C
30140C:  INPUT WALL PARAMETERS
30150 5 WRITE(1,600)
30160 READ,ZLV,ZLH,TW,PV,E,FR,ICASE,ZLVW,ZLHW,GAMMA
30170 WRITE(1,670)
30180C
30190C:  DETERMINE ELASTIC DEFLECTIØN AND MØMENT CØEFFICIENTS FØR
30200C:  TWØ-WAY WALLS WITHØUT INPLANE FØRCES
30205 IWALL=1
30210 R=ZLH/ZLV
30220 ALP=1.0/R$ ALP2=ALP*ALP
30230 IF(ICASE.LE.4)GØTØ 11
30240 R=0$ ALP=0$ ALP2=0
30250 11 AWALL=ZLV*ZLH
30260 AWIN=ZLVW*ZLHW
30270 AREA=AWALL-AWIN
30280 ZMASS=GAMMA*AREA*TW/(386.07*1728.0)
30290 B=0.5*(ALP*SQRT(3.0+ALP2)-ALP2)
30300 IG=TW**3/12.0
30310 CALL CØEF (ICASE,R,ASS,BSS,AF,BF,IG,ZLV,ZLH,PV,NX,CF,E,1)
30320 CALL TRANS (B,ZLV,ZLH,ICASE,0,ZKLM,ZKLMSE,ZKLMFE,ZKLMP,VH1S,
30330+   VH2S,VV1S,VV2S,VH1F,VH2F,VV1F,VV2F,VH1P,VH2P,VV1P,VV2P)
30340C
30350C:  DETERMINE MØDIFICATIØN FACTØR FØR WALL WITH WINDØWS
30360C
30370 290 QMULT=1.0
30380 IF(AWIN.NE.0)CALL WINDØW (QMULT,ZLV,ZLH,ZLVW,ZLHW,AWIN,AWALL,
30381+      R,ICASE)
30390 WRITE(1,620)ICASE,ZLV,ZLH,ALP,TW,FR,E,PV,GAMMA,ZLVW,ZLHW,QMULT
```

PROGRAM UNREINF (CONTINUED)

```
RETURN
C
C:  DETERMINE MAXIMUM RESISTANCE DURING DECAYING PHASE
C
262 MM=(FR+PV/TW)*TW*TW/6.0
W=ZLV*TW*GAMMA/1728.0
IF(ICASE.GT.4)GØTØ 278
QEZERØ=12.0*TW*(2.0*PV+W)*(1.0+0.5*ALP2/B)/(ZLV*ZLV*(3-2*B))
GØTØ 279
278 QEZERØ=8.0*TW*(PV+0.25*W)/(ZLV*ZLV)
279 CØNTINUE
YFAIL=TW
KEQ=QEZERØ/TW
C
C:  DETERMINE MAXIMUM RESISTANCE DURING INITIAL (FLEXURAL) PHASE
QU=MM/(BSS*ZLV*ZLV)
KS=E*IG/(ASS*ZLV**4)
YU=QU/KS
IF(ICASE.EQ.1.ØR.ICASE.EQ.5)GØTØ 280
C
C:  CASES 2,3,4
Q1=MM/(BF*ZLV*ZLV)
KF=E*IG/(AF*ZLV**4)
Y1=Q1/KF
KEP=(QU-Q1)/(YU-Y1)
GØTØ 280
C
C:  DETERMINE WHETHER BENDING ØR EQUILIBRIUM RESISTANCE IS LARGER
C
280 IF(QU.LE.QEZERØ)GØTØ 285
C
C:  QU>QEZERØ
Y2=YU
Q2=QEZERØ*(1.0-YU/TW)
GØTØ 295
C
C:  QEZERØ>QU
285 Y2=QEZERØ/(KS+KEQ)
Q2=KS*Y2
295 CØNTINUE
C
C:  MØDIFY RESISTANCE VALUES BY APPRØPRIATE FACTØR
310 Q1=Q1*QMULT
Q2=Q2*QMULT
QU=QU*QMULT
KS=KS*QMULT$ KF=KF*QMULT$ KEP=KEP*QMULT$ KEQ=KEQ*QMULT
C
C:  ØUTPUT LØAD-DEFLECTIØN CURVE
IF(KRAND.EQ.1)GØTØ 325
PRINT 650
IF(ICASE.EQ.1.ØR.ICASE.EQ.5)GØTØ 320
WRITE(1,660)Q1,Y1
320 XXXXX=0.0
WRITE(1,660)QU,YU,Q2,Y2,XXXXX,YFAIL
325 RETURN
C
C:  DETERMINE THE RESISTANCE (PER UNIT AREA) ØF THE WALL AS
C:  A FUNCTIØN ØF Y(I)
C
500 IF(Y(I).GT.Y2)GØTØ 520
IF(Y(I).GT.YU)GØTØ 502
GØTØ(502,510,510,510,502,510,510),ICASE
C
C:  ELASTIC: PHASE -- CASE 1
502 Q=Y(I)*KS
505 ZKLM=ZKLMSE
VH1=VH1S$ VH2=VH2S
VV1=VV1S$ VV2=VV2S
RETURN
C
510 IF(Y(I).GT.Y1)GØTØ 515
C
C:  ELASTIC PHASE -- CASES 2,3,4
Q=Y(I)*KF
```

```
ZKLM=ZKLMFE
VH1=VH1F$ VH2=VH2F
VV1=VV1F$ VV2=VV2F
RETURN
C
C:   *ELASTO-PLASTIC* PHASE (CASES 2,3,4)
515 Q=Q1+KEP*(Y(I)-Y1)
GOTO 505
C
C:   SECONDARY (EQUILIBRIUM) PHASE
520 IF(Y(I).GT.TW)GOTO 525
Q=KEQ*(TW-Y(I))
ZKLM=ZKLMP
VH1=VH1P$ VH2=VH2P
VV1=VV1P$ VV2=VV2P
RETURN
C
C:   WALL COLLAPSED -- NO RESISTANCE (TO AVOID DIFFICULTIES
C:   FOR CERTAIN CASES SET RESISTANCE EQUAL TO VERY SMALL VALUE)
525 Q=1E-10
RETURN
C
C
600 FORMAT(/*INPUT LV,LH,TW,PV,E,FR,ICASE,LVW,LHW GAMMA*)
620 FORMAT(/*WALL PROPERTIES -- SUPPORT TYPE NO.*,I2/
+ *  LV =*,F6.1,* IN.      LH =*,F6.1,* IN.    LV/LH =*,
+ F7.2/*  TW =*,F6.1,* IN.       FR =*,F7.1,* PSI*,5X,
+ * E =*,F10.1,* PSI*/*  PV =*,F6.1,* LB/IN.*,5X,
+ *GAMMA =*,F6.1,* PCF*,/,* LVW =*,F6.1,* IN.    LHW =*,
+ F6.1,* IN.   QMULT =*,F6.3)
650 FORMAT(/*LOAD-DEFLECTION CURVE*,/,3X,*Q (PSI)*,4X,*Y (IN.)*)
660 FORMAT(F9.2,F12.4)
670 FORMAT(1H )
C
END
SUBROUTINE COEF(ICASE,R,ASS,BSS,AF,BF,I,ZLV,ZLH,PV,NX,CF,
+     E,IENTRY)
C:   THIS SUBROUTINE DETERMINES MOMENT AND DEFLECTION COEFFICIENTS
C:   FOR ONE-WAY (CASES 5-7) AND TWO-WAY (CASES 1-4) WALLS
C
      REAL I,MPR,MPRSQ,NU
      IF(IENTRY.EQ.2)GOTO 200
      NX=1
       IF(ICASE.GT.4)GOTO 50
C
       R2=R*R
       R3=R*R2
       R4=R2*R2
       ASS=-.007030+.013990*R-.003456*R2+.000286*R3
       BSS=-.058332+.139314*R-.035609*R2+.003016*R3
8      GOTO(41,20,30,40),ICASE
C
C:   CASE 2. FIXED ON FOUR SIDES
20     NX=3
       AF=-.003430+.007327*R-.003365*R2+.0006646*R3-.00004766*R4
       BF=-.101150+.260875*R-.138992*R2+.034677*R3-.004016*R4
+    +.000170*R**5
      CF=-.1674+.3554*R-.1714*R2+.0286*R3
       GOTO 41
C
C:   CASE 3. FIXED ON SHORT SIDES, SIMPLY SUPPORTED ON LONG SIDES
30      NX=4
       AF=.004513-.017525*R+.023095*R2-.010325*R3+.002187*R4
+  -.0002208*R**5 + .000008409*R**6
      BF=-.122149+.313445*R-.153979*R2+.036132*R3-.004015*R4
+  +.0001646*R**5
      CF=2.1959-7.7564*R+10.8376*R2-7.2495*R3+2.344*R4
+        -.2954*R**5
       GOTO 41
C
C:   CASE 4. SIMPLY SUPPORTED ON SHORT SIDES, FIXED ON LONG SIDES
40     NX=3
       AF=-.002765+.008652*R-.005695*R2+.001829*R3-.0002859*R4
+  +.00001739*R**5
```

PROGRAM UNREINF (CONTINUED)

```
           BF=-.060320+.256515*R-.175648*R2+.057928*R3-.009227*R4
+      +.000569*R**5
 CF=5.8987*R-1.6669-7.9398*R2+5.3142*R3-1.7623*R4+.2313*R**5
C
 41     IF(R.GT.2.0)CF=1.0/12.0
        IF(PV.EQ.0)RETURN
        ARATIO=AF/ASS$  BRATIO=BF/BSS
        BFO=BF$ CFO=CF
        GOTO 220
C
 50     IF(PV.NE.0)GOTO 300
C:   CASE 5. ONE-WAY SIMPLY SUPPORTED WALL
        ASS=5.0/384.0
          BSS=0.125
          GOTO(270,270,270,270,270,60,70),ICASE
C
C:   CASE 6. ONE-WAY FIXED END WALL
 60       AF=1.0/384.0
          BF=1.0/12.0
        CF=1.0/12.0
        NX=3
        RETURN
C
C:   CASE 7. ONE-WAY PROPPED CANTILEVER WALL
 70       AF=1.0/185.0
          BF=0.125
        CF=0.125
        NX=3
        RETURN
C
 200    IF(ICASE.GT.4)GOTO 300
C:   DETERMINE ELASTIC DEFLECTION AND MOMENT COEFFICIENT FOR
C:   TWO-WAY WALL WITH INPLANE FORCES
 220    PI=3.14159165
          NU=0.3
        PE=4.0*PI*PI*E*I/(ZLV*ZLV*(1.0-NU*NU))
          BV=0
 230    AV=0
          PPE=PV/PE
          TERM6=4.0*PI*PI*R*SQRT(PPE)
C
C:   SERIES SOLUTION USED TO DETERMINE COEFFICIENTS
          DO 250 M=1,7,2
          MPR=M*PI*R
          MPRSQ=MPR**2
          GMSQ=MPRSQ+2.0*MPR*PI*SQRT(PPE)
          EMSQ=MPRSQ-2.0*MPR*PI*SQRT(PPE)
          TERM5=M*MPRSQ*(MPRSQ-4.0*PI*PI*PPE)
          COSHGM2=0.5*(EXP(0.5*SQRT(GMSQ))+EXP(-0.5*SQRT(GMSQ)))
          IF(EMSQ.LT.0)GOTO 240
          COSHEM2=0.5*(EXP(0.5*SQRT(EMSQ))+EXP(-0.5*SQRT(EMSQ)))
          GOTO 245
 240    COSHEM2=COS(0.5*SQRT(-EMSQ))
 245    AV=AV+(1.0+(EMSQ/COSHGM2-GMSQ/COSHEM2)/(M*TERM6))
+   *(-1)**((M-1)/2)/TERM5
          BV=BV+(MPRSQ*(GMSQ*(NU*EMSQ-MPRSQ)/COSHEM2-EMSQ*(NU*GMSQ
+           -MPRSQ)/COSHGM2)/(M*TERM6))*(-1)**((M-1)/2)/TERM5
 250    CONTINUE
C
C:   CASE 1
        AVSS=AV*(1.0-NU*NU)*R4*4.0/PI
          BVSS=BV*R2*4.0/PI
          IF(ICASE.EQ.1)GOTO 260
C
C:   CASES 2, 3, AND 4
        AVF=AVSS*ARATIO
        BVF=BVSS*BRATIO
        CF=CFO*BVF/BFO
 259    AF=AVF
          BF=BVF
 260    ASS=AVSS
          BSS=BVSS
 270    RETURN
```

```
C
C:  ONE-WAY WALLS
300  EIPV=E*I/PV
     U=ZLV/SQRT(EIPV)
     U2=0.5*U
     TERM1=1.0/COS(U2)-1.0
C
C:  CASE 5. ONE-WAY SIMPLY SUPPORTED WALL
     BSS=TERM1/U**2
     ASS=(BSS-0.125)/U**2
     GOTO(270,270,270,270,270,310,320),ICASE
C
C:  CASE 6. ONE-WAY FIXED END WALL
310  NX=3
     BF=(1.0-U2/TAN(U2))/U**2
     AF=-BF+BSS+ASS
     RETURN
C
C:  CASE 7. ONE-WAY PROPPED CANTILEVER WALL
320 NX=3
     BF=TAN(U)*(TAN(U2)-U2)/(U*(TAN(U)-U))
     AF=(BF*(0.5+SIN(U2)/TAN(U)-COS(U2))-(SIN(U2)/TAN(U)
+       -COS(U2)-SIN(U2)/SIN(U)+0.125*U*U+1.0)/U**2)/U**2
     RETURN
999 END
     SUBROUTINE TRANS (B,ZLV,ZLH,ICASE,KRAK,ZKLM,ZKLMSE,ZKLMFE,
+     ZKLMP,VL1S,VL2S,VS1S,VS2S,VL1F,VL2F,VS1F,VS2F,VL1P,VL2P,
+     VS1P,VS2P)
C
C: THIS SUBROUTINE DETERMINES LOAD AND MASS TRANSFORMATION FACTORS
C: AND DYNAMIC REACTION COEFFICIENTS FOR TWO-WAY WALLS.
C
C: DETERMINE LOAD AND MASS TRANSFORMATION FACTORS
     B2=B*B
     B3=B*B2
     B4=B2*B2
     B5=B2*B3
     B6=B3*B3
C
C: CASES 1 & 5 -- ELASTIC RANGE
330  ZKMSE1=20.48*B3*(1./12.-B2/7.5+B3/21+B4/14-B5/18+B6/90)
     ZKMSE2=0.5038-0.7066*B
     ZKLSE1=6.4*B2*(1./6.-B2/10.+B3/30.)
     ZKLSE2=0.64-0.8134*B
     BARS1=B*(1./12.-B2/15.+B3/42.)/(1./6.-B2/10.+B3/30.)
     BARS2=(0.127083-0.184524*B)/(0.4-0.508333*B)
     ZKMSE=ZKMSE1+ZKMSE2
     ZKLSE=ZKLSE1+ZKLSE2
     IF(KRAK.EQ.1)GOTO 335
C: CRACK PATTERN A
     CVS=0.5*B
     CVL=0.5*(1.0-B)
     XP=ZLH*B/3.0
     XBARS=BARS1*ZLH
     ZP=ZLV*(1.0-4.0*B/3.0)/(4.0*(1.0-B))
     ZBARS=BARS2*ZLV
     XBARP=0.5*B*ZLH
     ZBARP=ZLV*(1./24.-B/16.)/(1./8.-B/6.)
     GOTO 338
C: CRACK PATTERN B
335  CVS=0.5*(1.0-B)
     CVL=0.5*B
     XP=ZLH*(1.0-4.0*B/3.0)/(4.0*(1.0-B))
     XBARS=BARS2*ZLH
     ZP=ZLV*B/3.0
     ZBARS=BARS1*ZLV
     XBARP=ZLH*(1./24.-B/16.)/(1./8.-B/6.)
     ZBARP=0.5*B*ZLV
338  ZKLMSE=ZKMSE/ZKLSE
     ZKLM=ZKLMSE
     GOTO(390,340,350,360,390,340,470),ICASE
C
C: CASES 2, 3, & 4 -- ELASTIC RANGE
350  IF(KRAK.EQ.1)GOTO 365
```

```
      GØTØ 340
  360 IF(KRAK.EQ.0)GØTØ 365
C: CASES 2A, 2B, 3A, 4B, & 6
  340 ZKMFE1=512.0*B5*(1.0/30.-B/10.+3.*B2/28.-B3/18.+B4/90.)
      ZKLFE1=32.0*B3*(1./12.-B/10.+B2/30.)
      BARF1=B*(.05-B/15.+B2/42.)/(1./12.-B/10.+B2/30.)
      GØTØ(370,365,370,370,370,365),ICASE
C: CASES 2A, 2B, 3B, 4A, & 6
  365 ZKMFE2=0.4065-0.6144*B
      ZKLFE2=0.5344-0.7328*B
      BARF2=(.091667-.138095*B)/(.266 67-.366667*B)
      GØTØ(375,368,375,375,375,368),ICASE
C: CASES 2A & 2B
  368 ZKMFE=ZKMFE1+ZKMFE2
      ZKLFE=ZKLFE1+ZKLFE2
      GØTØ 380
C:  CASES 3A & 4B
  370 ZKMFE=ZKMFE1+ZKMSE2
      ZKLFE=ZKLFE1+ZKLSE2
      GØTØ 380
C: CASES 3B, 4A, & 6
  375 ZKMFE=ZKMSE1+ZKMFE2
      ZKLFE=ZKLSE1+ZKLFE2
  380 ZKLMFE=ZKMFE/ZKLFE
      ZKLM=ZKLMFE
      GØTØ 390
C: CASE 7
  470 ZKLMFE=0.78
C
C: ALL CASES -- PLASTIC RANGE
  390 ZKMP=(1.0-B)/3.0
      ZKLP=0.5-B/3.0
      ZKLMP=ZKMP/ZKLP
C
C
C: DETERMINE DYNAMIC REACTIØN CØF  'CIENTS FØR SHØRT (VS) AND
C: LØNG (VL) EDGES
C
      IF(ICASE.LT.5)GØTØ 395
      XBARS=1E-10$ BARF1=1E-10$ XBARP=1E-10
  395 CØNTINUE
      GØTØ(450,400,400,420,450,400,445),ICASE
  400 IF(KRAK.EQ.1)GØTØ 410
      XBARF=BARF1*ZLH
      IF(ICASE.EQ.3)GØTØ 430
  405 ZBARF=BARF2*ZLV
      GØTØ 440
  410 XBARF=BARF2*ZLH
      IF(ICASE.EQ.3)GØTØ 435
  415 ZBARF=BARF1*ZLV
      GØTØ 440
  420 IF(KRAK.EQ.1)GØTØ 425
      XBARF=BARS1*ZLH
      GØTØ 405
  425 XBARF=BARS2*ZLH
      GØTØ 415
  430 ZBARF=BARS2*ZLV
      [illegible]Ø 440
  4[illegible] [illegible]RF=BARS1*ZLV
  440 CØNTINUE
C
C: CASES 2, 3, 4, & 6 -- ELASTIC RANGE
      VS1F=CVS*(1.0-XP/XBARF)
      VS2F=CVS*(XP/XBARF)
      VL1F=CVL*(1.0-ZP/ZBARF)
      VL2F=CVL*(ZP/ZBARF)
      VS1=VS1F
      VL1=VL1F
      GØTØ 450
C
C: CASE 7 -- ELASTIC RANGE
  445 VS1F=0
      VS1=0
      VL1F=0.459
```

PROGRAM UNREINF (CONTINUED)

```
      VL1=VL1F
      VL2F=0.165
C
C: CASE 1 & 5 -- ELASTIC RANGE
  450 VS1S=CVS*(1.0-XP/XBARS)
      VS2S=CVS*(XP/XBARS)
      VL1S=CVL*(1.0-ZP/ZBARS)
      VL2S=CVL*(ZP/ZBARS)
      GOTO(455,460,460,460,455,460,460),ICASE
  455 VS1=VS1S
      VL1=VL1S
C
C: ALL CASES -- PLASTIC RANGE
  460 VS1P=CVS*(1.0-XP/XBARP)
      VS2P=CVS*(XP/XBARP)
      VL1P=CVL*(1.0-ZP/ZBARP)
      VL2P=CVL*(ZP/ZBARP)
      RETURN
      END
      SUBROUTINE WINDOW(OMULT,ZLV,ZLH,ZLVW,ZLHW,AWIN,AWALL,R,ICASE)
C
C: THIS SUBROUTINE DETERMINES THE STRUCTURAL
C: MODIFICATION FACTOR FOR WALLS WITH WINDOWS
      IF(ICASE.GT.4.AND.ICASE.NE.10)GOTO 320
      RWWS=ZLVW/ZLV
      RWWL=ZLHW/ZLH
      RAREA=AWIN/AWALL
      IF(R.LE.1.5)GOTO 300
      IF(RWWS.GT.0.7)GOTO 300
      IF(RWWL.LT.0.5)GOTO 300
      IF(RWWS.EQ.RWWL)GOTO 300
C
C:  CASE WHERE LV/LH >= 1.5, LVW/LV <= 0.7, AND LHW/LH >= 0.5
C:  (BUT LVW/LV NOT EQUAL TO LHW/LH)
      OMULT=-5.85461-12.6644*RAREA+4.39662*RWWS+0.94943*RWWL
     +      -0.223*R-1.07269*(ZLVW/ZLHW)**0.9+6.59942*EXP(RAREA)
      GOTO 315
C
C:  CASE WHERE ONE OR MORE OF ABOVE CONDITIONS IS NOT MET
  300 OMULT=0.62022-2.23415*RAREA**(RWWL**4)-0.79461*RWWL**2
     +      -2.27663*RWWL+0.62522*RWWL/RAREA**0.3
     +      +2.63043*EXP(RAREA)-0.09268*RWWS
  315 CONTINUE
      RETURN
C
C ONE-WAY ACTION WALLS
  320 OMULT=(AWALL-ZLV*ZLHW)/(AWALL-AWIN)
      RETURN
      END
      SUBROUTINE RANDOM (IENTRY)
C THIS SUBROUTINE INPUTS MEAN AND STANDARD DEVIATIONS FOR RANDOM
C VARIABLES; GENERATES RANDOM VALUES; AND CONTROLS REQUIRED
C NUMBER OF CASES TO BE RUN; AND OUTPUTS FINAL RESULTS AND SUMMARY
C
      COMMON Y(100),YU,YFAIL,Q,QU,AREA,ZMASS,ZKLM,VH1,VH2,VV1,VV2
      COMMON /WALL,KINC,KRE,KRAND,I,ICASE,FU,VFAIL,FR,FRM,EM,FDY
      COMMON FDC,D(4),LDTYPE,PEXT,PF,PSO,PUO,PC,TC,TO,PB,TIME,L,S
      COMMON /RAND/ TIMEC,IWALL
      DIMENSION CHI25(7),CHI975(7),TDIST(7)
C
C VALUES FOR 97.5% (F=19,24,29,34,39,44,49)
      DATA CHI25/.4693,.5167,.5533,.5825,.6065,.6267,.6440/
      DATA CHI975/1.7295,1.6402,1.5766,1.5284,1.4903,1.4591,1.4331/
      DATA TDIST/2.093,2.064,2.045,2.032,2.022,2.016,2.010/
C
      GOTO(5,50,70),IENTRY
    5 XDUMMY=XNORM1(-1.0,0.0,1.0)
C INITIALIZE RANDOM NUMBER GENERATOR
      PRINT,/,*INPUT NRAND*,
      READ,NRAND
      DO 47 I=1,NRAND
      XDUMMY=XNORM1(0.0,0.0,1.0)
   47 CONTINUE
      INDEX=0$  SPSO=0$  SSPSO=0
```

PROGRAM UNREINF (CONTINUED)

```
      ICHECK=20
C
C  INPUT MEAN AND STANDARD DEVIATIØN FØR RANDØM VARIABLES
      PRINT 97
      READ, SMEAN, SSD
      GØTØ(10,20), IWALL
C  UNREINFØRCED WALLS WITHØUT ARCHING
   10 PRINT 84
      READ, FRMEAN, FRSD
      PRINT 94
      RETURN
C  UNREINFØRCED WALLS WITH ARCHING
   20 PRINT 85
      READ, FPMMEAN, FPMSD
      PRINT 95
      RETURN
C
C  GENERATE RANDØM VALUES
   50 GØTØ(52,54), IWALL
   52 FR=XNØRM1(0.0, FRMEAN, FRSD)
      IF(FR.LE.0)GØTØ 52
      GØTØ 58
   54 FPM=XNØRM1(0.0, FPMMEAN, FPMSD)
      IF(FPM.LE.0)GØTØ 54
   55 ALPHA=XNØRM1(0.0,1.0,0.3)
      IF(ALPHA.LT.0.4.ØR.ALPHA.GT.1.6)GØTØ 55
      EM=1000.0*ALPHA*FPM
   58 IF(SMEAN.EQ.0)GØTØ 65
   60 S=XNØRM1(0.0, SMEAN, SSD)
      IF(S.LE.0)GØTØ 60
   65 INDEX=INDEX+1
      RETURN
C SUM VALUES ØF PSØ AND PSØ**2 FØR USE IN STATISTICAL ANALYSIS
   70 SPSØ=SPSØ+PSØ
      SSPSØ=SSPSØ+PSØ*PSØ
C
C ØUTPUT FINAL RESULTS
      GØTØ(72,74), IWALL
   72 PRINT 90, FR, S, PSØ, TIMEC
      GØTØ 80
   74 PRINT 91, FPM, EM, S, PSØ, TIMEC
      GØTØ 80
   80 IF(INDEX.LT.ICHECK)RETURN
C
C DETERMINE MEAN, STANDARD DEVIATIØN, AND STANDARD ERRØR FØR PSØ
      ZNØ=INDEX
      ZMEAN=SPSØ/ZNØ
      SD=SQRT((SSPSØ-ZNØ*ZMEAN*ZMEAN)/ZNØ)
      STDERR=SD/(SQRT(ZNØ-1))
C CHECK IF MAXIMUM ØF 50 PSØ SAMPLES ØBTAINED
      IF(INDEX.EQ.50)GØTØ 62
C CHECK IF 95% CØNFIDENCE INTERVAL FØR MEAN PSØ VALUE IS
      IF(STDERR*TDIST((INDEX-15)/5)/ZMEAN.GT.0.10)GØTØ 61
C
C CØNFIDENCE INTERVAL IS WITHIN 10% -- DETERMINE UPPER LIMIT ØF
C 95% CØNFIDENCE INTERVAL FØR STANDARD DEVIATIØN
C PRØBABILITY VALUE AND ITS 95% CØNFIDENCE INTERVAL UPPER LIMIT
   62 SDU=SD/(SQRT(CHI25((INDEX-15)/5)))
C CHECK IF MAXIMUM ØF 50 PSØ SAMPLES ØBTAINED
      IF(INDEX.EQ.50)GØTØ 53
C CHECK IF UPPER VALUE ØF 95% CØNFIDENCE INTERVAL FØR STANDARD
C DEVIATIØN IS WITHIN 0.10*MEAN ØF THE STANDARD DEVIATIØN
      IF(((SDU-SD)/ZMEAN).GT.0.10)GØTØ 61
C
C 95% CØNFIDENCE INTERVAL IS WITHIN 10% FØR BØTH MEAN AND 90%
C PRØBABILITY VALUE -- THEREFØRE SUFFICIENT SAMPLES ØBTAINED
C DETERMINE 95% CØNFIDENCE INTERVALS FØR MEAN, STANDARD DEVIATIØN
.
C AND 10% AND 90% PRØBABILITY VALUES
   53 ZMEANL=ZMEAN-STDERR*TDIST((INDEX-15)/5)
      ZMEANU=ZMEAN+STDERR*TDIST((INDEX-15)/5)
      SDL=SD/(SQRT(CHI975((INDEX-15)/5)))
      P10=ZMEAN-1.282*SD
      P10L=ZMEAN-1.282*SDU
```

PROGRAM UNREINF (CONTINUED)

```
      P10U=ZMEAN-1.282*SDL
      P90=ZMEAN+1.282*SD
      P90L=ZMEAN+1.282*SDL
      P90=ZMEAN+1.282*SD
      P90U=ZMEAN+1.282*SDU
      P90U=ZMEAN+1.282*SDU
C
C OUTPUT STATISTICAL PARAMETERS OF INCIPIENT COLLAPSE PRESSURE
      PRINT 100,ZMEAN,ZMEANL,ZMEANU,SD,SDL,SDU,P10,P10L,P10U,
+       P90,P90L,P90U
      PRINT 105,INDEX,STDERR
      GOTO 999
C
C 95% CONFIDENCE INTERVAL IS NOT WITHIN 10% FOR BOTH MEAN AND 90
C
C VALUES -- THEREFORE OBTAIN 5 ADDITIONAL SAMPLES
   61 ICHECK=ICHECK+5
      RETURN
C
 34 FORMAT(/*INPUT MEAN AND STANDARD DEVIATION FOR FR*,*)
 85 FORMAT(/*INPUT MEAN AND STANDARD DEVIATION FOR FPM*,*)
  87 FORMAT(/*INPUT MEAN AND STANDARD DEVIATION FOR S*,*)
 90 FORMAT(F9.2,F11.2,F10.2,F14.3)
 91 FORMAT(F9.1,F15.1,F12.2,F10.2,F14.3)
 94 FORMAT(///,5X,*FR*,10X,*S*,8X,*PSO*,6X,*COLLAPSE TIME*)
 95 FORMAT(///,5X,*FPM*,11X,*EM*,12X,*S*,8X,*PSO*,6X,
+    *COLLAPSE TIME*)
 100 FORMAT(///,11X,*STATISTICAL PROPERTIES OF INCIPIENT PSO*,
+        //,34X,*95% CONFIDENCE LIMITS*,/,7X,*ITEM*,18X,
+        *VALUE       LOWER        UPPER*,//,* MEAN*,F29.2,
+        2F12.2,//,* STANDARD DEVIATION*,F15.2,2F12.2,//,
+        * 10% PROBABILITY VALUE*,3F12.2,//,
+        * 90% PROBABILITY VALUE*,3F12.2)
 105 FORMAT(//,5X,*NUMBER OF OBSERVATIONS =*,I3,/,5X,
+        *STANDARD ERROR =*,F5.2)
C
 999 STOP$ END
FUNCTION XNORM1(X,A,B)
IF(X)10,20,20
10 X0=RANF(-1.0)
20 X1=RANF(0.0)
X2=RANF(0.0)
Y=SQRT(-2.0*ALOG(X1))*(COS(6.283184*X2))
XNORM1=A+Y*B
RETURN
END
```

ADDITION TO PROGRAM UNREINF TO INCLUDE LOAD TYPES 2 THROUGH 5:

```
C      2. TRIANGULAR LOAD
C      3. STEP PULSE
C      4. HRS SHOCK TUNNEL LOAD
C      5. ARBITRARY LOAD SHAPE
DIMENSION TT(20),PP(20)
GOTO(1,2,3,4),IENTRY
1 GOTO(100,110,110,120,130),LDTYPE
110 PRINT 610
READ,TR,TO
GOTO 125
120 PRINT 615
READ,TR,T1,TO
125 IF(KINC.EQ.1)RETURN
PRINT 630
READ,PSO
RETURN
130 PRINT 620
READ,NPOINT,(TT(I),PP(I),I=1,NPOINT)
```

PROGRAM UNREINF (CONCLUDED)


```
[illegible]
```

ARCHING

Unreinforced Masonry Wall With Arching

PROGRAM ARCHING

```
      PROGRAM JIMBOC(INPUT,OUTPUT,TAPE1=OUTPUT)
C:  THIS ROUTINE IS THE CONTROLLING ROUTINE FOR THE PROGRAM USED
C:  IN THE ANALYSIS OF ONE-WAY OR TWO-WAY ACTION WALLS.
C
      COMMON Y(100),YU,YFAIL,Q,QU,AREA,ZMASS,ZKLM,VH1,VH2,VV1,VV2
      COMMON KWALL,KINC,KRF,KRAND,I,ICASE,FU,VFAIL,FR,FPM,EM,FDY
      COMMON FDC,D(4),LDTYPE,PEXT,PF,PSO,PDO,PC,TC,TO,PO,TIME,L,S
      COMMON /RAND/  TIMEC,IWALL
      DIMENSION A(100),V(100),T(100),VV(100),VH(100),
     +  PEX(100),PIN(100),PN(100)
C
C:  READ TITLE AND CONTROL PARAMETERS
 5    PRINT 67
      READ 68,TITLE
      PRINT 85
      READ,KWALL,KINC,LDTYPE,KRF,KRAND
      DELAY=0
      VFAIL=1E10
      CALL RESIST(1)
      CALL FORCE(1)
      IF(KRF.EQ.0)GOTO 12
      CALL FILL(PINT,1)
 12   IF(KWALL.EQ.0)GOTO 14
      PRINT 86
      READ,DELAY
      DELAY=DELAY/1000.0
 14   IF(KRAND.NE.1)GOTO 35
      CALL FORCE(4)
      CALL RANDOM(1)
 34   CALL RANDOM(2)
 35   CALL RESIST(3)
C
C: MINIMUM, MAXIMUM, AND STARTING VALUES ARE DETERMINED FOR CASES
C: WHERE THE LOAD CAUSING INCIPIENT COLLAPSE IS TO BE FOUND
 13   IF(KINC.EQ.0)GOTO 23
        PF=QU
        PFMAX=0
        PFMIN=PF/2.0
      GOTO 20
 16     PF=(PFMIN+PFMAX)/2.0
 20   CALL FORCE(2)
 23   IF(KRF.EQ.0)GOTO 24
      CALL FILL(PINT,2)
C
C: INITIALIZE VALUES FOR BETA METHOD (BETA = 1/6) AND COMPUTE VA U
C: FOR FIRST TIME INTERVAL ASSUMING WALL INITIALLY AT REST
 24     I=1
        TIME=0
      T(1)=0$ V(1)=0$ Y(1)=0
        DELTA=0.001
      IF(KWALL.EQ.0)GOTO 29
 27   IF(TIME.GE.(DELAY-0.00001))GOTO 28
      TIME=TIME+DELTA
      CALL FILL(PINT,3)
      GOTO 27
 28   PIN(1)=PINT
      TPNET=AREA*PINT
      T(1)=TIME
      GOTO 30
 29   CALL FORCE(3)
      PEX(I)=PEXT
      TPNET=AREA*PEXT
        PN(1)=PEXT
 30   CALL RESIST(2)
        A(1)=TPNET/(ZMASS+ZKLM)
        VV(1)=VV1*TPNET
        VH(1)=VH1*TPNET
C
C: PROCEDURE FOR ALL SUBSEQUENT TIME INTERVALS
 1      I=I+1
      IF(I.LT.101)GOTO 11
      PRINT 98,TIME
  98  FORMAT(/,*I=101$  TIME =*,F6.3,*$  WALL ASSUMED TO NOT FAIL*)
      GOTO 6
```

PROGRAM ARCHING (CONTINUED)

```
11    TIME=TIME+DELTA
       T(I)=TIME
       A(I)=A(I-1)
      CALL FORCE(3)
       PEX(I)=PEXT
      IF(KWALL.EQ.1)GOTO 10
      IF(KRF.EQ.0)GOTO 3
      CALL FILL(PINT,3)
       PIN(I)=PINT
      TPNET=AREA*(PEXT-PINT)
       GOTO 2
3      TPNET=AREA*PEXT
      GOTO 2
10    CALL FILL(PINT,3)
      PIN(I)=PINT
      TPNET=AREA*PINT
2     PN(I)=TPNET/AREA
      DO 8 JJ=1,10
       Y(I)=Y(I-1)+DELTA*V(I-1)+DELTA*DELTA*(A(I-1)/3.+A(I)/6.)
       CALL RESIST(2)
       QT=Q*AREA
4      ANEW=(TPNET-QT)/(ZMASS*ZKLM)
       ADELTA=ANEW-A(I)
       A(I)=ANEW
 IF(ANEW.EQ.0)PRINT,*1985*,TIME,TPNET,QT,ZMASS,ZKLM,Y(I),A(I-1)
       IF(ABS(ADELTA/ANEW).LT.0.01)GOTO 9
8      CONTINUE
       A(I)=ANEW-ADELTA/2.0
       WRITE(1,80)TIME,PF,A(I),Y(I)
9      CONTINUE
       Y(I)=Y(I-1)+DELTA*V(I-1)+DELTA*DELTA*(A(I-1)/3.+A(I)/6.)
       V(I)=V(I-1)+DELTA*(A(I)+A(I-1))/2.0
       VV(I)=VV1*TPNET+VV2*QT
       VH(I)=VH1*TPNET+VH2*QT
      IF(VV(I).GT.VFAIL)GOTO 7
C
C: CHECK FOR MAXIMUM DEFLECTION OR FAILURE OF WALL
C:   IF MAXIMUM DEFLECTION REACHED, WALL DID NOT FAIL
      IF(Y(I).LE.Y(I-1).AND.PN(I).LE.PN(I-1))COTO 6
      IF(Y(I).LT.0)GOTO 6
      IF(TIME-DELAY.GE.0.010)DELTA=0.002
      IF(Y(I).LT.YU)GOTO 1
      IF(TIME-DELAY.GE.0.020)DELTA=0.005
      IF(TIME-DELAY.GE.J.100)DELTA=0.[illegible]
      IF(TIME-DELAY.GE.0.500)DELTA=0.[illegible]
C   IF FAILURE DEFLECTION REACHED, WALL FAILED
      IF(Y(I:.GE.YFAIL)GOTO 7
       GOTO 1
C
C: INTERVAL HALVING PROCEDURE TO DETERMINE LOAD CAUSING INCIPIENT
C: COLLAPSE FOR CASES WHERE DESIRED
C:   WALL DID NOT FAIL - SET PFMIN TO PF
6      CONTINUE
       IF(KRAND.EQ.1)GOTO 3b
       IF(KINC.EQ.0)GOTO 18
36     PFMIN=PF
        IF(PFMAX.G[illegible].0)COTO 17
        PF=2.0*PF
        GOTO 20
C:   WALL FAILED - SET PFMAX TO PF
7      CONTINUE
       TIMEC=TIME
       IF(KRAND.EQ.1)GOTO 37
       IF(KINC.EQ.0)GOTO 18
37     PFMAX=PF
C:   CHECK TO SEE IF LOAD RANGE IS WITHIN DESIRED ACCURACY
17      IF((PFMAX-PFMIN)/PFMIN.GT.0.01)GOTO 16
       IF(KRAND.NE.1)GOTO 18
       CALL RANDOM(3)
       GOTO 34
C
C: OUTPUT DATA INCLUDES THE MAXIMUM DEFLECTION AND TIME OF
C: OCCURANCE FOR A NON-FAILING WALL OR THE TIME AND VELOCITY
C: AT COLLAPSE FOR A FAILING WALL.  OPTIONAL OUTPUT IS THE
```

```
C: ENTIRE BEHAVIOR TIME-HISTORY OF THE WALL.
C
C: OUTPUT LOAD DATA
 18     CALL FORCE(4)
C
C: OUTPUT FINAL RESULTS
        IF(Y(I).LT.YFAIL)WRITE(1,70)Y(I),T(I)
        IF(Y(I).GE.YFAIL)WRITE(1,71)T(I),V(I)
C
C: CHECK TO SEE IF ENTIRE TIME-HISTORY OF WALL IS DESIRED
        WRITE(1,72)
 READ,M
        IF(M.EQ.0)GOTO 25
       IF(KWALL.EQ.1)GOTO 32
       IF(KRF.EQ.0)GOTO 26
        WRITE(1,75)(T(J),PEX(J),PIN(J),PN(J),Y(J),VV(J),VH(J),J=1,I)
        GOTO 25
 26     WRITE(1,76)(T(J),PEX(J),A(J),V(J),Y(J),VV(J),VH(J),J=1,I)
       GOTO 25
 32    WRITE(1,76)(T(J),PIN(J),A(J),V(J),Y(J),VV(J),VH(J),J=1,I)
 25     WRITE(1,77)
        GOTO 5
C
 67 FORMAT(/*INPUT TITLE*,+)
 68 FORMAT(A59)
 70 FORMAT(/*WALL DID NOT FAIL - MAX. DEFLECTION OF*F6.2
+     * IN. REACHED AT*F7.3,* SEC*)
 71 FORMAT(/*WALL FAILED AT*,F7.3,* SEC (FINAL VELOCITY =*
+     F7.2* IN./SEC)*)
 72 FORMAT(/*IS TIME HISTORY OF WALL DESIRED (YES=1, NO=0)*,+)
 75 FORMAT(/,15X,*PRESSURE ON WALL*/*  TIME   EXTERIOR  *
+ *INTERIOR     NET     DISPLACEMENT     VV       VH*/
+     (F6.3,3F10.3,F12.4,F11.0,F8.0))
 76 FORMAT(/*  TIME  PRESSURE  ACCELERATION  VELOCITY  *
+     *DISPLACEMENT     VV       VH*/
+     (F6.3,F9.3,F12.1,F12.2,F12.4,F11.0,F8.0))
 77     FORMAT(///,7(*----------*))
 80 FORMAT(/*ACCELERATION NOT CONVERGING AT TIME =*,F6.3,
+             * SEC (PF =*,F7.3,* PSI)*/*      A(I) SET EQUAL TO
+             F8.1,* (AVG OF LAST 2 ITERATIONS)*/*      Y(I) =*,
+             F8.4,* IN.*)
 85 FORMAT(/*INPUT KWALL(0=EXT,1=INT),KINC,LDTYPE,KRF,KRAND*,
+     *(1=RANDOM)*)
 86 FORMAT(/*INPUT DELAY TIME (MSEC) TO INITIAL LOADING OF*,
+     * INTERIOR WALL*,+)
C
 999    STOP
        END
 SUBROUTINE FORCE(IENTRY)
C  THIS SUBROUTINE INPUTS THE LOAD PARAMETERS AND DETERMINES
C  THE LOAD AT A GIVEN TIME FOR THE FOLLOWING LOAD TYPES:
C      1. IDEALIZED BLAST LOAD (FRONT OR SIDE FACE)
C
 COMMON Y(100),YU,YFAIL,Q,QU,AREA,ZMASS,ZKLM,VH1,VH2,VV1,VV2
 COMMON KWALL,KINC,KRF,KRAND,I,ICASE,FU,VFAIL,FR,FPM,EM,FDY
 COMMON FDC,D(4),LDTYPE,P,PR,PSO,PDO,PC,TC,TO,PO,TIME,LL,S
C
 GOTO(100,200,300,4),IENTRY
C
C  INPUT LOAD PARAMETERS
 100 IF(KRAND.EQ.0)GOTO 102
 W=1000 $ PO=14.7 $ CO=1120.   LOC=1
 RETURN
 102 PRINT 600
 READ,W,PO,CO,LOC,S
 IF(LOC.EQ.1)GOTO 105
 PRINT 605
 READ,ZLEN,CD
 105 IF(KINC.EQ.1)RETURN
 PRINT 630
 READ,PSO
 PR=2.0*PSO*(7.0*PO+4.0*PSO)/(7.0*PO+PSO)
 GOTO 215
C
```

PROGRAM ARCHING (CONTINUED)

```
11010C  CALCULATE LØAD PRØPERTIES FØR GIVEN PEAK PRESSURE
11030 200 GØTØ(205,210),LØC
11040 205 PSØ=(PR-14.0*PØ+SQRT(196.0*PØ*PØ+196.0*PØ*PR+PR*PR))/16.0
11050 GØTØ 215
11060 210 PSØ=PR
11070 215 PDØ=2.5*PSØ*PSØ/(7.0*PØ+PSØ)
11080 U=CØ*SQRT(1.0+(6.0*PSØ)/(7.0*PØ))
11090 TO=W**0.3333/(2.2399+0.1886*PSØ)
11100 GØTØ(220,225),LØC
11110 220 TC=3.0*S/U
11120 PC=PSØ*(1-TC/TO)*EXP(-TC/TO)+PDØ*(1-TC/TO)**2*EXP(-2*TC/TO)
11130 CD=1.0
11140 RETURN
11150 225 TA=ZLEN/U
11160 TA2=TA/2.0
11170 TA2TO=TA2/TO
11180 PA=PSØ*(1-TA2TO)*EXP(-TA2TO)+CD*PDØ*(1-TA2TO)**2*EXP(-2*TA2TO)
11190 RETURN
12000C
12010C  CALCULATE LØAD
12030 300 GØTØ(305,310),LØC
12040 305 TTO=TIME/TO
12050 IF(TIME.GT.TC)GØTØ 320
12060 P=PC+(TC-TIME)*(PR-PC)/TC
12070 RETURN
12080 310 TTO=(TIME-TA2)/TO
12090 IF(TIME.GT.TA)GØTØ 320
12100 P=PA*TIME/TA
12110 RETURN
12120 320 IF(TTO.GE.1.0)GØTØ 330
12130 P=PSØ*(1-TTO)*EXP(-TTO)+CD*PDØ*(1-TTO)**2*EXP(-2*TTO)
12150 RETURN
12160 330 P=0
12170 RETURN
13000C
13010C  PRINT LØAD DATA
13020 4 IF(KINC.EQ.0)GØTØ 400
13030 PRINT 640,LDTYPE
13040 GØTØ 410
13050 400 PRINT 645,LDTYPE
13060 410 CØNTINUE
13070 415 GØTØ(420,425),LØC
13080 420 PRINT 650
13090 GØTØ 430
13100 425 PRINT 655
13110 430 PRINT 660,W,PØ,CØ
13120 IF(KRAND.NE.0)RETURN
13130 GØTØ(435,440),LØC
13140 435 PRINT 665,S,TC,PR
13150 GØTØ 445
13160 440 PRINT 670,ZLEN,TA,PA
13170 445 PRINT 675,U,TO,CD,PSØ,PDØ
13180 RETURN
14000C
14010 600 FØRMAT(/*INPUT W,PØ,CØ,LØC,S*,†)
14020 605 FØRMAT(/*INPUT L,CD*,†)
14060 630 FØRMAT(/*INPUT PSØ*,†)
14070 640 FØRMAT(/*LØAD CAUSING INCIPIENT FAILURE IS AS FØLLØWS:*,
14071*      /,5X,*LØAD TYPE NUMBER*,I2)
14080 645 FØRMAT(/*PRØPERTIES ØF LØAD ACTING ØN WALL ARE AS FØLLØWS:*
14081*      /,5X,*LØAD TYPE NUMBER*,I2)
14090 650 FØRMAT(8X,*(FRØNT FACE)*)
14100 655 FØRMAT(8X,*(SIDE FACE)*)
14110 660 FØRMAT(10X,*W =*,F8.1,* KT   PØ =*,F6.2,* PSI      CØ =*,
14111*      F7.1,* FPS*)
14120 665 FØRMAT(10X,*S =*,F6.1,* FT      TC =*,F6.3,* SEC      PR =*,
14121*      F7.3,* PSI*)
14130 670 FØRMAT(10X,*L =*,F6.1,* FT      TA =*,F6.3,* SEC      PA =*,
14131*      F7.3,* PSI*)
14140 675 FØRMAT(10X,*U =*,F7.1,* FPS   TO =*,F6.3,* SEC     C  =*
14141*      F5.1,/,8X,*PSØ =*,F7.3,* PSI  PDØ =*,F7.3,* PSI*)
15000 END
20000 SUBRØUTINE FILL(P3,IENTRY)
20010C: CØMPUTES AVERAGE AIR PRESSURE IN RØØM DUE TØ BLAST WAVE
```

PROGRAM ARCHING (CONTINUED)

```
C: INCIDENT HEAD-ØN UPØN FRØNT WALL.
C
 CØMMØN Y(100),YU,YFAIL,Q,QU,AREA,ZMASS,ZKLM,VH1,V.2,VV1,VV2
 CØMMØN KWALL,KINC,KRF,KRAND,II,ICASE,FU,VFAIL,FR,FPM,EM,FDY
 CØMMØN FDC,D(4),LDTYPE,PEXT,PR,PSØ,PDØ,PC,TC,TO,PØ,TIME,L,S
 DIMENSIØN AA(8,2),NN(8)
 LØGICAL L1,L2,L3
 GØTØ(10,13,11),IENTRY
 10 PRINT 700
 RHØØ=0.076 $ L1=.FALSE.
 READ,NWIN,V3
 AT=0$ AFRØNT=0$ ASIDE=0
 DØ 18 I=1,NWIN
 PRINT 710,I
 READ,AA(I,1),NN(I),AA(I,2)
 AA(I,2)=AA(I,2)/1000.0
 AT=AT+AA(I,1)
 M=NN(I)$ GØTØ(12,14,14),M
 12 AFRØNT=AFRØNT+AA(I,1)
 GØTØ 18
 14 ASIDE=ASIDE+AA(I,1)
 18 CØNTINUE
 AFRØNT=AFRØNT/AT$ ASIDE=ASIDE/AT
 700 FØRMAT(/*INPUT NWIN AND RØØM VØLUME (CF)*,*)
 710 FØRMAT(/*INPUT AREA (SQ FT),LØCATIØN CØDE & DELAY(MSEC)*
+ * FØR WINDØW*,I2,*)
 G=1.4 $ G2=1./G $ G3=1.-G2 $ G4=2./G3 $ G5=G+1.
 G6=2.*G/G5 $ G7=(G-1.)/G5
 PP2=.1912
 C=SQRT(G*PØ*32.*144./RHØØ)
 TAU=2.*(V3**(1./3.))/C
 NSTEP=4
 DT=TAU/NSTEP
 RETURN
C
 13 P3Ø=PØ
 TT=0.$ TØ=0.
 RHØ3Ø=RHØØ
 L2=.FALSE. $ L3=.FALSE.
 RETURN
C
 11 IF(L1)GØTØ 52
 IF(L2.A.L3)GØTØ 9
 52 DDT=(TIME-T?)*0.5
 ISTØP=2
 53 IF(DDT.LT.DT)?ØTØ 51
 50 DDT=0.5*DDT
 ISTØP=2*ISTØP
 GØ TØ 53
 51 CØNTINUE
 DØ 99 I=1,ISTØP
 TT=TØ+I*DDT
 IF(TT.GT.TO)GØ TØ 99
 DM=0. $ WW=0. $ NW=0
 DØ 500 K=1,NWIN
 M=NN(K) $ DLY=AA(K,2)+0.000001
 IF(DLY.GE.TT)GØ TØ 500
 GØTØ(15,16,16),M
 15 CDF=1.0
 IF(TT-TC,20,20,21
 20  P11=(TC-TT)*(PR-PC)/TC+PC
 P11=P11+PØ
 GØ TØ 30
 16 CDF=-0.4
 21  R=TT/TO $ RR=1.-R
 PD=PDØ*RR*RR*EXP(-2.*R)
 PS=PSØ*RR*EXP(-R)
 P11=PS+CDF*PD
 P11=P11+PØ
 30  RHØ1=RHØØ*((P11/PØ)**G2)
 IF(P11-P3Ø)36,36,37
 36  JSIGN=-1
 L2=.TRUE.
 303  P2=P11
```

```
RHØ2=((P2/P3Ø)**G2)*RHØ3Ø
X=P3Ø/RHØ3Ø
GØ TØ 38
37  JSIGN=-1
306 P2=R2*P11
RHØ2=((P2/P11)**G2)*RHØ1
X=P11/RHØ1
38  U22=G4*(X-P2/RHØ2)*32.*144.
IF(U22)40,39,39
40  PRINT,*U22 NEGATIVE*,U22
STØP
39  U2=SQRT(U22)*JSIGN
DDM=U2*RHØ2*AA(K,I)*DDT
DM=DM+DDM
WW=WW+P11*DDM/(G1*RHØ1)
500 CØNTINUE
P3Ø=P3Ø+(G-1.)*WW/V3
RHØ3Ø=RHØ3Ø+DM/V3
99  CØNTINUE
TØ=TT
P3=P3Ø-PØ
IF(TIME.GE.TC)L3=.TRUE.
RETURN
9 R=TIME/TO $ RR=1.0-R
PD=PDØ*RR*RR*EXP(-2.0*R)
PS=PSØ*RR*EXP(-R)
P3=PS+PD*(AFRØNT-0.4*ASIDE)
999 RETURN
END
SUBRØUTINE RESIST (IENTRY)
C
C THIS SUBRØUTINE INPUTS THE REQUIRED DATA AND CØMPUTES VARIØUS
C QUANTITIES USED IN THE ANALYSIS ØF WALLS WITH ØNE-WAY ARCHING
C (SUPPØRT CASE 9) ØR TWØ-WAY ARCHING (SUPPØRT CASE 10)
C
      CØMMØN Y(100),YU,YFAIL,Q,QU,AREA,ZMASS,ZKLM,VH1,VH2,VV1,VV2
      CØMMØN KWALL,KINC,KSF,KRAND,I,ICASE,FU,VFAIL,FR,FPM,EM,FDY
      CØMMØN FDC,D(4),LDTYPE,PEXT,PF,PSØ,PDØ,PC,TC,TO,PØ,TIME,L,S
      CØMMØN /RAND/ TIMEC,IWALL
C
      GØTØ(1,99,2),IENTRY
C
C INPUT WALL DATA
    1 PRINT 60
      READ,ZLV,ZLH,TW,TFLG,EM,FPM,ICASE,ZLVW,ZLHW,GAMMA
C
C DETERMINE VALUES ØF VARIØUS CØNSTANTS
      IWALL=2
      ZL2=ZLV/2.0
      AWALL=ZLV*ZLH
      AWIN=ZLVW*ZLHW
      AREA=AWALL-AWIN
      ZMASS=GAMMA*AREA*TW/(386.07*1728.0)
      RATIØ=ZLV/ZLH
      IF(ICASE.EQ.9)RATIØ=0
     BETA=0.5*(SQRT(3.0*RATIØ**2+RATIØ**4)-RATIØ**2)
     ZKLM=(2.0-2.0*BETA)/(3.0-2.0*BETA)
     VV1=BETA/6.0
     VV2=BETA/3.0
     VH2=(3.0-4.0*BETA)**2/(12.0*(2.0-3.0*BETA))
     VH1=(1.0-BETA)/2.0-VH2
     ZLD=SQRT(ZL2*ZL2+TW*TW)
     EPS=(ZLD-ZL2)/ZLD
     C2=2.0-4.0*BETA
     C3=0
     IF(RATIØ.EQ.0)GØTØ 9
     C3=4.0*BETA+RATIØ**2/BETA
    9 CØNTINUE
     C4=12.0/(ZLV*ZLV*(3.0-2.0*BETA))
     YFAIL=TW
C
C  DETERMINE MØDIFICATIØN FACTØR FØR WALL WITH WINDØWS
C
      QMULT=1.0
```

```
      IF(AWIN.NE.0)CALL WINDOW (QMULT,ZLV,ZLH,ZLVW,ZLHW,AWIN,
+        AWALL,RATIO,ICASE)
C
C  OUTPUT WALL PROPERTIES
      PRINT 62,ICASE,ZLV,ZLH,RATIO,TW,EM,FPM,GAMMA,TFLG,ZLVW,
+        ZLHW,QMULT
      RETURN
C
   2  YU=TW*FPM/(EM*EPSM)
      TFLG2=2.0*TFLG
      IF(TFLG.NE.0)GOTO 3
C
C SOLID MASONRY WALL
    5 ZMU=0.25*FPM*(TW-YU)**2
      C1=0.5*ZMU*YU+FPM*(TW-YU)**3/12.0
      GOTO 4
C
C HOLLOW MASONRY UNIT WALL
    3 Y2=TW-2.0*TFLG
      IF(YU.LT.Y2)GOTO 6
C YU>Y2 -- TREAT WALL AS SOLID WALL
      TFLG2=0$ GOTO 5
    6 PFLG=FPM*TFLG
      TWF=TW-TFLG
      ZMU=PFLG*(TWF-YU)
      ZM2=PFLG*TFLG
      C6=0.5*(ZMU*YU+PFLG*(TWF-YU)**2)
      C7=C6+FPM*TFLG**3/6.0
    4 QU=8.0*ZMU/(ZLV*ZLV)
      RETURN
C
   60 FORMAT(//*INPUT LV,LH,TW,TFLG,EM,FPM,ICASE,LVW,LHW,GAMMA*)
   62 FORMAT(//*PROPERTIES OF UNREINFORCED MASONRY WALL (ARCHING)
+       * -- SUPPORT TYPE NO.*,I3,
+       /,6X,*LV =*,F6.1,* INCHES*,6X,*LH =*,F6.1,
+       * INCHES      RATIO =*,F6.3,/6X,*TW =*,F5.1,* INCHES*,
+       7X,*EM =*,F10.1,* PSI    FPM =*,F8.2,* PSI*,/,3X,
+       *GAMMA =*,F6.1,* PCF*,7X,*TFLG =*,F6.3,* INCHES*,/,
+       5X,*LVW =*,F6.1,* INCHES*,5X,*LHW =*,F6.1,* INCHES*,
+       5X,*QMULT =*,F6.3)
C
C  DETERMINE THE RESISTANCE FOR THE WALL AT A DEFLECTION OF Y(I)
   99 IF(Y(I).GT.YU)GOTO 100
C
C DEFLECTION OF WALL LESS THAN YU
      ZMC=Y(I)*ZMU/YU
      ZMAVG=0.5*ZMC
      GOTO 130
C
C DEFLECTION OF WALL GREATER THAN YU
  100 IF(TFLG2.NE.0)GOTO 110
      ZMC=0.25*FPM*(TW-Y(I))**2
      ZMAVG=(C1-FPM*(TW-Y(I))**3/12.0)/Y(I)
      GOTO 130
C
C HOLLOW MASONRY UNIT WALL -- DEFLECTION LESS THAN Y2
  110 IF(Y(I).GT.Y2)GOTO 120
      ZMC=PFLG*(TWF-Y(I))
      ZMAVG=(C6-0.5*PFLG*(TWF-Y(I))**2)/Y(I)
      GOTO 130
C
C HOLLOW MASONRY UNIT WALL -- DEFLECTION GREATER THAN Y2
  120 ZMC=0.25*FPM*(TW-Y(I))**2
      ZMAVG=(C7-(TW-Y(I))**2/12.0)/Y(I)
      GOTO 130
C
C COMPUTE TOTAL RESISTANCE OF WALL
  130 Q=C4*(C2*ZMC+C3*ZMAVG)*QMULT
C     IF DEFLECTION IS GREATER THAN WALL THICKNESS, RESISTANCE = 0
      IF(Y(I).GT.TW)Q=1E-10
      RETURN
      END
      SUBROUTINE WINDOW(QMULT,ZLV,ZLH,ZLVW,ZLHW,AWIN,AWALL,R,ICASE)
C
```

PROGRAM ARCHING (CONTINUED)

```
C: THIS SUBROUTINE DETERMINES THE STRUCTURAL
C: MODIFICATION FACTOR FOR WALLS WITH WINDOWS
IF(ICASE.GT.4.AND.ICASE.NE.10)GOTO 320
RWWS=ZLVW/ZLV
RWWL=ZLHW/ZLH
RAREA=AWIN/AWALL
IF(R.LE.1.5)GOTO 300
IF(RWWS.GT.0.7)GOTO 300
IF(RWWL.LT.0.5)GOTO 300
IF(RWWS.EQ.RWWL)GOTO 300
C
C:  CASE WHERE LV/LH >= 1.5, LVW/LV <= 0.7, AND LHW/LH >= 0.5
C:  (BUT LVW/LV NOT EQUAL TO LHW/LH)
QMULT=-5.85461-12.6644*RAREA+4.39662*RWWS+0.84843*RWWL
+     -0.223*R-1.07269*(ZLVW/ZLHW)**0.9+6.59942*EXP(RAREA)
GOTO 315
C
C:  CASE WHERE ONE OR MORE OF ABOVE CONDITIONS IS NOT MET
300 QMULT=0.62022-2.23415*RAREA**(RWWL**4)-0.79461*RWWL**2
+     -2.27663*RWWL+0.62522*RWWL/RAREA**0.3
+     +2.63043*EXP(RAREA)-0.09268*RWWS
315 CONTINUE
RETURN
C
C ONE-WAY ACTION WALLS
320 QMULT=(AWALL-ZLV*ZLHW)/(AWALL-AWIN)
RETURN
END
      SUBROUTINE RANDOM (IENTRY)
C THIS SUBROUTINE INPUTS MEAN AND STANDARD DEVIATIONS FOR RANDOM
C VARIABLES;  GENERATES RANDOM VALUES;  AND CONTROLS REQUIRED
C NUMBER OF CASES TO BE RUN;  AND OUTPUTS FINAL RESULTS AND SUMMARY
C
      COMMON Y(100),YU,YFAIL,Q,QU,AREA,ZMASS,ZKLM,VH1,VH2,VV1,VV2
      COMMON KWALL,KINC,KRF,KRAND,I,ICASE,FU,VFAIL,FR,FPM,EM,FDY
      COMMON FDC,D(4),LDTYPE,PEXT,PF,PSO,PDO,PC,TC,TO,PO,TIME,L,S
      COMMON /RAND/ TIMEC,IWALL
      DIMENSION CHI25(7),CHI975(7),TDIST(7)
C
C VALUES FOR 97.5%  (F=19,24,29,34,39,44,49)
      DATA CHI25/.4688,.5167,.5533,.5825,.6065,.6267,.6440/
      DATA CHI975/1.7295,1.6402,1.5766,1.5284,1.4903,1.4591,1.4331/
      DATA TDIST/2.093,2.064,2.045,2.032,2.022,2.016,2.010/
C
      GOTO(5,50,70),IENTRY
    5 XDUMMY=XNORM1(-1.0,0.0,1.0)
C INITIALIZE RANDOM NUMBER GENERATOR
      PRINT,/,*INPUT NRAND*,
      READ,NRAND
      DO 47 I=1,NRAND
      XDUMMY=XNORM1(0.0,0.0,1.0)
   47 CONTINUE
      INDEX=0$  SPSO=0$  SSPSO=0
      ICHECK=20
C
C  INPUT MEAN AND STANDARD DEVIATION FOR RANDOM VARIABLES
      PRINT 87
      READ,SMEAN,SSD
      GOTO(10,20),IWALL
C  UNREINFORCED WALLS WITHOUT ARCHING
   10 PRINT 84
      READ,FRMEAN,FRSD
      PRINT 94
      RETURN
C  UNREINFORCED WALLS WITH ARCHING
   20 PRINT 85
      READ,FPMMEAN,FPMSD
      PRINT 95
      RETURN
C
C  GENERATE RANDOM VALUES
   50 GOTO(52,54),IWALL
   52 FR=XNORM1(0.0,FRMEAN,FRSD)
      IF(FR.LE.0)GOTO 52
```

```
      GOTO 58
 54   FPM=XNORM1(0.0,FPMMEAN,FPMSD)
      IF(FPM.LE.0)GOTO 54
 55   ALPHA=XNORM1(0.0,1.0,0.3)
      IF(ALPHA.LT.0.4.OR.ALPHA.GT.1.6)GOTO 55
      EM=1000.0*ALPHA*FPM
 58   IF(SMEAN.EQ.0)GOTO 65
 60   S=XNORM1(0.0,SMEAN,SSD)
      IF(S.LE.0)GOTO 60
 65   INDEX=INDEX+1
      RETURN
C SUM VALUES OF PS0 AND PS0**2 FOR USE IN STATISTICAL ANALYSIS
 70   SPS0=SPS0+PS0
      SSPS0=SSPS0+PS0*PS0
C
C OUTPUT FINAL RESULTS
      GOTO(72,74),IWALL
 72   PRINT 90,FR,S,PS0,TIMEC
      GOTO 80
 74   PRINT 91,FPM,EM,S,PS0,TIMEC
      GOTO 80
 80   IF(INDEX.LT.ICHECK)RETURN
C
C DETERMINE MEAN, STANDARD DEVIATION, AND STANDARD ERROR FOR PS0
      ZN0=INDEX
      ZMEAN=SPS0/ZN0
      SD=SQRT((SSPS0-ZN0*ZMEAN*ZMEAN)/ZN0)
      STDERR=SD/(SQRT(ZN0-1))
C CHECK IF MAXIMUM OF 50 PS0 SAMPLES OBTAINED
      IF(INDEX.EQ.50)GOTO 62
C CHECK IF 95% CONFIDENCE INTERVAL FOR MEAN PS0 VALUE IS
      IF(STDERR*TDIST((INDEX-15)/5)/ZMEAN.GT.0.10)GOTO 61
C
C CONFIDENCE INTERVAL IS WITHIN 10% -- DETERMINE UPPER LIMIT OF
C 95% CONFIDENCE INTERVAL FOR STANDARD DEVIATION
C PROBABILITY VALUE AND ITS 95% CONFIDENCE INTERVAL UPPER LIMIT
  62 SDU=SD/(SQRT(CHI25((INDEX-15)/5)))
C CHECK IF MAXIMUM OF 50 PS0 SAMPLES OBTAINED
      IF(INDEX.EQ.50)GOTO 53
C CHECK IF UPPER VALUE OF 95% CONFIDENCE INTERVAL FOR STANDARD
C DEVIATION IS WITHIN 0.10*MEAN OF THE STANDARD DEVIATION
      IF(((SDU-SD)/ZMEAN).GT.0.10)GOTO 61
C
C 95% CONFIDENCE INTERVAL IS WITHIN 10% FOR BOTH MEAN AND 90%
C PROBABILITY VALUE -- THEREFORE SUFFICIENT SAMPLES OBTAINED
C DETERMINE 95% CONFIDENCE INTERVALS FOR MEAN, STANDARD DEVIATION
,
C AND 10% AND 90% PROBABILITY VALUES
  53 ZMEANL=ZMEAN-STDERR*TDIST((INDEX-15)/5)
      ZMEANU=ZMEAN+STDERR*TDIST((INDEX-15)/5)
      SDL=SD/(SQRT(CHI975((INDEX-15)/5)))
      P10=ZMEAN-1.282*SD
      P10L=ZMEAN-1.282*SDU
      P10U=ZMEAN-1.282*SDL
      P90=ZMEAN+1.282*SD
      P90L=ZMEAN+1.282*SDL
      P90=ZMEAN+1.282*SD
      P90U=ZMEAN+1.282*SDU
      P90U=ZMEAN+1.282*SDU
C
C OUTPUT STATISTICAL PARAMETERS OF INCIPIENT COLLAPSE PRESSURE
      PRINT 100,ZMEAN,ZMEANL,ZMEANU,SD,SDL,SDU,P10,P10L,P10U,
+       P90,P90L,P90U
      PRINT 105,INDEX,STDERR
      GOTO 999
C
C 95% CONFIDENCE INTERVAL IS NOT WITHIN 10% FOR BOTH MEAN AND 90
B
C VALUES -- THEREFORE OBTAIN 5 ADDITIONAL SAMPLES
  61 ICHECK=ICHECK+5
      RETURN
C
  84 FORMAT(/*INPUT MEAN AND STANDARD DEVIATION FOR FR*,*)
  85 FORMAT(/*INPUT MEAN AND STANDARD DEVIATION FOR FPM*,*)
```

PROGRAM ARCHING (CONTINUED)

```
71280   87 FØRMAT(/*INPUT MEAN AND STANDARD DEVIATIØN FØR S*,+)
71290  90 FØRMAT(F9.2,F11.2,F10.2,F14.3)
71300  91 FØRMAT(F9.1,F15.1,F12.2,F10.2,F14.3)
71320  94 FØRMAT(///,5X,*FR*,10X,*S*,8X,*PS0*,6X,*CØLLAPSE TIME*)
71330  95 FØRMAT(///,5X,*FPM*,11X,*EM*,12X,*S*,8X,*PS0*,6X,
71340+    *CØLLAPSE TIME*)
71360  100 FØRMAT(///,11X,*STATISTICAL PRØPERTIES ØF INCIPIENT PS0*,
71370+        //,39X,*95% CØNFIDENCE LIMITS*, /,7X,*ITEM*,18X,
71380+        *VALUE         LØWER          UPPER*, //,* MEAN*,F29.2,
71390+        2F12.2,//,* STANDARD DEVIATIØN*,F15.2,2F12.2, //,
71400+        * 10% PRØBABILITY VALUE*,3F12.2,//,
71410+        * 90% PRØBABILITY VALUE*,3F12.2)
71420  105 FØRMAT(//,5X,*NUMBER ØF ØBSERVATIØNS =*,I3, /,5X,
71430+        *STANDARD ERRØR =*,F5.2)
71440C
71450  999 STØP$ END
71460 FUNCTIØN XNØRM1(X,A,B)
71470 IF(X)10,20,20
71480 10 X0=RANF(-1.0)
71490 20 X1=RANF(0.0)
71500 X2=RANF(0.0)
71510 Y=SQRT(-2.0*ALØG(X1))*(CØS(6.283184*X2))
71520 XNØRM1=A+Y*B
71530 RETURN
71540 END
```

ADDITIØN TØ PRØGRAM ARCHING TØ INCLUDE LØAD TYPES 2 THRØUGH 5:

```
10040C      2. TRINGULAR LØAD
10050C      3. STEP PULSE
10060C      4. URS SHØCK TUNNEL LØAD
10070C      5. ARBITRARY LØAD SHAPE
10120 DIMENSIØN TT(20),PP(20)
10150 GØTØ(1,2,3,4),IENTRY
10180 1 GØTØ(100,110,110,120,130),LDTYPE
10290 110 PRINT 610
10300 READ,TR,TO
10310 GØTØ 125
10320 120 PRINT 615
10330 READ,TR,T1,TO
10340 125 IF(KINC.EQ.1)RETURN
10350 PRINT 630
10360 READ,PS0
10370 RETURN
10380 130 PRINT 620
10390 READ,NPØINT,(TT(J),PP(J),J=1,NPØINT)
10400 FACTØR=1.0
10410 IF(KINC.EQ.0)GØTØ 150
10420 PMAX=PP(1)
10430 DØ 140 J=2,NPØINT
10440 140 IF(PP(J).GT.PMAX)PMAX=PP(J)
10450 150 PX=PP(2)-PP(1)
10460 TX=TT(2)-TT(1)
10465 JJ=1
10470 RETURN
11020 2 GØTØ(200,230,230,230,240),LDTYPE
11200 230 PS0=PR
11210 RETURN
11220 240 FACTØR=PR/PMAX
11225 GØTØ 150
11230 RETURN
12020 3 GØTØ(300,340,360,370,380),LDTYPE
12180 340 IF(TIME-TR)342,345,345
12190 342 P=PS0*TIME/TR
12200 RETURN
12210 345 IF(TIME-TO)347,350,350
12220 347 P=PS0*(TO-TIME)/(TO-TR)
```

PROGRAM ARCHING (CONCLUDED)

```
      RETURN
  350 P=0
      RETURN
  360 IF(TIME-TR)342,362,362
  362 IF(TIME-TO)364,364,350
  364 P=PS0
      RETURN
      IF(TIME-TR)342,372,372
  372 IF(TIME-T1)364,364,374
  374 IF(TIME-TO)376,376,350
  376 P=PS0*(TO-TIME)/(TO-T1)
      RETURN
  380 IF(TIME.LE.TT(JJ+1))GOTO 385
      JJ=JJ+1
      PX=PP(JJ+1)-PP(JJ)
      TX=TT(JJ+1)-TT(JJ)
      GOTO 380
  385 P=FACTOR*(PP(JJ)+(TIME-TT(JJ))*PX/TX)
      RETURN
  410 GOTO(415,450,450,460,470),LDTYPE
  450 PRINT 680,TR,TO,PS0
      RETURN
  460 PRINT 685,TR,T1,TO,PS0
      RETURN
  470 DO 480 J=1,NPOINT
      P=FACTOR*PP(J)
  480 PRINT 690,TT(J),P
      RETURN
  610 FORMAT(/*INPUT TR,TO*,↑)
  615 FORMAT(/*INPUT TR,T1,TO*,↑)
  620 FORMAT(/*INPUT NUMBER OF LOAD POINTS AND THE TIME AND *,
     +    *PRESSURE AT EACH POINT*)
  680 FORMAT(/10X,*TR =*,F6.3,* SEC   TO =*,F6.3,* SEC   PS0 =*,
     +    F7.3,* PSI*)
  685 FORMAT(/10X,*TR =*,F6.3,* SEC   T1 =*,F6.3,* SEC    TO =*,
     +    F6.3,* SEC*,/,9X,*PS0 =*,F7.3,* PSI*)
  690 FORMAT(F15.3,F12.2)
  695 FORMAT(/10X,*TIME      PRESSURE*)
```

RCWALL

Reinforced Concrete Wall

PROGRAM RCWALL

```
PROGRAM RCWALL1 (INPUT,OUTPUT)
CALL RETR(7HRCWALL2,7HRCWALL2)
C: THIS PORTION OF THE PROGRAM INPUTS THE REQUIRED WALL AND LOAD
C: DATA AND INITIALIZES CERTAIN PARAMETERS
C
COMMON KWALL,KINC,LDTYPE,KRF,KRAND,TIME,I,Y(100),Q,QU,YU,YFAIL,
+  ZLV,ZLH,TW,PV,FPC,FDY,ICASE,NOBAR,AS(4),APS(4),D(4),DP(4),FDC,
+  EC,ES,R,ALP,ALP2,AREA,ZMASS,QMULT,VFAIL,ZKLM,VH1,VH2,VV1,VV2,
+  W,PO,CO,LOC,S,ZLEN,CD,PSO,PDO,PR,PEXT,PC,TC,TO,DELAY,
+  NWIN,RHOO,VOL1,AA(8,2),NN(8),AFRONT,ASIDE,G,G2,G3,G4,PP2,DT
C
C:  READ TITLE AND CONTROL PARAMETERS
PRINT 67
READ 68,TITLE
PRINT 85
READ,KWALL,KINC,LDTYPE,KRF,KRAND
DELAY=0
VFAIL=1E10
67 FORMAT(/*INPUT TITLE*,+)
68 FORMAT(A59)
85 FORMAT(/*INPUT KWALL(0=EXT,1=INT),KINC,LDTYPE,KRF,KRAND*,
+   *(1=RANDOM)*)
C
C: * DETERMINE WALL PROPERTIES INDEPENDENT OF FDC,FDY, AND D
C
4 DO 5 I=1,4
AS(I)=0
5 CONTINUE
C
C: * INPUT AND ECHO WALL AND REINFORCEMENT PROPERTIES *
PRINT 615
READ,ZLV,ZLH,TW,PV,FPC,FDY,ICASE,ZLVW,ZLHW,NOBAR
FDC=1.25*FPC
EC=57619.0*SQRT(FPC)
ES=29E6
ECKIP=EC/1000$ ESKIP=ES/1000
R=ZLH/ZLV
ALP=1.0/R$ ALP2=ALP*ALP
IF(ICASE.LE.4)GOTO 11
R=0$ ALP=0$ ALP2=0
11 PRINT 670
DO 8 I=1,4
PRINT 625,I
READ,AS(I),D(I),APS(I),DP(I)
IF(I.NE.1)GOTO 3
GOTO(8,8,8,8,9,7,7),ICASE
3 GOTO(9,8,7,6,9,9,9),ICASE
6 IF(I.EQ.3)GOTO 9
GOTO 8
7 I=I+1
8 CONTINUE
9 CONTINUE
C
C: **************************************************
C: * DETERMINE DEFLECTION AND MOMENT COEFFICIENTS *
C: **************************************************
C
AWALL=ZLV*ZLH
AWIN=ZLVW*ZLHW
AREA=AWALL-AWIN
ZMASS=150.0*AREA*TW/(386.07*1728.0)
QMULT=1.0
IF(AWIN.NE.0)CALL WINDOW (QMULT,ZLV,ZLH,ZLVW,ZLHW,AWIN,AWALL,
+      R,ICASE)
PRINT 620,ICASE,ZLV,ZLH,ALP,TW,FPC,ECKIP,FDY,ESKIP,PV,
+  ZLVW,ZLHW,QMULT
PRINT 630
DO 110 I=1,4
IF(AS(I).EQ.0)GOTO 110
P=AS(I)/(12.0*D(I)) $ PP=APS(I)/(12.0*D(I))
PRINT 640,I,AS(I),P,D(I),APS(I),PP,DP(I)
C  CHANGE UNITS OF REINFORCEMENT FROM SQ.IN./FT TO SQ.IN./IN.
AS(I)=AS(I)/12.0
APS(I)=APS(I)/12.0
```

PROGRAM RCWALL (CONTINUED)

```
110 CØNTINUE
615 FØRMAT(/*INPUT LV,LH,TW,PV,F'C,FDY,ICASE,LVW,LHW,NØBAR*)
620 FØRMAT(//*PRØPERTIES ØF REINFØRCED CØNCRETE WALL --*,
+      * SUPPØRT TYPE NØ.*,I2,/,
+      *  LV =*,F6.1,* IN.     LH =*,F6.1,* IN.   LV/LH =*,
+      F5.2/  *  TW =*,F6.1,* IN.     F'C =*,F7.1,* PSI*,5X,
+      *EC =*,F7.1,* KSI*,/,* FDY =*,F8.1,* PSI   ES =*,
+      F8.1,* KSI     PV =*,F6.1,* LB/IN.*/
+      * LVW =*,F6.1,* IN.     LHW =*,F6.1,* IN.   QMULT =*,F6.3)
625 FØRMAT(*INPUT AS,D,A'S,&D' FØR SECTIØN*,I2,+)
630 FØRMAT(//*REINFØRCEMENT VALUES*/* SECTIØN     AS      (P)*,
+      9X,*D*,8X,*A'S     (P')*,8X,*D'*,/,8X,*(SQ.IN./FT.)*,8X,
+      *(IN.)    (SQ.IN./FT.)*,8X,*(IN.)*)
640 FØRMAT(I5,F11.4,* (*,F6.4,*)*,F9.3,F10.4,* (*,F6.4,*)*,F9.3)
670 FØRMAT(1H )
C
C  INPUT LØAD PARAMETERS
IF(LDTYPE.EQ.5)GØTØ 25
100 IF(KRAND.EQ.0)GØTØ 102
W=1000.0 $ PØ=14.7 $ CØ=1120.0 $ LØC=1
GØTØ 106
102 PRINT 600
READ,W,PØ,CØ,LØC,S
IF(LØC.EQ.1)GØTØ 105
PRINT 605
READ,ZLEN,CD
105 IF(KINC.EQ.1)GØTØ 106
PRINT 610
READ,PSØ
PR=2.0*PSØ*(7.0*PØ+4.0*PSØ)/(7.0*PØ+PSØ)
600 FØRMAT(/*INPUT W,PØ,CØ,LØC,S*,+)
605 FØRMAT(/*INPUT L,CD*,+)
610 FØRMAT(/*INPUT PSØ*,+)
C
C * INPUT RØØM-FILLING PARAMETERS *
106 IF(KRF.EQ.0)GØTØ 20
10 PRINT 700
RHØØ=0.076 $ L1=.FALSE.
READ,NWIN,V3
AT=0$ AFRØNT=0$ ASIDE=0
DØ 18 I=1,NWIN
PRINT 710,I
READ,AA(I,1),NN(I),AA(I,2)
AA(I,2)=AA(I,2)/1000.0
AT=AT+AA(I,1)
M=NN(I)$ GØTØ(12,14,14),M
12 AFRØNT=AFRØNT+AA(I,1)
GØTØ 18
14 ASIDE=ASIDE+AA(I,1)
18 CØNTINUE
AFRØNT=AFRØNT/AT$ ASIDE=ASIDE/AT
700 FØRMAT(/*INPUT NWIN AND RØØM VØLUME (CF)*,+)
710 FØRMAT(/*INPUT AREA (SQ FT),LØCATIØN CØDE & DELAY(MSEC)*
+      * FØR WINDØW*,I2,+)
G=1.4 $ G2=1./G $ G3=1.-G2 $ G4=2./G3 $ G5=G+1.
PP2=.1912
C=SQRT(G*PØ*32.*144./RHØØ)
TAU=2.*(V3**(1./3.))/C
DT=TAU/4.0
C
20 IF(KWALL.EQ.0)GØTØ 25
PRINT 86
86 FØRMAT(/*INPUT DELAY TIME (MSEC) TØ INITIAL LØADING *,
+      *INTERIØR WALL*,+)
READ,DELAY
DELAY=DELAY/1000.0
25 CALL CHAIN(RCWALL2)
99 STØP
END
SUBRØUTINE WINDØW (QMULT,ZLV,ZLH,ZLVW,ZLHW,AWIN,AWALL,R,ICASE)
C
C: THIS SUBRØUTINE DETERMINES THE STRUCTURAL
C: MØDIFICATIØN FACTØR FØR WALLS WITH WINDØWS
IF(ICASE.GT.4.AND.ICASE.NE.10)GØTØ 320
```

PROGRAM RCWALL (CONTINUED)

```
RWWS=ZLVW/ZLV
RWWL=ZLHW/ZLH
RAREA=AWIN/AWALL
IF(R.LE.1.5)GOTO 300
IF(RWWS.GT.0.7)GOTO 300
IF(RWWL.LT.0.5)GOTO 300
IF(RWWS.EQ.RWWL)GOTO 300
C
C:  CASE WHERE LV/LH >= 1.5, LVW/LV <= 0.7, AND LHW/LH >= 0.5
C:  (BUT LVW/LV NOT EQUAL TO LHW/LH)
OMULT=-5.85461-12.6644*RAREA+4.39662*RWWS+0.84843*RWWL
+      -0.2230*R-1.07269*(ZLVW/ZLHW)**0.9+6.59542*EXP(RAREA)
GOTO 315
C
C:  CASE WHERE ONE OR MORE OF ABOVE CONDITIONS IS NOT MET
300 OMULT=0.62022-2.23415*RAREA**(RWWL**4)-0.79461*RWWL**2
+      -2.27663*RWWL+0.62522*RWWL/RAREA**0.3
+      +2.63043*EXP(RAREA)-0.09268*RWWS
315 CONTINUE
RETURN
C
C  ONE-WAY ACTION WALLS
320 OMULT=(AWALL-ZLV*ZLHW)/(AWALL-AWIN)
RETURN
END
SEGMENT RCWALL2 (INPUT,OUTPUT)
C  THIS SEGMENT CALCULATES THE RESISTANCE FUNCTION AND
C  LOADING AND SOLVES THE DYNAMIC EQUATIONS OF MOTION
C
COMMON KWALL,KINC,LDTYPE,KRF,KRAND,TIME,I,Y(100),Q,QU,YU,YFAIL,
+  ZLV,ZLH,TW,PV,FPC,FDY,ICASE,NOBAR,AS(4),APS(4),D(4),DP(4),FDC,
+  EC,ES,R,ALP,ALP2,AREA,ZMASS,OMULT,VFAIL,ZKLM,VH1,VH2,VV1,VV2,
+  W,P3,CO,L3C,S,ZLEN,CD,PSO,PDO,PF,PEXT,PC,TC,TO,DELAY,
+  NWIN,RHOO,V3,L1,AA(8,2),NN(8),AFRONT,ASIDE,G,G2,G3,G4,PP2,DT
COMMON /RAND/ TIMEC
DIMENSION A(80),V(80),T(80),VV(80),VH(80),PEX(80),PIN(80),PN(80)
C
IF(KINC.NE.1.OR.LDTYPE.EQ.5)CALL FORCE(1)
14    IF(KRAND.NE.1)GOTO 35
      CALL FORCE(4)
      CALL RANDOM(1)
34    CALL RANDOM(2)
35    CALL RESIST(3)
C
C: MINIMUM, MAXIMUM, AND STARTING VALUES ARE DETERMINED FOR CASES
C: WHERE THE LOAD CAUSING INCIPIENT COLLAPSE IS TO BE FOUND
13    IF(KINC.EQ.0)GOTO 23
       PF=QU
       PFMAX=0
       PFMIN=PF/2.0
      GOTO 20
16     PF=(PFMIN+PFMAX)/2.0
20    CALL FORCE(2)
23    IF(KRF.EQ.0)GOTO 24
      CALL FILL(PINT,2)
C
C: INITIALIZE VALUES FOR BETA METHOD (BETA = 1/6) AND COMPUTE VALU
C: FOR FIRST TIME INTERVAL ASSUMING WALL INITIALLY AT REST
24     I=1
       TIME=0
      T(1)=0$ V(1)=0$ Y(1)=0
       DELTA=0.001
      IF(KWALL.EQ.0)GOTO 29
27    IF(TIME.GE.(DELAY-0.00001))GOTO 28
      TIME=TIME+DELTA
      CALL FILL(PINT,3)
      GOTO 27
28    PIN(1)=PINT
      TPNET=AREA*PINT
      T(1)=TIME
      GOTO 30
29    CALL FORCE(3)
      TPNET=AREA*PEXT
       PN(1)=PEXT
```

PROGRAM RCWALL (CONTINUED)

```
30    CALL RESIST(2)
A(1)=TPNET/(ZMASS*ZKLM)
VV(1)=VV1*TPNET
VH(1)=VH1*TPNET
C
C: PROCEDURE FOR ALL SUBSEQUENT TIME INTERVALS
1       I=I+1
IF(I.LT.31)GOTO 11
        PRINT 99,TIME
99 FORMAT(/*I=31:  TIME =*,F6.3,*  FAILURE ASSUMED TO NOT OCCUR*)
        GOTO 6
11    TIME=TIME+DELTA
       T(I)=TIME
       A(I)=A(I-1)
      CALL FORCE(3)
       PEX(I)=PEXT
      IF(KWALL.EQ.1)GOTO 10
      IF(KRF.EQ.0)GOTO 3
      CALL FILL(PINT,3)
       PIN(I)=PINT
      TPNET=AREA*(PEXT-PINT)
       GOTO 2
3      TPNET=AREA*PEXT
      GOTO 2
10    CALL FILL(PINT,3)
      PIN(I)=PINT
      TPNET=AREA*PINT
2     PN(I)=TPNET/AREA
      DO 8 II=1,10
       Y(I)=Y(I-1)+DELTA*V(I-1)+DELTA*DELTA*(A(I-1)/3.+A(I)/6.)
       CALL RESIST(2)
       QT=Q*AREA
4 ANEW=(TPNET-QT)/(ZMASS*ZKLM)
       ADELTA=ANEW-A(I)
       A(I)=ANEW
IF(ANEW.EQ.0.0)PRINT,*1985*,TIME,TPNET,QT,ZMASS,ZKLM,Y(I),A(I-1)
      IF(ABS(ADELTA/(ANEW+1E-10)).LT.0.01)GOTO 9
8      CONTINUE
       A(I)=ANEW-ADELTA/2.0
PRINT 80,TIME,PF,A(I),Y(I)
9      CONTINUE
       Y(I)=Y(I-1)+DELTA*V(I-1)+DELTA*DELTA*(A(I-1)/3.+A(I)/6.)
       V(I)=V(I-1)+DELTA*(A(I)+A(I-1))/2.0
VV(I)=VV1*TPNET+VV2*QT
VH(I)=VH1*TPNET+VH2*QT
C
C: CHECK FOR MAXIMUM DEFLECTION OR FAILURE OF WALL
C:   IF MAXIMUM DEFLECTION REACHED, WALL DID NOT FAIL
       IF(Y(I).LE.Y(I-1).AND.PN(I).LE.PN(I-1))GOTO 6
       IF(Y(I).LT.0)GOTO 6
       IF(TIME-DELAY.GE.0.010)DELTA=0.002
       IF(TIME-DELAY.GE.0.020)DELTA=0.005
       IF(TIME-DELAY.GE.0.100)DELTA=0.010
       IF(TIME-DELAY.GE.0.500)DELTA=0.050
C   IF FAILURE DEFLECTION REACHED, WALL FAILED
       IF(Y(I).GE.YFAIL)GOTO 7
        GOTO 1
C
C: INTERVAL HALVING PROCEDURE TO DETERMINE LOAD CAUSING INCIPIENT
C: COLLAPSE FOR CASES WHERE DESIRED
C:   WALL DID NOT FAIL - SET PFMIN TO PF
6     CONTINUE
      IF(KINC.EQ.0)GOTO 18
36    PFMIN=PF
IF(PFMAX.GT.0)GOTO 16
PF=2.0*PF
       GOTO 20
C:   WALL FAILED - SET PFMAX TO PF
7     CONTINUE
      TIMEC=TIME
      IF(KINC.EQ.0)GOTO 18
37    PFMAX=PF
C:   CHECK TO SEE IF LOAD RANGE IS WITHIN DESIRED ACCURACY
17     IF((PFMAX-PFMIN)/PFMIN.GT.0.01)GOTO 16
```

```
      IF(KRAND.NE.1)GØTØ 18
      CALL RANDØM(3)
      GØTØ 34
C
C: ØUTPUT DATA INCLUDES THE MAXIMUM DEFLECTIØN AND TIME ØF
C: ØCCURANCE FØR A NØN-FAILING WALL ØR THE TIME AND VELØCITY
C: AT CØLLAPSE FØR A FAILING WALL.  ØPTIØNAL ØUTPUT IS THE
C: ENTIRE BEHAVIØR TIME-HISTØRY ØF THE WALL.
C
C: ØUTPUT LØAD DATA
 18   CALL FØRCE(4)
C
C: ØUTPUT FINAL RESULTS
 IF(Y(I).LT.YFAIL)PRINT 70,Y(I),T(I)
 IF(Y(I).GE.YFAIL)PRINT 71,T(I),V(I)
C
C: CHECK TØ SEE IF ENTIRE TIME-HISTØRY ØF WALL IS DESIRED
 PRINT 72
 READ,M
      IF(M.EQ.0)GØTØ 25
      IF(KWALL.EQ.1)GØTØ 32
      IF(KRF.EQ.0)GØTØ 26
 PRINT 75,(T(J),PEX(J),PIN(J),PN(J),Y(J),VV(J),VH(J),J=1,I)
      GØTØ 25
 26 PRINT 76,(T(J),PEX(J),A(J),V(J),Y(J),VV(J),VH(J),J=1,I)
      GØTØ 25
 32 PRINT 76,(T(J),PIN(J),A(J),V(J),Y(J),VV(J),VH(J),J=1,I)
 25 PRINT 77
 STØP
C
 70 FØRMAT(/*WALL DID NØT FAIL - MAX. DEFLECTIØN ØF*F6.2
+    * IN. REACHED AT*F7.3,* SEC*)
 71 FØRMAT(/*WALL FAILED AT*,F7.3,* SEC (FINAL VELØCITY =*
+   F7.2* IN./SEC)*)
 72 FØRMAT(/*IS TIME HISTØRY ØF WALL DESIRED (YES=1, NØ=0)*,+)
 75 FØRMAT(15X*PRESSURE ØN WALL*/*  TIME   EXTERIØR  *
+ *INTERIØR    NET     DISPLACEMENT    VV      VH*/
+   (F6.3,3F10.3,F12.4,F11.0,F8.0))
 76 FØRMAT(/*  TIME  PRESSURE  ACCELERATIØN  VELØCITY  *
+   *DISPLACEMENT     VV       VH*/
+   (F6.3,F9.3,F12.1,F12.2,F12.4,F11.0,F8.0))
 77     FØRMAT(///,7(*----------*))
 80 FØRMAT(/*ACCELERATIØN NØT CØNVERGING AT TIME =*,F6.3,
+            * SEC (PF =*,F7.3,* PSI)*/*      A(I) SET EQUAL TØ*,
+            F8.1,* (AVG ØF LAST 2 ITERATIØNS)*/*      Y(I) =*,
+            F6.4,* IN.*)
 999   STØP
       END
 SUBRØUTINE FØRCE(IENTRY)
C  THIS SUBRØUTINE INPUTS THE LØAD PARAMETERS AND DETERMINES
C  THE LØAD AT A GIVEN TIME FØR THE FØLLØWING LØAD TYPES:
C      1. IDEALIZED BLAST LØAD (FRØNT ØR SIDE FACE)
 CØMMØN KWALL,KINC,LDTYPE,KRF,KRAND,TIME,I,Y(100),C,QU,YU,YFAIL,
+  ZLV,ZLH,TW,PV,FPC,FDY,ICASE,NØBAR,AS(4),APS(4),D(4),DP(4),FDC,
+   EC,ES,R,ALF,ALF2,AREA,ZMASS,QMULT,VFAIL,ZKLM,VM1,VM2,VV1,VV2,
+   W,PØ,CØ,LØC,S,ZLEN,CD,PSØ,PDØ,PR,P,PC,TC,TO,DELAY,
+   NWIN,RHØB,V3,L1,AA(8,2),VV(8),AFRØNT,ASIDE,G,G2,G3,G4,PP2,DT
 DIMENSIØN TT(10),PP(10)
C
 IF(LDTYPE.EQ.5)GØTØ 500
C
 GØTØ(215,200,300,4),IENTRY
C
C  CALCULATE LØAD PRØPERTIES FØR GIVEN PEAK PRESSURE
 200 GØTØ(205,210),LØC
 205 PSØ=(PR-14.0*PØ+SQRT(196.0*PØ*PØ+196.0*PØ*PR+PR*PR))/15.0
 GØTØ 215
 210 PSØ=PR
 215 PDØ=2.5*PSØ*PSØ/(7.0*PØ+PSØ)
 U=CØ*SQRT(1.0+(6.0*PSØ)/(7.0*PØ))
 TO=W**0.3333/(2.2399+0.1596*PSØ)
 GØTØ(220,225),LØC
 220 TC=3.0*S/U
 PC=PSØ*(1-TC/TO)*EXP(-TC/TO)+PDØ*(1-TC/TO)**2*EXP(-2*TC/TO)
```

PROGRAM RCWALL (CONTINUED)

```
CO=1.0
RETURN
225 TA=ZLEN/1
TA2=TA/2.0
TA2TO=TA2/TO
PA=PSO*(1-TA2TO)*EXP(-TA2TO)+CO*PO2*(1-TA2TO)**2*EXP(-2*TA2TO)
RETURN
C
C  CALCULATE LOAD
300 GOTO(305,310),LOC
305 TTO=TIME/TO
IF(TIME.GT.TC)GOTO 320
P=PC+(TC-TIME)*(PA-PC)/TC
RETURN
310 TTO=(TIME-TA2)/TO
IF(TIME.GT.TA)GOTO 320
P=PA*TIME/TA
RETURN
320 IF(TTO.GE.1.0)GOTO 330
P=PSO*(1-TTO)*EXP(-TTO)+CO*PO2*(1-TTO)**2*EXP(-2*TTO)
RETURN
330 P=0
RETURN
C
C  PRINT LOAD DATA
4 IF(LTC.EQ.1)GOTO 400
PRINT 640,LTYPE
GOTO 410
400 PRINT 645,LTYPE
410 CONTINUE
415 GOTO(420,425),LOC
420 PRINT 650
GOTO 430
425 PRINT 655
430 PRINT 660,W,RO,CO
IF(KRAND.NE.0)RETURN
GOTO(435,440),LOC
435 PRINT 665,S,TC,PC
GOTO 445
440 PRINT 670,ZLEN,TA,PA
445 PRINT 675,TO,CO,PSO,PO2
RETURN
C
C  LOAD TYPE 5 -- ARBITRARY LOAD SHAPE
500 GOTO(510,520,530,540),IENTRY
C
C  INPUT LOAD DATA
510 PRINT 680
READ,NPOINT,(TT(I),PP(I),I=1,NPOINT)
FACTOR=1.0
IF(ING.EQ.0)GOTO 518
PMAX=PP(1)
DO 515 J=2,NPOINT
515 IF(PP(J).GT.PMAX)PMAX=PP(J)
518 PX=PP(2)-PP(1)
TX=TT(2)-TT(1)
JJ=1
RETURN
C
C  CALCULATE MAXIMUM LOAD
520 FACTOR=PR/PMAX
GOTO 518
RETURN
C
C  CALCULATE LOAD
530 IF(TIME.LE.TT(JJ+1))GOTO 535
JJ=JJ+1
PX=PP(JJ+1)-PP(JJ)
TX=TT(JJ+1)-TT(JJ)
IF(TX.EQ.0)TX=1E-10
GOTO 530
535 P=FACTOR*(PP(JJ)+(TIME-TT(JJ))*PX/TX)
RETURN
C
```

```
C  PRINT LOAD DATA
540 IF(KINC.EQ.1)PRINT 640,LDTYPE
IF(KINC.EQ.0)PRINT 645,LDTYPE
PRINT 690
DO 545 I=1,NPOINT
P=FACTOR*PP(I)
545 PRINT 695,TT(I),P
RETURN
C
600 FORMAT(/*INPUT W,P0,C0,L0C,S*,*)
605 FORMAT(/*INPUT L,CD*,*)
630 FORMAT(/*INPUT PSO*,*)
640 FORMAT(/*LOAD -USING INCIPIENT FAILURE IS AS FOLLOWS:*,
+     /,5X,*LOAD TYPE NUMBER*,I2)
645 FORMAT(/*PROPERTIES OF LOAD ACTING ON WALL ARE AS FOLLOWS:*,
+     /,5X,*LOAD TYPE NUMBER*,I2)
650 FORMAT(8X,*(FRONT FACE)*)
655 FORMAT(8X,*(SIDE FACE)*)
660 FORMAT(10X,*W =*,F8.1,* KT   P0 =*,F6.2,* PSI     C0 =*,
+     F7.1,* FPS*)
665 FORMAT(10X,*S =*,F6.1,* FT     TC =*,F6.3,* SEC     PR =*,
+     F7.3,* PSI*)
670 FORMAT(10X,*L =*,F6.1,* FT     TA =*,F6.3,* SEC     PA =*,
+     F7.3,* PSI*)
675 FORMAT(10X,*U =*,F7.1,* FPS   TO =*,F6.3,* SEC     CD =*,
+     F5.1,/,8X,*PSO =*,F7.3,* PSI  PDO =*,F7.3,* PSI*)
680 FORMAT(/*INPUT NUMBER OF LOAD POINTS AND THE TIME AND *,
+     *PRESSURE AT EACH POINT*)
690 FORMAT(/10X,*TIME        PRESSURE*)
695 FORMAT(F15.3,F12.2)
END
SUBROUTINE FILL(P3,IENTRY)
C: COMPUTES AVERAGE AIR PRESSURE IN ROOM DUE TO BLAST WAVE
C: INCIDENT HEAD-ON UPON FRONT WALL.
C
COMMON KWALL,KINC,LDTYPE,KRF,KRAND,TIME,II,Y(100),Q,QU,YU,YFAIL,
+  ZLV,ZLH,TW,PV,FPC,FDY,ICASE,NOBAR,AS(4),APS(4),D(4),DP(4),FDC,
+  EC,ES,R,ALP,ALP2,AREA,ZMASS,QMULT,VFAIL,ZKLM,VM1,VM2,VV1,VV2,
+  W,P0,C0,L0C,S,ZLEN,CD,PSO,PDO,PR,PEXT,PC,TC,TO,DELAY,
+  NWIN,RHOO,V3,L1,AA(8,2),NN(8),AFRONT,ASIDE,G,G2,G3,G4,PP2,DT
LOGICAL L1,L2,L3
C
GOTO(10,13,11),IENTRY
10 RETURN
C
13 P30=P3
TT=0.$ TO=0.
RHO30=RHOO
L2=.FALSE. $ L3=.FALSE.
RETURN
C
11 IF(L1)GOTO 52
IF(L2.A.L3)GOTO 9
52 DDT=(TIME-TO)*0.5
ISTOP=2
53 IF(DDT.LT.DT)GOTO 51
50 DDT=0.5*DDT
ISTOP=2*ISTOP
GO TO 53
51 CONTINUE
DO 99 I=1,ISTOP
TT=TO+I*DDT
IF(TT.GT.TO)GO TO 99
DM=0. $ WW=0. $ VW=0
DO 500 K=1,NWIN
M=NN(K) $ DLY=AA(K,2)+0.000001
IF(DLY.GE.TT)GO TO 500
GOTO(15,16,16),M
15 CDF=1.0
IF(TT-TC)20,20,21
20  P11=(TC-TT)*(PR-PC)/TC+PC
P11=P11+P0
GO TO 30
16 CDF=-0.4
```

PROGRAM RCWALL (CONTINUED)

```
20600 21   R=TT/TO $ RR=1.-R
20610 PD=PD3*RR*RR*EXP(-2.*R)
20620 PS=PS3*RR*EXP(-R)
20630 P11=PS+CDF*PD
20640 P11=P11+P1
20650 30  RH31=RH33*((P11/P3)**G2)
20660 IF(P11-P33)36,36,37
20670      ISIGN=-1
20680 L2=.TRUE.
20770 303  P2=P11
20780 RH32=((P2/P33)**G2)*RH33)
20790 X=P33/RH333
20800 GO TO 38
20810 37  ISIGN=+1
20820 306 P2=PP2*P11
20830 RH32=((P2/P11)**G2)*RH31
20840 X=P11/RH31
20850 38  U22=G4*(X-P2/RH32)*32.*144.
20860 IF(U22)40,39,39
20870 40  PRINT,*U22 NEGATIVE*,U22
20880 STOP
20890 39  U2=SQRT(U22)*ISIGN
20900 DDM=U2*RH32*AA(K,1)*DDT
20910 DM=DM+DDM
20920 ww=...+P11*DDM/(G3*RH31)
20925C
20930 500 CONTINUE
20940 P33=P33+(G-1.)*WW/V3
20950 RH333=RH333+DM/V3
20960 99  CONTINUE
20970 TO=TT
20980 P3=P33-P0
20982 IF(TIME.GE.TC)L3=.TRUE.
20983 RETURN
20984 9 R=TIME/TO $ RR=1.0-R
20985 PD=PD3*RR*RR*EXP(-2.0*R)
20986 PS=PS3*RR*EXP(-R)
20987 P3=PS+PD*(AFRONT-0.4*ASIDE)
20990 999 RETURN
21020 END
30000 SUBROUTINE RESIST (IENTRY)
30010C: THIS SUBROUTINE INPUTS THE REQUIRED WALL DATAS DETERMINES THE
30020C: RESISTANCE FUNCTION, TRANSFORMATION FACTORS, AND REACTION
30030C: COEFFICIENTSS AND SUPPLIES THE REACTION VALUES FOR SPECIFIC
30040C: DEFLECTIONS REQUIRED IN THE DYNAMIC ANALYSIS
30045C
30050 COMMON KWALL,KINC,LDTYPE,KRF,KRAND,TIME,I,Y(100),D,DU,YU,YFAIL,
30052+   ZLV,ZLH,TW,PV,FDC,FDY,ICASE,NOBAR,AS(4),APS(4),D(4),DP(4),FDC,
30054+   EC,ES,R,ALP,ALP2,AREA,ZMASS,DMULT,VFAIL,ZKLM,VH1,VH2,VV1,VV2,
30056+   W,P3,C3,LOC,S,ZLEN,CD,PS3,PD3,PR,PEXT,PC,TC,TO,DELAY,
30058+   NWIN,RH333,V3,L1,AA(3,2),NN(8),AFRONT,ASIDE,G,G2,G3,G4,PP2,DT
30100 REAL N,IC,IG,MM,KK1,KK2,KK3,MU(4),ICR(4)
30135C
30140 GOTO(4,500,45),IENTRY
30150 4 RETURN
30790C
30800C: ****************************************
30810C: * ENTRY 2: DETERMINE WALL PROPERTIES    *
30820C: *           DEPENDENT ON FDC, FDY, AND D *
30830C: ****************************************
30840C
30850 45 N=ES/EC
30900 FR=8.0*SQRT(FDC)
30916 IG=TW**3/12+(N-1)*(AS(1)*(D(1)-TW/2)**2+APS(1)*(TW/2-DP(1))**2)
30917 CALL COEF(ICASE,R,ASS,ASS,DF,RF,IG,ZLV,ZLH,PV,NK,CF,EC,1)
30920 MM=2.0*IG*(FR+PV/TW)/TW
30925 CALL MOMENT(FDC,FDY,ES,N,PV,1.0,AS,APS,D,DP,MU,ICR,IC)
31470 GMU=MU(2)/MU(1)
31480C
31490C: DETERMINE POSITION OF YIELD LINES AND ULTIMATE RESISTANCE
31500C: COEFFICIENTS FOR TWO-WAY WALLS
31510 IF(ICASE.GT.4)GOTO 106
31520 Z11=MU(4)/MU(2)
31530 Z13=MU(3)/MU(1)
```

PROGRAM RCWALL (CONTINUED)

```
GAMMA12=2.0*SQRT(1.0+ZI1)
GAMMA34=2.0*SQRT(1.0+ZI3)
GRAT=GAMMA12/GAMMA34
B=SQRT(1+ZI1)*(GMU*ALP2/GAMMA34)*(SQRT(GRAT**2+3/
+   (GMU*ALP2))-GRAT)
C: IF BETA IS GREATER THAN 0.5 ASSUMPTION OF CRACK PATTERN A
C: NOT VALID AND CRACK PATTERN B IS ASSUMED TO OCCUR
IF(B.LE.0.5)GOTO 105
C
C: CRACK PATTERN B
KRAK=1
B=SQRT(1.0+ZI3)*(SQRT(1.0/GRAT**2+GMU*ALP2*3.0)-1.0/GRAT)
+  /(GMU*ALP2*GAMMA12)
QUTERM=6.0*GAMMA12**2*ALP*GMU/(SQRT(3+1/(GMU*ALP2*GRAT**2))
+  -1.0/(SQRT(GMU)*ALP*GRAT))**2
GOTO 108
C
C: CRACK PATTERN A
105 CONTINUE
KRAK=0
QUTERM=6.0*GAMMA34**2/(ALP*(SQRT(3+GMU*ALP2*GRAT**2)
+  -ALP*GRAT*SQRT(GMU))**2)
GOTO 106
C
C: DETERMINE MOMENT AND DEFLECTION COEFFICIENTS
C: FOR CRACKED PORTION OF WALL BEHAVIOR
106 B=0
108 IF(PV.EQ.0)GOTO 180
CALL COEF(ICASE,R,ASSC,BSSC,AFC,BFC,IC,ZLV,ZLH,PV,NX,CF,EC,2)
GOTO(195,195,195,195,112,115,120),ICASE
112 QUTERM=1.0/(BSSC*ZLV)
GOTO 195
115 QUTERM=(1.0/BSSC+(ZLV*ZLV*PV/(EC*IC))*MU(3)/(MU(1)*(1.0
+  -COS(0.5*ZLV*SQRT(PV/(EC*IC))))))/ZLV
CF=BFC
GOTO 195
120 QUTERM=(1.0+0.5*MU(3)/(MU(1)*COS(0.5*ZLV*SQRT
+  (PV/(EC*IC)))))/(BSSC*ZLV)
CF=BFC
GOTO 195
180 ASSC=ASS$ AFC=AF
GOTO(195,195,195,195,182,185,190),ICASE
182 QUTERM=1.0/(BSS*ZLV)
GOTO 195
185 QUTERM=(MU(3)/MU(1)+1.0)/(BSS*ZLV)
GOTO 195
190 QUTERM=(0.5*MU(3)/MU(1)+1.0)/(BSS*ZLV)
195 GOTO(200,210,210,210,200,210,210),ICASE
C
C: ***************************************************************
C: * DETERMINE RESISTANCE CURVE FOR WALL                         *
C: * (Q IS IN UNITS OF PSI, KK IN LB/CU.IN., AND Y IN INCHES)    *
C: ***************************************************************
C
C: CASES 1 AND 5
200 Q1=MM/(BSS*ZLV*ZLV)
KK1=EC*IG/(ASS*ZLV**4)
Y1=Q1/KK1
KK2=EC*IC/(ASSC*ZLV**4)
IF(ICASE.EQ.5)GOTO 205
QU=QUTERM*MU(1)/(ZLV*ZLH+0.5*QUTERM*PV/KK2)
GOTO 208
205 QU=QUTERM*MU(1)/ZLV
208 YU=QU/KK2
GOTO 280
C
C: CASES 2, 3, 4, 6, & 7
210 Q1=MM/(BF*ZLV*ZLV)
KK1=EC*IG/(AF*ZLV**4)
Y1=Q1/KK1
Q2=MU(NX)/(CF*ZLV*ZLV)
KK2=EC*IC/(AFC*ZLV**4)
Y2=Q2/KK2
KK3=EC*IC/(ASSC*ZLV**4)
```

```
IF(ICASE.GT.4)GOTO 215
QU=QUTERM*(MU(1)-PV*(Y2-Q2/KK3)/2)/(ZLV*ZLH+0.5*QUTERM*PV/KK3)
GOTO 220
215 QU=QUTERM*MU(1)/ZLV
220 YU=Y2+(QU-Q2)/KK3
230 IF(PV.NE.0)YFV=MU(1)/(PV*(1.0-B))
C
C: CHECK FOR TYPE OF FAILURE - LIGHTLY REINFORCED OR CONVENTIONAL
IF(MU(1).LT.1.5*MM)GOTO 288
C
C: CONVENTIONAL TYPE FAILURE
YFAIL=YU*0.1/(AS(1)/D(1))
C: DUCTILITY FACTOR MUST BE <= 30
IF(YFAIL.GT.30.0*YU)YFAIL=30.0*YU
GOTO 285
C
C: LIGHTLY REINFORCED TYPE OF FAILURE
C:  THE FOLLOWING EXPRESSION IS BASED ON A STEEL ELONGATION OF 20%
288 UCOEF=30.
UPU=UCOEF*SQRT(FDC)
ABAR=3.14159*(NOBAR/16.)**2
290 YFAIL=SQRT((0.2*ABAR*FDY/UPU+ZLV/2.)**2-(ZLV/2.)**2)
285 IF(PV.EQ.0)GOTO 295
C:
C: IF FAILURE DEFLECTION DUE TO INSTABILITY IS LESS THAN VALUE
C: BASED ON REINFORCEMENT, SUBSTITUTE THIS VALUE FOR YFAIL
IF(YFAIL.GT.YFV)YFAIL=YFV
QFAIL=QU*(YFV-YFAIL)/(YFV-YU)
GOTO 298
295 QFAIL=QU
298 CONTINUE
C
C: MODIFY RESISTANCE VALUES BY WINDOW MODIFICATION FACTOR
Q1=Q1*QMULT
Q2=Q2*QMULT
QU=QU*QMULT
QFAIL=QFAIL*QMULT
KK1=KK1*QMULTS $ KK2=KK2*QMULTS $ KK3=KK3*QMULT
C
C: OUTPUT LOAD-DEFLECTION CURVE
IF(KRAND.EQ.1)GOTO 335
PRINT 650
IF(ICASE.EQ.1.OR.ICASE.EQ.5)GOTO 320
PRINT 660,Q1,Y1,Q2,Y2
GOTO 330
320 PRINT 660,Q1,Y1
330 PRINT 660,QU,YU,QFAIL,YFAIL
335 CONTINUE
C
CALL TRANS (B,ZLV,ZLH,ICASE,KRAK,ZKLMSE,ZKLMFE,ZKLMP,VH1S,VH2S,
+  VV1S,VV2S,VH1F,VH2F,VV1F,VV2F,VH1P,VH2P,VV1P,VV2P)
RETURN
C
C: ****************************************************
C: * ENTRY 3: DETERMINE THE RESISTANCE (PER UNIT AREA) *
C: *          OF THE WALL AS A FUNCTION OF Y(I)        *
C: ****************************************************
C
500 IF(Y(I).GE.YFAIL)GOTO 560
IF(Y(I).GT.YU)GOTO 540
GOTO(501,520,520,520,501,520,520),ICASE
501 CONTINUE
C
C: ELASTIC RANGE -- CASES 1 AND 5
ZKLM=ZKLMSE
VH1=VH1S $ VH2=VH2S
VV1=VV1S $ VV2=VV2S
IF(Y(I).GT.Y1)GOTO 510
C
C: UNCRACKED PORTION -- ALL CASES
505 Q=Y(I)*KK1
RETURN
C
C: CRACKED PORTION -- CASES 1 AND 5
```

PROGRAM RCWALL (CONTINUED)

```
510 Q=Q1+(Y(I)-Y1)*(QU-Q1)/(YU-Y1)
RETURN
C
520 IF(Y(I).GT.Y2)GOTO 530
C
C: ELASTIC RANGE -- CASES 2,3,4,6,7
ZKLM=ZKLMFE
VH1=VH1F $ VH2=VH2F
VV1=VV1F $ VV2=VV2F
IF(Y(I).LT.Y1)GOTO 505
C: CRACKED PORTION -- CASES 2,3,4,6,7
Q=Q1+(Y(I)-Y1)*(Q2-Q1)/(Y2-Y1)
RETURN
C
C: ELASTO-PLASTIC RANGE -- CASES 2,3,4,6,7
530 ZKLM=ZKLMSE
VH1=VH1S $ VH2=VH2S
VV1=VV1S $ VV2=VV2S
Q=Q2+KK3*(Y(I)-Y2)
RETURN
C
C: PLASTIC RANGE -- ALL CASES
540 ZKLM=ZKLMP
VH1=VH1P $ VH2=VH2P
VV1=VV1P $ VV2=VV2P
IF(PV.GT.0)GOTO 550
C: NO INPLANE FORCES
Q=QU
RETURN
C: WITH INPLANE FORCES
550 Q=QU*(YFV-Y(I))/(YFV-YU)
RETURN
C
C: WALL COLLAPSED - NO RESISTANCE (TO AVOID NUMERICAL DIFFICULTIEES
C: FOR CERTAIN CASES SET RESISTANCE EQUAL TO VERY SMALL VALUE)
560 Q=1E-10
RETURN
C
650 FORMAT(//*LOAD-DEFLECTION CURVE*,/,3X,*Q (PSI)     Y (IN.)*)
660 FORMAT(F9.2,F12.4)
END
SUBROUTINE MOMENT(FDC,FDY,ES,N,PV,B,AS,APS,D,DP,MU,ICR,IC)
C THIS SUBROUTINE DETERMINES THE ULTIMATE MOMENT CAPACITY AND
C CRACKED MOMENT OF INERTIA  FOR REQUIRED SECTIONS
REAL [illegible]K2,K3,KUD,N,IC,ICTOT,MU(4),ICR(4),AS(4),APS(4),D(4),DP(4)
C
C: DETERMINE VALUES OF CONCRETE PARAMETERS
45 K1=0.94-FDC/26E3
K2=0.50-FDC/8E4
K3=(3900.0+0.35*FDC)/(3E3+0.82*FDC-FDC*FDC/26E3)
EPSC=0.004-FDC/65E5
C: ***********************************************
C: * DETERMINE ULTIMATE MOMENT CAPACITY AND CRACKED *
C: * MOMENT OF INERTIA FOR REQUIRED SECTIONS        *
C: ***********************************************
C
II=0$ ICTOT=0
DO 170 I=1,4
IF(AS(I).EQ.0)GOTO 170
II=II+1
C: ALL PROPERTIES ARE COMPUTED FOR A WIDTH B
TENS=AS(I)*FDY+PV
IF(APS(I).LE.0)GOTO 150
C
C: WALL HAS COMPRESSION REINFORCEMENT
C=K1*K3*FDC*B*DP(I)
TERM1=0.5*(TENS/APS(I)+ES*EPSC)
TERM2=ES*EPSC*(TENS-C)/APS(I)
C: DETERMINE LOCATION OF NEUTRAL AXIS
IF(TENS.LE.C)GOTO 140
C
C: KUD > D'
FPS=TERM1+K3*FDC/2.0-SQRT((TERM1+K3*FDC/2.0)**2
+  -(TERM2+ES*EPSC*K3*FDC))
```

PROGRAM RCWALL (CONTINUED)

```
C: F'S MUST BE <= FDY
IF(FPS.LT.FDY)GOTO 130
FPS=FDY
130 TPS=APS(I)*(FPS-K3*FDC)
KUD=(TENS-TPS)/(K1*K3*FDC*B)
MU(I)=(TENS-TPS)*(D(I)-K2*KUD)+TPS*(D(I)-DP(I))
ICR(I)=B*KUD**3/3.0+N*AS(I)*(D(I)-KUD)**2
+  +(N-1)*APS(I)*(KUD-DP(I))**2
GOTO 152
C
C: KUD < D'
140 FPS=-TERM1+SQRT(TERM1**2-TERM2)
C: F'S MUST BE <= FDY
IF(FPS.LT.FDY)GOTO 145
FPS=FDY
145 TERM3=TENS+APS(I)*FPS
KUD=TERM3/(K1*K3*FDC*B)
MU(I)=TERM3*(D(I)-K2*KUD)-APS(I)*FPS*(D(I)-DP(I))
ICR(I)=B*KUD**3/3+N*AS(I)*(D(I)-KUD)**2+N*APS(I)*(DP(I)-KUD)**2
GOTO 152
C
C: WALL HAS NO COMPRESSION REINFORCEMENT
150 KUD=TENS/(K1*K3*FDC*B)
MU(I)=TENS*(D(I)-K2*KUD)
ICR(I)=B*KUD**3/3.0+N*AS(I)*(D(I)-KUD)**2
C
152 ICTOT=ICTOT+ICR(I)
170 CONTINUE
C
C: DETERMINE AVERAGE CRACKED MOMENT OF INERTIA
175 IC=ICTOT/II
RETURN
END
SUBROUTINE COEF(ICASE,R,ASS,BSS,AF,BF,I,ZLV,ZLH,PV,NX,CF,
+      E,IENTRY)
C:  THIS SUBROUTINE DETERMINES MOMENT AND DEFLECTION COEFFICIENTS
C:  FOR ONE-WAY (CASES 5-7) AND TWO-WAY (CASES 1-4) WALLS
C
       REAL I,MPR,MPRSQ,NI
       IF(IENTRY.EQ.2)GOTO 200
       NX=1
        IF(ICASE.GT.4)GOTO 50
C
        R2=R*R
        R3=R*R2
        R4=R2*R2
        ASS=-.007030+.013890*R-.003456*R2+.000286*R3
        BSS=-.058332+.139314*R-.035609*R2+.003016*R3
8       GOTO(41,20,30,40),ICASE
C
C:  CASE 2. FIXED ON FOUR SIDES
20     NX=3
        AF=-.003430+.007327*R-.003365*R2+.0006646*R3-.00004766*R4
        BF=-.101150+.260875*R-.138982*R2+.034677*R3-.004016*R4
+    +.000170*R**5
       CF=-.1674+.3554*R-.1714*R2+.0286*R3
        GOTO 41
C
C:  CASE 3. FIXED ON SHORT SIDES, SIMPLY SUPPORTED ON LONG SIDES
30     NX=4
        AF=.004513-.017525*R+.023095*R2-.010325*R3+.002187*R4
+   -.0002208*R**5 + .00000840B*R**6
       BF=-.122149+.313445*R-.153979*R2+.036192*R3-.004015*R4
+   +.0001646*R**5
       CF=2.1958-7.7564*R+10.8376*R2-7.2495*R3+2.344*R4
+          -.2954*R**5
        GOTO 41
C
C:  CASE 4. SIMPLY SUPPORTED ON SHORT SIDES, FIXED ON LONG SIDES
40     NX=3
        AF=-.002765+.008652*R-.005698*R2+.001829*R3-.0002559*R4
+   +.00001739*R**5
        BF=-.060320+.256515*R-.175649*R2+.057928*R3-.009227*R4
+    +.000569*R**5
```

PROGRAM RCWALL (CONTINUED)

```
      CF=5.8987*R-1.6669-7.9398*R2+5.3142*R3-1.7623*R4+.2313*R**5
C
 41   IF(R.GT.2.0)CF=1.0/12.0
      IF(PV.EQ.0)RETURN
      ARATIO=AF/ASS$  BRATIO=BF/BSS
      BFO=BF$ CFO=CF
      GOTO 220
C
 50   IF(PV.NE.0)GOTO 300
C:  CASE 5. ONE-WAY SIMPLY SUPPORTED WALL
      ASS=5.0/384.0
       BSS=0.125
       GOTO(270,270,270,270,270,60,70),ICASE
C
C:  CASE 6. ONE-WAY FIXED END WALL
 60    AF=1.0/384.0
       BF=1.0/12.0
      CF=1.0/12.0
      NX=3
      RETURN
C
C:  CASE 7. ONE-WAY PROPPED CANTILEVER WALL
 70    AF=1.0/185.0
       BF=0.125
      CF=0.125
      NX=3
      RETURN
C
 200  IF(ICASE.GT.4)GOTO 300
C:  DETERMINE ELASTIC DEFLECTION AND MOMENT COEFFICIENT FOR
C:  TWO-WAY WALL WITH INPLANE FORCES
 220  PI=3.14159165
       NU=0.3
      PE=4.0*PI*PI*E*I/(ZLV*ZLV*(1.0-NU*NU))
       BV=0
 230   AV=0
       PPE=PV/PE
       TERM6=4.0*PI*PI*R*SQRT(PPE)
C
C:  SERIES SOLUTION USED TO DETERMINE COEFFICIENTS
       DO 250 M=1,7,2
       MPR=M*PI*R
       MPRSQ=MPR**2
       GMSQ=MPRSQ+2.0*MPR*PI*SQRT(PPE)
       EMSQ=MPRSQ-2.0*MPR*PI*SQRT(PPE)
       TERM5=M*MPRSQ*(MPRSQ-4.0*PI*PI*PPE)
       COSHGM2=0.5*(EXP(0.5*SQRT(GMSQ))+EXP(-0.5*SQRT(GMSQ)))
       IF(EMSQ.LT.0)GOTO 240
       COSHEM2=0.5*(EXP(0.5*SQRT(EMSQ))+EXP(-0.5*SQRT(EMSQ)))
       GOTO 245
 240   COSHEM2=COS(0.5*SQRT(-EMSQ))
 245   AV=AV+(1.0-EMSQ/COSHGM2-GMSQ/COSHEM2)/(M*TERM6))
+  *(-1)**((M-1)/  /TERM5
       BV=BV+(MPRSQ*(GMSQ*(NU*EMSQ-MPRSQ)/COSHEM2-EMSQ*(NU*GMSQ
+        -MPRSQ)/COSHGM2)/(M*TERM6))*(-1)**((M-1)/2)/TERM5
 250   CONTINUE
C
C:  CASE 1
      AVSS=AV*(1.0-NU*NU)*R4*4.0/PI
       BVSS=BV*R2*4.0/PI
       IF(ICASE.EQ.1)GOTO 260
C
C:  CASES 2, 3, AND 4
      AVF=AVSS*ARATIO
      BVF=BVSS*BRATIO
      CF=CFO*BVF/BFO
 258  AF=AVF
       BF=BVF
 260  ASS=AVSS
       BSS=BVSS
 270  RETURN
C
C:  ONE-WAY WALLS
 300  EIPV=E*I/PV
```

PROGRAM RCWALL (CONTINUED)

```
      U=ZLV/SQRT(EIPV)
      U2=0.5*U
      TERM1=1.0/COS(U2)-1.0
C
C:  CASE 5. ONE-WAY SIMPLY SUPPORTED WALL
      BSS=TERM1/U**2
      ASS=(BSS-0.125)/U**2
      GOTO(270,270,270,270,270,310,320),ICASE
C
C:  CASE 6. ONE-WAY FIXED END WALL
 310  NX=3
      BF=(1.0-U2/TAN(U2))/U**2
      AF=-BF*BSS+ASS
      RETURN
C
C:  CASE 7. ONE-WAY PROPPED CANTILEVER WALL
 320 NX=3
      BF=TAN(U)*(TAN(U2)-U2)/(U*(TAN(U)-U))
      AF=(BF*(0.5*SIN(U2)/TAN(U)-COS(U2))-(SIN(U2)/TAN(U)
+        -COS(U2)-SIN(U2)/SIN(U)+0.125*U*U+1.0)/U**2)/U**2
      RETURN
 999 END
 SUBROUTINE TRANS (B,ZLV,ZLH,ICASE,KRAK,ZKLMSE,ZKLMFE,ZKLMP,VL1S,
+  VL2S,VS1S,VS2S,VL1F,VL2F,VS1F,VS2F,VL1P,VL2P,VS1P,VS2P)
C
C: THIS SUBROUTINE DETERMINES LOAD AND MASS TRANSFORMATION FACTORS
C: AND DYNAMIC REACTION COEFFICIENTS FOR TWO-WAY WALLS.
C
C: DETERMINE LOAD AND MASS TRANSFORMATION FACTORS
      B2=B*B
      B3=B*B2
      B4=B2*B2
      B5=B2*B3
      B6=B3*B3
C
C: CASES 1 & 5 -- ELASTIC RANGE
 330  ZKMSE1=20.48*B3*(1./12.-B2/7.5+B3/21+B4/14-B5/18+B6/90)
      ZKMSE2=0.5038-0.7066*B
      ZKLSE1=6.4*B2*(1./6.-B2/10.+B3/30.)
      ZKLSE2=0.64-0.8134*B
      BARS1=B*(1./12.-B2/15.+B3/42.)/(1./6.-B2/10.+B3/30.)
      BARS2=(0.127083-0.184524*B)/(0.4-0.508333*B)
      ZKMSE=ZKMSE1+ZKMSE2
      ZKLSE=ZKLSE1+ZKLSE2
      IF(KRAK.EQ.1)GOTO 335
C: CRACK PATTERN A
      CVS=0.5*B
      CVL=0.5*(1.0-B)
      XP=ZLH*B/3.0
      XBARS=BARS1*ZLH
      ZP=ZLV*(1.0-4.0*B/3.0)/(4.0*(1.0-B))
      ZBARS=BARS2*ZLV
      XBARP=0.5*B*ZLH
      ZBARP=ZLV*(1./24.-B/16.)/(1./8.-B/6.)
      GOTO 338
C: CRACK PATTERN B
 335  CVS=0.5*(1.0-B)
      CVL=0.5*B
      XP=ZLH*(1.0-4.0*B/3.0)/(4.0*(1.0-B))
      XBARS=BARS2*ZLH
      ZP=ZLV*B/3.0
      ZBARS=BARS1*ZLV
      XBARP=ZLH*(1./24.-B/16.)/(1./8.-B/6.)
      ZBARP=0.5*B*ZLV
 338    LMSE=ZKMSE/ZKLSE
      GOTO(390,340,350,360,390,340,470),ICASE
C
C: CASES 2, 3, & 4 -- ELASTIC RANGE
 350 IF(KRAK.EQ.1)GOTO 365
      GOTO 340
 360  IF(KRAK.EQ.0)GOTO 365
C: CASES 2A  2B, 3A, 4B, & 6
 340  ZKM-. 512.0*B5*(1.0/30.-B/10.5+3.*B2/28.-B3/18.+B4/90.)
      ZKLE. 32.0*B3*(1./12.-B/10.+B2/30.)
```

```
          BARF1=B*(.05-B/15.+B2/42.)/(1./12.-B/10.+B2/30.)
          GOTO(370,365,370,370,370,365),ICASE
C: CASES 2A, 2B, 3B, 4A, & 6
     365  ZKMFE2=0.4065-0.6144*B
          ZKLFE2=0.5344-0.7328*B
          BARF2=(.091667-.138095*B)/(.266667-.366667*B)
          GOTO(375,368,375,375,375,368),ICASE
C: CASES 2A & 2B
     368  ZKMFE=ZKMFE1+ZKMFE2
          ZKLFE=ZKLFE1+ZKLFE2
          GOTO 380
C:  CASES 3A & 4B
     370  ZKMFE=ZKMFE1+ZKMSE2
          ZKLFE=ZKLFE1+ZKLSE2
          GOTO 380
C: CASES 3B, 4A, & 6
     375  ZKMFE=ZKMSE1+ZKMFE2
          ZKLFE=ZKLSE1+ZKLFE2
     380  ZKLMFE=ZKMFE/ZKLFE
          GOTO 390
C: CASE 7
     470  ZKLMFE=0.78
C
C: ALL CASES -- PLASTIC RANGE
     390  ZKMP=(1.0-B)/3.0
          ZKLP=0.5-B/3.0
          ZKLMP=ZKMP/ZKLP
C
C
C: DETERMINE DYNAMIC REACTION COEFFICIENTS FOR SHORT (VS) AND
C: LONG (VL) EDGES
C
          IF(ICASE.LT.5)GOTO 395
          XBARS=1E-10$ BARF1=1E-10$ XBARP=1E-10
     395  CONTINUE
          GOTO(450,400,400,420,450,400,445),ICASE
     400  IF(KRAK.EQ.1)GOTO 410
          XBARF=BARF1*ZLH
          IF(ICASE.EQ.3)GOTO 430
     405  ZBARF=BARF2*ZLV
          GOTO 440
     410  XBARF=BARF2*ZLH
          IF(ICASE.EQ.3)GOTO 435
     415  ZBARF=BARF1*ZLV
          GOTO 440
     420  IF(KRAK.EQ.1)GOTO 425
          XBARF=BARS1*ZLH
          GOTO 405
     425  XBARF=BARS2*ZLH
          GOTO 415
     430  ZBARF=BARS2*ZLV
          GOTO 440
     435  ZBARF=BARS1*ZLV
     440  CONTINUE
C
C: CASES 2, 3, 4, & 6 -- ELASTIC RANGE
          VS1F=CVS*(1.0-XP/XBARF)
          VS2F=CVS*(XP/XBARF)
          VL1F=CVL*(1.0-ZP/ZBARF)
          VL2F=CVL*(ZP/ZBARF)
          GOTO 450
C
C: CASE 7 -- ELASTIC RANGE
     445  VS1F=0
          VL1F=0.459
          VL2F=0.165
C
C: CASE 1 & 5 -- ELASTIC RANGE
     450  VS1S=CVS*(1.0-XP/XBARS)
          VS2S=CVS*(XP/XBARS)
          VL1S=CVL*(1.0-ZP/ZBARS)
          VL2S=CVL*(ZP/ZBARS)
C
C: ALL CASES -- PLASTIC RANGE
```

PROGRAM RCWALL (CONTINUED)

```
  460   VS1P=CVS*(1.0-XP/XBARP)
        VS2P=CVS*(XP/XBARP)
        VL1P=CVL*(1.0-ZP/ZBARP)
        VL2P=CVL*(ZP/ZBARP)
        RETURN
        END
        SUBROUTINE RANDOM (IENTRY)
C THIS SUBROUTINE INPUTS MEAN AND STANDARD DEVIATIONS FOR RANDOM
C VARIABLES;  GENERATES RANDOM VALUES;  AND CONTROLS REQUIRED
C NUMBER OF CASES TO BE RUN;  AND OUTPUTS FINAL RESULTS AND SUMMARY
C
 COMMON K,WALL,KINC,LOTYPE,KRF,KRAND,TIME,I,Y(100),Q,QU,YU,YFAIL,
+  ZLV,ZLH,TW,PV,FPC,FDY,ICASE,NOBAR,AS(4),APS(4),D(4),DP(4),FDC,
+  EC,ES,R,ALP,ALP2,AREA,ZMASS,QULT,VFAIL,ZKLM,VH1,VH2,VV1,VV2,
+  Z,PO,CO,LOC,S,ZLEN,CD,PSO,PDO,PR,PEXT,PC,TC,TO,DELAY,
+  NWIN,RHOO,V3,L1,AA(8,2),NN(8),AFRONT,ASIDE,G,G2,G3,G4,PP2,DT
      COMMON /RAND/ TIMEC
      DIMENSION CHI25(7),CHI975(7),TDIST(7)
C
C VALUES FOR 97.5%  (F=19,24,29,34,39,44,49)
       DATA CHI25/.4639,.5167,.5533,.5825,.6065,.6267,.6440/
       DATA CHI975/1.7295,1.6402,1.5766,1.5284,1.4903,1.4591,1.4331/
       DATA TDIST/2.093,2.064,2.045,2.032,2.022,2.014,2.010/
C
       GOTO(5,50,70),IENTRY
     5 XDUMMY=XNORM1(-1.0,0.0,1.0)
C INITIALIZE RANDOM NUMBER GENERATOR
       PRINT,/,*INPUT NRAND*,
       READ,NRAND
       DO 47 I=1,NRAND
       XDUMMY=XNORM1(0.0,0.0,1.0)
   47 CONTINUE
       INDEX=0$  SPSO=0$  SSPSO=0
       ICHECK=20
C
C  INPUT MEAN AND STANDARD DEVIATION FOR RANDOM VARIABLES
       PRINT 87
       READ,SMEAN,SSD
C  REINFORCED CONCRETE WALLS
  30   PRINT 86
       READ,FDYMEAN,FDYSD
       PRINT 96
       RETURN
C
C  GENERATE RANDOM VALUES
  50   FDY=XNORM1(0.0,FDYMEAN,FDYSD)
       IF(FDY.LE.0)GOTO 50
       IF(SMEAN.EQ.0)GOTO 65
  60   S=XNORM1(0.0,SMEAN,SSD)
       IF(S.LE.0)GOTO 60
  65   INDEX=INDEX+1
       RETURN
C SUM VALUES OF PSO AND PSO**2 FOR USE IN STATISTICAL ANALYSIS
  70   SPSO=SPSO+PSO
       SSPSO=SSPSO+PSO*PSO
C
C OUTPUT FINAL RESULTS
  76   PRINT 92,FDY,S,PSO,TIMEC
  80   IF(INDEX.LT.ICHECK)RETURN
C
C DETERMINE MEAN, STANDARD DEVIATION, AND STANDARD ERROR FOR PSO
       ZNO=INDEX
       ZMEAN=SPSO/ZNO
       SD=SQRT((SSPSO-ZNO*ZMEAN*ZMEAN)/ZNO)
       STDERR=SD/(SQRT(ZNO-1))
C CHECK IF MAXIMUM OF 50 PSO SAMPLES OBTAINED
       IF(INDEX.EQ.50)GOTO 62
C CHECK IF 95% CONFIDENCE INTERVAL FOR MEAN PSO VALUE IS
       IF(STDERR*TDIST((INDEX-15)/5)/ZMEAN.GT.0.10)GOTO 61
C
C CONFIDENCE INTERVAL IS WITHIN 10% -- DETERMINE UPPER LIMIT OF
C 95% CONFIDENCE INTERVAL FOR STANDARD DEVIATION
C PROBABILITY VALUE AND ITS 95% CONFIDENCE INTERVAL UPPER LIMIT
    62 SDU=SD/(SQRT(CHI25((INDEX-15)/5)))
```

PROGRAM RCWALL (CONCLUDED)

```
C CHECK IF MAXIMUM OF 50 PSI SAMPLES OBTAINED
      IF(INDEX.EQ.50)GOTO 53
C CHECK IF UPPER VALUE OF 95% CONFIDENCE INTERVAL FOR STANDARD
C DEVIATION IS WITHIN 0.10*MEAN OF THE STANDARD DEVIATION
      IF(((SDU-SD)/ZMEAN).GT.0.10)GOTO 61
C
C 95% CONFIDENCE INTERVAL IS WITHIN 10% FOR BOTH MEAN AND 90%
C PROBABILITY VALUE -- THEREFORE SUFFICIENT SAMPLES OBTAINED
C DETERMINE 95% CONFIDENCE INTERVALS FOR MEAN, STANDARD DEVIATION
,
C AND 10% AND 90% PROBABILITY VALUES
   53 ZMEANL=ZMEAN-STDERR*TDIST((INDEX-15)/5)
      ZMEANU=ZMEAN+STDERR*TDIST((INDEX-15)/5)
      SDL=SD/(SQRT(CHI975((INDEX-15)/5)))
      P10=ZMEAN-1.282*SD
      P10L=ZMEAN-1.282*SDU
      P10U=ZMEAN-1.282*SDL
      P90=ZMEAN+1.282*SD
      P90L=ZMEAN+1.282*SDL
      P90=ZMEAN+1.282*SD
      P90U=ZMEAN+1.282*SDU
      P90U=ZMEAN+1.282*SDU
C
C OUTPUT STATISTICAL PARAMETERS OF INCIPIENT COLLAPSE PRESSURE
      PRINT 100,ZMEAN,ZMEANL,ZMEANU,SD,SDL,SDU,P10,P10L,P10U,
+        P90,P90L,P90U
      PRINT 105,INDEX,STDERR
      GOTO 999
C
C 95% CONFIDENCE INTERVAL IS NOT WITHIN 10% FOR BOTH MEAN AND 90
R
C VALUES -- THEREFORE OBTAIN 5 ADDITIONAL SAMPLES
   61 ICHECK=ICHECK+5
      RETURN
C
   86 FORMAT(/*INPUT MEAN AND STANDARD DEVIATION FOR FDY*,*)
   87 FORMAT(/*INPUT MEAN AND STANDARD DEVIATION FOR S*,*)
   90 FORMAT(F9.2,F11.2,F10.2,F14.3)
   92 FORMAT(F9.1,F11.2,F10.2,F14.3)
   96 FORMAT(///,5X,*FDY*,9X,*S*,8X,*PSO*,6X,*COLLAPSE TIME*)
  100 FORMAT(///,11X,*STATISTICAL PROPERTIES OF INCIPIENT PSO*,
+        //,39X,*95% CONFIDENCE LIMITS*,/,7X,*ITEM*,18X,
+        *VALUE        LOWER         UPPER*,//,* MEAN*,F29.2,
+        2F12.2,//,* STANDARD DEVIATION*,F15.2,2F12.2,//,
+        * 10% PROBABILITY VALUE*,3F12.2,//,
+        * 90% PROBABILITY VALUE*,3F12.2)
  105 FORMAT(//,5X,*NUMBER OF OBSERVATIONS =*,I3,/,5X,
+        *STANDARD ERROR =*,F5.2)
C
  999 STOP$ END
FUNCTION XNORM1(X,A,B)
IF(X)10,20,20
10 XO=RANF(-1.0)
20 X1=RANF(0.0)
X2=RANF(0.0)
Y=SQRT(-2.0*ALOG(X1))*(COS(6.283184*X2))
XNORM1=A+Y*B
RETURN
END
```

RCSLAB

Reinforced Concrete Slab

PROGRAM RCSLAB

```
00100 PROGRAM RCSLAB1 (INPUT,OUTPUT,TAPE1)
00105 CALL RETR(7HRCSLAB2,7HRCSLAB2)
00110C * THIS PORTION OF THE PROGRAM INPUTS THE REQUIRED ELEMENT AND
00115C:  LOAD DATA AND INITIALIZES CERTAIN PARAMETERS *
00120C
00150 COMMON KINC,LDTYPE,KRF,KRAND,TIME,I,Y(80),Q,QU,YU,YFAIL,
00152+  ZLS,ZLL,HS,PV,FPC,FDY,ICASE,NBBAR,AS(4),APS(4),D(4),DP(4),FDC,
00154+  EC,ES,R,ALP,ALP2,AREA,ZMASS,VFAIL,ZKLM,VL1,VL2,VS1,VS2,
00155+  MEMB,ASCL,ASCS,VCL,VCS,ASS,BSS,AF,BF,CF,NK,
00156+  W,PO,CO,LOC,S,ZLEN,CD,PSO,PDO,PR,PEXT,PC,TC,TO,DELAY,
00158+  NMIN,RHOO,VO,L1,AA(8,2),NN(8),AFRNT,ASIDE,G,G2,G3,G4,PH2,DT
00160 LOGICAL L1
00165C
00170C:  READ TITLE AND CONTROL PARAMETERS
00175 PRINT 67
00180 READ 68,TITLE
00185 PRINT 85
00190 READ,KINC,LDTYPE,KRF,KRAND
00195 DELAY=0
00200 YFAIL=1E10
00205 67 FORMAT(/*INPUT TITLE*,+)
00210 68 FORMAT(A39)
00215 85 FORMAT(/>INPUT KINC,LDTYPE,KRF,KRAND(1=RANDOM)*,+)
00234C
00236 4 DO 5 I=1,4
00238 AS(I)=0
00240 5 CONTINUE
00242C
00244C * INPUT AND ECHO SLAB AND REINFORCEMENT PROPERTIES *
00246 PRINT 615
00248 READ,ZLS,ZLL,HS,FPC,FDY,ICASE,NBBAR
00250 FDC=1.25*FPC
00252 EC=57619.0*SQRT(FPC)
00254 ES=29E6
00256 AREA=ZLS*ZLL
00258 ECKIP=EC/1000$ ESKIP=ES/1000
00260 R=ZLL/ZLS
00262 ALP=1.0/R$ ALP2=ALP*ALP
00264 IF(ICASE.LE.4)GOTO 11
00266 R=0$ ALP=0$ ALP2=0
00268 11 PRINT 670
00270 DO 8 I=1,4
00272 PRINT 625,I
00274 READ,AS(I),D(I),APS(I),DP(I)
00276 IF(I.NE.1)GOTO 3
00278 GOTO(9,9,8,8,9,7,7),ICASE
00280 3 GOTO(9,8,7,6,9,9,9),ICASE
00282 6 IF(I.EQ.3)GOTO 9
00284 GOTO 8
00286 7 I=I+1
00288 8 CONTINUE
00290 9 CONTINUE
00292 PRINT 711
00294 READ,MEMB
00296 IF(MEMB.NE.1)GOTO 15
00300 IF(ICASE.GT.4)GOTO 13
00302 PRINT 701
00304 READ,ASCS,ASCL
00306 ASCL=ASCL/12.0
00308 GOTO 16
00310 13 PRINT 706
00312 READ,ASCS
00314 ASCL=0.0
00316 16 ASCS=ASCS/12.0
00318C
00320C: ***************************************************
00322C: * DETERMINE DEFLECTION AND MOMENT COEFFICIENTS *
00324C: ***************************************************
00326C
00328 15 ZMASS=150.0*AREA*HS/(386.07*1728.0)
00330 PRINT 620,ICASE,ZLS,ZLL,HS,FPC,FDC,ECKIP,FDY,ESKIP
00332 PRINT 630
00334 DO 110 I=1,4
00336 IF(AS(I).EQ.0)GOTO 110
```

```
      P=AS(I)/(12.0*D(I)) $ PP=APS(I)/(12.0*D(I))
      PRINT 645,I,AS(I),P,D(I),APS(I),PP,DP(I)
C  CHANGE UNITS OF REINFORCEMENT FROM SQ.IN./FT TO ..   ..
      AS(I)=AS(I)/12.0 $ APS(I)=APS(I)/12.0
  110 CONTINUE
C
      VCL=2.24*SQRT(FPC)/(1-2*D(1)/ZLS)+3000*(AS(1)/D(1))/(1-D(1)/ZLS)
      VCLMAX=3.5*SQRT(FPC)/(1.0-2.0*D(1)/ZLS)
      IF(VCL.GT.VCLMAX)VCL=VCLMAX
      IF(ICASE.GT.4)GOTO 30
      VCS=2.24*SQRT(FPC)/(1-2*D(2)/ZLL)+3000*(AS(2)/D(2))/(1-D(2)/ZLL)
      VCSMAX=3.5*SQRT(FPC)/(1.0-2.0*D(2)/ZLL)
      IF(VCS.GT.VCSMAX)VCS=VCSMAX
      PRINT 680,VCL,VCS
      GOTO 35
   30 PRINT 685,VCL
   35 CONTINUE
      CALL CSEZ(ICASE,H,ASS,BSS,AF,BF,O,O,ZLS,ZLL,PV,NX,CF,EC,1)
  615 FORMAT(/*INPUT LS,LL,HS,F'C,FDY,ICASE,NOBAR*,/)
  625 FORMAT(*INPUT AS,D,A'S,D' FOR SECTION*,I2,/)
  620 FORMAT(//*PROPERTIES OF REINFORCED CONCRETE SLAB ---*,
     +       * SUPPORT TYPE NO.*,I2,/,
     +       * LS =*,F6.1,* IN.      LL =*,F6.1,* IN.*,6X,*HS =*,
     +       F6.1,* IN.*,/,* F'C =*,F7.1,* PSI   FDC =*,F7.1,
     +       * PSI*,5X,*EC =*,F7.1,* KSI*,/,* FDY =*,F4.1,* PSI*,
     +       3X,*ES =*,F5.1,* KSI*)
  635 FORMAT(//*REINFORCEMENT VALUES*/* SECTION    AS     (P)*,
     +       9X,*D*,8X,*A'S    (P')*,8X,*D'*,/,3X,*(SQ IN./FT)*,10X,
     +       *(IN.)  (SQ IN./FT)*,10X,*(IN.)*)
  640 FORMAT(I5,F11.4,* (*,F6.4,*)*,F9.3,F10.3,* (*,F6.4,*)*,F9.3)
  675 FORMAT(1H )
  680 FORMAT(/*VCL =*,F6.1,* PSI     VCS =*,F6.1,* PSI*)
  685 FORMAT(/*VCL =*,F6.1,* PSI*)
  701 FORMAT(/*INPUT CONTINUOUS REINFORCEMENT PARALLEL TO *,
     +       *SHORT AND*,/,*LONG SPANS (SQ IN./FT)*,/)
  705 FORMAT(/*INPUT CONTINUOUS REINFORCEMENT PARALLEL TO *,
     +       *SHORT SPAN (SQ IN./FT)*,/)
  711 FORMAT(/*IS TENSILE MEMBRANE TO BE INCLUDED (0=NO)*,
     +       *1=YES)*,/)
C
C  INPUT LOAD PARAMETERS
      IF(LOTYPE.EQ.5)GOTO 20
C  LOCATION 1.  FRONT FACE LOADING (USED IN ROOM-FILLING PROCEDURE
  100 W=1000.0 $ P0=14.7 $ C0=1120.0
      IF(KPF.NE.1)GOTO 102
      LOC=1
      IF(KRAN.EQ.1)GOTO 106
      PRINT 603
      READ,S
      GOTO 105
C  LOCATION 2.  TOP FACE LOADING
  102 CO=0 $ LOC=2
      ZLEN=ZLS/12.0
  105 IF(KINC.EQ.1)GOTO 106
      PRINT 610
      READ,PS3
      PR=2.0*PS3*(7.0*P0+4.0*PS3)/(7.0*P0+PS3)
  600 FORMAT(/*INPUT S*,/)
  610 FORMAT(/*INPUT PS3*,/)
C
C * INPUT ROOM-FILLING PARAMETERS *
  106 IF(KRF.EQ.0)GOTO 20
   10 PRINT 700
      RHOO=0.076 $ L1=.FALSE.
      DELAY=1E10
      READ,NWIN,VO
      AT=0$ AFRONT=0$ ASIDE=0
      DO 14 I=1,NWIN
      PRINT 710,I
      READ,AA(I,1),NN(I),AA(I,2)
      AA(I,2)=AA(I,2)/1000.0
      AT=AT+AA(I,1)
      M=NN(I)$ GOTO(12,14,14),M
   12 AFRONT=AFRONT+AA(I,1)
```

PROGRAM RCSLAB (CONTINUED)

```
      GOTO 15
 14   ASIDE=ASIDE+AA(I,1)
 15   IF(AA(I,2).LT.DELAY)DELAY=AA(I,2)
      AFRONT=AFRONT/AT$ ASIDE=ASIDE/AT
 700  FORMAT(/*INPUT NWIN AND ROOM VOLUME (CF)*,*)
 710  FORMAT(/*INPUT AREA (SQ FT),LOCATION CODE & DELAY(MSEC)*
+        * FOR WINDOW*,I2,*)
      G=1.4 $ G2=1./G $ G3=1.-G2 $ G4=2./G3 $ G5=G+1.
      PP2=.1912
      C=SQRT(G*P0*32.*144./RHO0)
      TAU=2.*(V3**(1./3.))/C
      DT=TAU/4.0
C
 20   CONTINUE
 25   CALL CHAIN(RCSLAB2)
 99   STOP
      END
      SUBROUTINE COEF(ICASE,R,ASS,BSS,AF,BF,I,ZLV,ZLH,PV,NX,CF,
+        E,IENTRY)
C:  THIS SUBROUTINE DETERMINES MOMENT AND DEFLECTION COEFFICIENTS
C:  FOR ONE-WAY (CASES 5-7) AND TWO-WAY (CASES 1-4) ELEMENTS
C
         REAL I,MPR,MPR ),NU
         NX=1
          IF(ICASE.GT.4)GOTO 50
C
          R2=R*R
          R3=R*R2
          R4=R2*R2
          ASS=-.007030+.013890*R-.003456*R2+.000286*R3
          BSS=-.058332+.139314*R-.035609*R2+.003016*R3
          GOTO(41,20,30,40),ICASE
C
C:  CASE 2. FIXED ON FOUR SIDES
 20    NX=3
          AF=-.003430+.007327*R-.003365*R2+.0006646*R3-.00004766*R4
          BF=-.101150+.260875*R-.138982*R2+.034677*R3-.004016*R4
+     +.000170*R**5
         CF=-.1674+.3554*R-.1714*R2+.0286*R3
          GOTO 41
C
C:  CASE 3. FIXED ON SHORT SIDES, SIMPLY SUPPORTED ON LONG SIDES
 30     NX=4
          AF=.004513-.017525*R+.023095*R2-.010325*R3+.002187*R4
+    -.0002208*R**5 + .000008408*R**6
          BF=-.122149+.313445*R-.153979*R2+.036192*R3-.004015*R4
+    +.0001646*R**5
         CF=2.1958-7.7564*R+10.8376*R2-7.2495*R3+2.344*R4
+           -.2954*R**5
          GOTO 41
C
C:  CASE 4. SIMPLY SUPPORTED ON SHORT SIDES, FIXED ON LONG SIDES
 40    NX=3
          AF=-.002765+.006652*R-.005498*R2+.001829*R3-.0002859*R4
+    +.00001739*R**5
          BF=-.060320+.256515*R-.175648*R2+.057928*R3-.009227*R4
+    +.000569*R**5
      CF=5.8987*R-1.6669-7.9398*R2+5.3142*R3-1.7623*R4+.2313*R**5
C
 41    IF(R.GT.2.0)CF=1.0/12.0
         RETURN
C
 50    CONTINUE
C:  CASE 5. ONE-WAY, SIMPLY SUPPORTED
         ASS=5.0/384.0
          BSS=0.125
          GOTO(270,270,270,270,270,60,70),ICASE
C:  CASE 6. ONE-WAY, FIXED ENDS
C:  CASE 6. ONE-WAY FIXED END WALL
 60     AF=1.0/384.0
          BF=1.0/12.0
         CF=1.0/12.0
         NX=3
         RETURN
```

PROGRAM RCSLAB (CONTINUED)

```
C
C:  CASE 7. ØNE-WAY, PRØPPED CANTILEVER
 70     AF=1.0/185.0
        BF=0.125
        CF=0.125
        NX=3
 270    RETURN
        END
SEGMENT RCSLAB2 (INPUT,ØUTPUT,TAPE1)
C  THIS SEGMENT CALCULATES THE RESISTANCE FUNCTIØN AND
C  LØADING AND SØLVES THE DYNAMIC EQUATIØNS ØF MØTIØN
C
CØMMØN KINC,LDTYPE,KRF,KRAND,TIME,I,Y(80),Q,QU,YU,YFAIL,
+   ZLS,ZLL,HS,PV,FPC,FPY,ICASE,NØBAR,AS(4),APS(4),D(4),DP(4),FDC,
+   EC,ES,R,ALP,ALP2,AREA,ZMASS,VFAIL,ZKLM,VL1,VL2,VS1,VS2,
+   ME4B,ASCL,ASCS,VCL,VCS,ASS,BSS,AF,BF,CF,NX,
+   W,PØ,CØ,LØC,S,ZLEN,CD,PSØ,PDØ,PF,PEXT,PC,TC,TØ,DELAY,
+   NWIN,RHØØ,V3,L1,AA(8,2),NN(8),AFRØNT,ASIDE,G,G2,G3,G4,PP2,DT
CØMMØN /SAR/ SAREAS,SAREAL
CØMMØN /RAND/ TIMEC
DIMENSIØN A(80),V(80),T(80),VS(80),VL(80),PN(80)
C
IF(KINC.NE.1.ØR.LDTYPE.EQ.5)CALL FØRCE(1)
 14     IF(KRAND.NE.1)GØTØ 35
        CALL FØRCE(4)
        CALL RANDØM(1)
 34     CALL RANDØM(2)
 35     CALL RESIST(3)
C
C: MINIMUM, MAXIMUM, AND STARTING VALUES ARE DETERMINED FØR CASES
C: WHERE THE LØAD CAUSING INCIPIENT CØLLAPSE IS TØ BE FØUND
 13     IF(KINC.EQ.0)GØTØ 23
         PF=QU
         PFMAX=0
         PFMIN=PF/2.0
        GØTØ 20
 16      PF=(PFMIN+PFMAX)/2.0
 20     CALL FØRCE(2)
 23     IF(KRF.EQ.0)GØTØ 24
        CALL FILL(PINT,2)
C
C: INITIALIZE VALUES FØR BETA METHØD (BETA = 1/6) AND CØMPUTE VALU
C: FØR FIRST TIME INTERVAL ASSUMING ELEMENT INITIALLY AT REST
 24      I=1
         TIME=0
        T(1)=0$ V(1)=0$ Y(1)=0
         DELTA=0.001
        IF(KRF.NE.1)GØTØ 30
 27     IF(TIME.GE.(DELAY-.00001))GØTØ 30
        TIME=TIME+DELTA
CALL FILL(PINT,3)
        GØTØ 27
IF(Y(I).GE.YFAIL)PRINT 71,T(I),Y(I)
 30     CALL RESIST(2)
        A(1)=0.0 $ VS(1)=0.0 $ VL(1)=0.0
        T(1)=TIME
C
C: PRØCEDURE FØR ALL SUBSEQUENT TIME INTERVALS
 1       I=I+1
IF(I.LT.81)GØTØ 11
        PRINT 98,TIME
 98 FØRMAT(/*I=81:  TIME =*,F6.3,*  FAILURE ASSUMED TØ NØT ØCCUR*)
        GØTØ 6
 11     TIME=TIME+DELTA
         T(I)=TIME
         A(I)=A(I-1)
        IF(KRF.NE.0)GØTØ 10
        CALL FØRCE(3)
        PN(I)=PEXT
        GØTØ 2
 10     CALL FILL(PINT,3)
        PN(I)=PINT
 2 CØNTINUE
        DØ 8 JJ=1,10
```

PROGRAM RCSLAB (CONTINUED)

```
      Y(I)=Y(I-1)+DELTA*V(I-1)+DELTA*DELTA*(A(I-1)/3.+A(I)/6.)
      CALL RESIST(2)
4     ANEW=AREA*(PN(I)-Q)/(ZMASS*Z(L))
      ADELTA=ANEW-A(I)
      A(I)=ANEW
IF(ANEW.EQ.0)PRINT,*1985*,TIME,PN(I),Q,ZMASS,Z(L),Y(I),A(I-1)
      IF(ABS(ADELTA/(ANEW+1E-10)).LT.0.01)GOTO 9
8     CONTINUE
      A(I)=ANEW-ADELTA/2.0
PRINT 80,TIME,PF,A(I),Y(I)
9     CONTINUE
      Y(I)=Y(I-1)+DELTA*V(I-1)+DELTA*DELTA*(A(I-1)/3.+A(I)/6.)
      V(I)=V(I-1)+DELTA*(A(I)+A(I-1))/2.0
      VS(I)=AREA*(VS1*PN(I)+VS2*Q)
      VL(I)=AREA*(VL1*PN(I)+VL2*Q)
C
C: CHECK FOR MAXIMUM DEFLECTION OR FAILURE OF WALL
C:   IF MAXIMUM DEFLECTION  ACHED, ELEMENT DID NOT FAIL
      IF(Y(I).LE.Y(I-1).A   PN(I).LE.PN(I-1))GOTO 6
      IF(Y(I).LT.0)GOTO 6
      IF(TIME-DELAY.GE.0.010)DELTA=0.002
      IF(TIME-DELAY.GE.0.020)DELTA=0.005
      IF(TIME-DELAY.GE.0.100)DELTA=0.010
      IF(TIME-DELAY.GE.0.500)DELTA=0.050
C  IF FAILURE DEFLECTION REACHED, WALL FAILED
      IF(Y(I).GE.Y  IL)GOTO 7
      GOTO 1
C
C: INTERVAL HALVING PROCEDURE TO DETERMINE LOAD CAUSING INCIPIENT
C: COLLAPSE FOR CASES WHERE DESIRED
C:   ELEMENT DID NOT FAIL -- SET PFMIN TO PF
6     CONTINUE
      IF(KINC.EQ.0)GOTO 18
36    PFMIN=PF
IF(PFMAX.GT.0)GOTO 16
PF=2.0*PF
      GOTO 20
C:   ELEMENT FAILED -- SET PFMAX TO PF
7     CONTINUE
      TIMEC=TIME
      IF(KINC.EQ.0)GOTO 18
37    PFMAX=PF
C:   CHECK TO SEE IF LOAD RANGE IS WITHIN DESIRED ACCURACY
17    IF((PFMAX-PFMIN)/PFMIN.GT.0.01)GOTO 16
      IF(KRAND.NE.1)GOTO 18
      CALL RANDOM(3)
      GOTO 34
C
C: OUTPUT DATA INCLUDES THE MAXIMUM DEFLECTION AND TIME OF
C: OCCURANCE FOR A NON-FAILING WALL OR THE TIME AND VELOCITY
C: AT COLLAPSE FOR A FAILING ELEMENT.  OPTIONAL OUTPUT
C: ENTIRE BEHAVIOR TIME-HISTORY OF THE WALL.
C
C: OUTPUT LOAD DATA
18    CALL FORCE(4)
C
C: OUTPUT FINAL RESULTS
IF(Y(I).LT.YFAIL)PRINT 70,Y(I),T(I)
IF(Y(I).GE.YFAIL)PRINT 71,T(I),V(I)
C
PRINT 90
READ,KFILE
IF(KFILE.EQ.0)GOTO 40
PRINT 95
READ,NAMEF
CALL PFUR(3HRET,1,NAMEF)
WRITE(1,)SAREAS,SAREAL,HS
WRITE(1,)I
WRITE(1,)(T(J),PN(J),VS(J),VL(J), J=1,I)
CALL PFUR(3HREP,1,NAMEF)
40 CONTINUE
C  CHECK TO SEE IF ENTIRE TIME-HISTORY IS DESIRED
PRINT 72
READ,M
```

```
      IF(A.EQ.0)GOTO 25
 PRINT 76,(T(J),PV(J),A(J),V(J),Y(J),VS(J),VL(J),J=1,I)
 25 PRINT 77
C
 70 FORMAT(/*NO FAILURE - MAX. DEFLECTION OF*,F6.2,
+   * IN. REACHED*,F7.3,* SEC*)
 71 FORMAT(/*FAILURE OCCURRED AT*,F7.3,* SEC (FINAL VELOCITY =*,
+   F7.2,* IN./SEC)*)
 72 FORMAT(/*IS TIME HISTORY DESIRED (YES=1, NO=0)*,+)
 76 FORMAT(/*  TIME  PRESSURE  ACCELERATION  VELOCITY  *
+   *DISPLACEMENT    VS       VL*,/,
+   (F6.3,F9.3,F12.1,F12.2,F12.4,F11.0,F8.0))
 77    FORMAT(///,7(*-----------*))
 80 FORMAT(/*ACCELERATION NOT CONVERGING AT TIME =*,F6.3,
+           * SEC (PF =*,F7.3,* PSI)*/*      A(I) SET EQUAL TO*,
+           F8.1,* (AVG OF LAST 2 ITERATIONS)*/*      Y(I) =*,
+           F8.4,* IN.*)
 90 FORMAT(/*ARE REACTIONS TO BE OUTPUT TO FILE (0=NO,1=YES)*,+)
 95 FORMAT(/*INPUT NAME OF SLAB REACTION DATA FILE*,+)
 999    STOP
        END
 SUBROUTINE FORCE(IENTRY)
C  THIS SUBROUTINE INPUTS THE LOAD PARAMETERS AND DETERMINES
C  THE LOAD AT A GIVEN TIME FOR THE FOLLOWING LOAD TYPES:
C      1. IDEALIZED BLAST LOAD (FRONT OR SIDE FACE)
 COMMON KINC,LOTYPE,KRF,KRAND,TIME,I,Y(80),Q,QJ,YU,YFAIL,
+  ZLS,ZLL,HS,PV,FPC,FDY,ICASE,NOBAR,AS(4),APS(4),D(4),DP(4),FDC,
+  EC,ES,P,ALP,ALP2,AREA,ZMASS,VFAIL,ZKLM,VL1,VL2,VS1,VS2,
+  HEMH,ASCL,ASCS,VCL,VCS,ASS,RSS,AF,BF,CF,VK,
+  W,PO,CO,LOC,S,ZLEN,CD,PS0,PDO,PR,P,PC,TC,TO,D-LAY,
+  NWIN,RHOO,VO,L1,AA(4,2),NN(4),AFRONT,ASIDE,G,G2,G3,G4,PP2,DT
 DIMENSION TT(20),PP(20)
C
 IF(LOTYPE.EQ.5)GOTO 500
C
 GOTO(215,200,300,-),IENTRY
C
C  CALCULATE LOAD PROPERTIES FOR GIVEN PEAK PRESSURE
 200 GOTO(205,210),LOC
 205 PS0=(PR-14.0*PO+SQRT(196.0*PO*PO+196.0*PO*PR+PR*PR))/16.0
 GOTO 215
 210 PS0=PR
 215 PDO=2.5*PS0*PS0/(7.0*PO+PS0)
 U=CO*SQRT(1.0+(6.0*PS0)/(7.0*PO))
 TO=W**0.3333/(2.2399+0.1996*PS0)
 GOTO(220,225),LOC
 220 TC=3.0*S/U
 PC=PS0*(1-TC/TO)*EXP(-TC/TO)+PDO*(1-TC/TO)**2*EXP(-2*TC/TO)
 CD=1.0
 RETURN
 225 TA=ZLEN/U
 TA2=TA/2.0
 TA2TO=TA2/TO
 PA=PS0*(1-TA2TO)*EXP(-TA2TO)+CD*PDO*(1-TA2TO)**2*EXP(-2*TA2TO)
 RETURN
C
C  CALCULATE LOAD
 300 GOTO(305,310),LOC
 305 TTO=TIME/TO
 IF(TIME.GT.TC)GOTO 320
 P=PC+(TC-TIME)*(PR-PC)/TC
 RETURN
 310 TTO=(TIME-TA2)/TO
 IF(TIME.GT.TA)GOTO 320
 P=PA*TIME/TA
 RETURN
 320 IF(TTO.GE.1.0)GOTO 330
 P=PS0*(1-TTO)*EXP(-TTO)+CD*PDO*(1-TTO)**2*EXP(-2*TTO)
 RETURN
 330 P=0
 RETURN
C
C  PRINT LOAD DATA
 - IF(KINC.EQ.0)GOTO 400
```

PROGRAM RCSLAB (CONTINUED)

```
13030 PRINT 640,LDTYPE
13040 GOTO 410
13050 400 PRINT 645,LDTYPE
13060 410 CONTINUE
13070 415 GOTO(420,425),L3C
13080 420 PRINT 650
13090 GOTO 430
13100 425 PRINT 655
13110 430 PRINT 660,W,PO,CO
13120 IF(KRAND.NE.0)RETURN
13130 GOTO(435,440),L3C
13140 435 PRINT 665,S,TC,PR
13150 GOTO 445
13160 440 PRINT 670,ZLEN,TA,PA
13170 445 PRINT 675,U,TO,CD,PSO,PDO
13180 RETURN
13500C
13510C  LOAD TYPE 5 -- ARBITRARY LOAD SHAPE
13520 500 GOTO(510,520,530,540),IENTRY
13530C
13540C  INPUT LOAD DATA
13550 510 PRINT 680
13560 READ,NPOINT,(TT(J),PP(J),J=1,NPOINT)
13570 FACTOR=1.0
13580 IF(KINC.EQ.0)GOTO 518
13590 PMAX=PP(1)
13600 DO 515 J=2,NPOINT
13610 515 IF(PP(J).GT.PMAX)PMAX=PP(J)
13620 518 PX=PP(2)-PP(1)
13630 TX=TT(2)-TT(1)
13640 JJ=1
13650 RETURN
13660C
13670C  CALCULATE MAXIMUM LOAD
13680 520 FACTOR=PR/PMAX
13690 GOTO 518
13700 RETURN
13710C
13720C  CALCULATE LOAD
13730 530 IF(TIME.LE.TT(JJ+1))GOTO 535
13740 JJ=JJ+1
13750 PX=PP(JJ+1)-PP(JJ)
13760 TX=TT(JJ+1)-TT(JJ)
13765 IF(TX.EQ.0)TX=1E-10
13770 GOTO 530
13780 535 P=FACTOR*(PP(JJ)+(TIME-TT(JJ))*PX/TX)
13790 RETURN
13800C
13810C  PRINT LOAD DATA
13815 540 IF(KINC.EQ.1)PRINT 640,LDTYPE
13820 IF(KINC.EQ.0)PRINT 645,LDTYPE
13825 PRINT 690
13830 DO 545 J=1,NPOINT
13840 P=FACTOR*PP(J)
13850 545 PRINT 695,TT(J),P
13860 RETURN
14000C
14070 640 FORMAT(/*LOAD CAUSING INCIPIENT FAILURE IS AS FOLLOWS:*,
14071+     /,5X,*LOAD TYPE NUMBER*,I2)
14080 645 FORMAT(/*PROPERTIES OF LOAD ACTING ON ELEMENT ARE AS *
14081+     *FOLLOWS:*,/,5X,*LOAD TYPE NUMBER*,I2)
14090 650 FORMAT(9X,*(FRONT FACE)*)
14100 655 FORMAT(5X,*(SIDE OR TOP FACE)*)
14110 660 FORMAT(10X,*W =*,F8.1,* KT    PO =*,F6.2,* PSI      CO =*,
14111+     F7.1,* FPS*)
14120 665 FORMAT(10X,*S =*,F6.1,* FT      TC =*,F6.3,* SEC      PR =*,
14121+     F7.3,* PSI*)
14130 670 FORMAT(10X,*L =*,F6.1,* FT      TA =*,F6.3,* SEC      PA =*,
14131+     F7.3,* PSI*)
14140 675 FORMAT(10X,*U =*,F7.1,* FPS    TO =*,F6.3,* SEC      CD =*,
14141+     F5.1,/,5X,*PSO =*,F7.3,* PSI   PDO =*,F7.3,* PSI*)
14150 680 FORMAT(/*INPUT NUMBER OF LOAD POINTS AND THE TIME AND *,
14151+     *PRESSURE AT EACH POINT*)
14160 690 FORMAT(/10X,*TIME       PRESSURE*)
```

PROGRAM RCSLAB (CONTINUED)

```
14170 605 FORMAT(F15.3,F12.2)
15000 END
20000 SUBROUTINE FILL(P3,IENTRY)
20010C: COMPUTES AVERAGE AIR PRESSURE IN ROOM DUE TO BLAST WAVE
20020C: INCIDENT HEAD-ON UPON FRONT WALL.
20030C
20050 COMMON KINC,LOTYPE,KRF,KRAND,TIME,II,Y(80),D,DD,YJ,YFAIL,
20052+   ZLS,ZLL,HS,PV,FPC,FDY,ICASE,NBAR,AS(4),APS(4),D(4),DP(4),FDC,
20054+   EC,ES,DJM,ALP,ALP2,AREA,ZMASS,VFAIL,ZKLM,VL1,VL2,VF1,VF2,
20055+   HSMR,ASCL,ASCS,VCL,VCS,ASS,BSS,AF,BF,CF,NK,
20056+   M,P3,C3,L3C,S,ZLEN,CD,PS3,PD3,PR,PEXT,PC,TC,TO,DELAY,
20058+   NWIN,RH33,V3,L1,AA(8,2),NN(8),AFRONT,ASIDE,G,G2,G3,G4,PP2,DT
20090 LOGICAL L1,L2,L3
20095C
20100 GOTO(10,13,11),IENTRY
20110 10 RETURN
20310C
20320 13 P33=P3
20330 TT=0.$ T0=0.
20340 RH333=RH33
20350 L2=.FALSE. $ L3=.FALSE.
20360 RETURN
20370C
20380 11 IF(L1)GOTO 52
20385 IF(L2.A.L3)GOTO 9
20390 52 DDT=(TIME-T0)*0.5
20395 ISTOP=2
20400 53 IF(DDT.LT.DT)GOTO 51
20410 50 DDT=0.5*DDT
20415 ISTOP=2*ISTOP
20420 GO TO 53
20430 51 CONTINUE
20440 DO 99 I=1,ISTOP
20450 TT=T0+I*DDT
20460 IF(TT.GT.TO)GO TO 99
20470 DM=0. $ WW=0. $ NW=0
20480 DO 500 K=1,NWIN
20490 M=NN(K) $ DLY=AA(K,2)+0.000001
20500 IF(DLY.GE.TT)GO TO 500
20510 GOTO(15,16,16),M
20520 15 CDF=1.0
20530 IF(TT-TC)20,20,21
20540 20  P11=(TC-TT)*(PR-PC)/TC+PC
20550 P11=P11+P3
20560 GO TO 30
20570 16 CDF=-0.4
20600 21  R=TT/TO $ RR=1.-R
20610 PD=PD3*RR*RR*EXP(-2.*R)
20620 PS=PS3*RR*EXP(-R)
20630 P11=PS+CDF*PD
20640 P11=P11+P3
20650 30  RH31=RH33*((P11/P3)**G2)
20660 IF(P11-P33)36,36,37
20670 36  JSIGN=-1
20680 L2=.TRUE.
20770 303  P2=P11
20780 RH32=((P2/P33)**G2)*RH333
20790 X=P33/RH333
20800 GO TO 38
20810 37  JSIGN=+1
20820 306 P2=PP2*P11
20830 RH32=((P2/P11)**G2)*RH31
20840 X=P11/RH31
20850 38  U22=G4*(X-P2/RH32)*32.*144.
20860 IF(U22)40,39,39
20870 40  PRINT,*U22 NEGATIVE*,U22
20880 STOP
20890 39  U2=SQRT(U22)*JSIGN
20900 DDM=U2*RH32*AA(K,1)*DDT
20910 DM=DM+DDM
20920 WW=WW+P11*DDM/(G3*RH31)
20925C
20930 500 CONTINUE
20940 P33=P33+(G-1.)*WW/V3
```

PROGRAM RCSLAB (CONTINUED)

```
20950 RHO33=RHO33+DM/V3
20960 99  CONTINUE
20970 T3=TT
20980 P3=P33-P3
20982 IF(TIME.GE.TC)L3=.TRUE.
20983 RETURN
20984 9 R=TIME/TO $ RR=1.0-R
20985 PD=PD0*RR*RR*EXP(-2.0*R)
20986 PS=PS0*RR*EXP(-R)
20987 P3=PS+PD*(AFRONT-0.4*ASIDE)
20990 999 RETURN
21020 END
30000 SUBROUTINE RESIST (IENTRY)
30010C: THIS SUBROUTINE INPUTS THE REQUIRED WALL DATA$ DETERMINES THE
30020C: RESISTANCE FUNCTION, TRANSFORMATION FACTORS, AND REACTION
30030C: COEFFICIENTS$ AND SUPPLIES THE REACTION VALUES FOR SPECIFIC
30040C: DEFLECTIONS REQUIRED IN THE DYNAMIC ANALYSIS
30045C
30050 COMMON KINC,LDTYPE,KRF,KRAND,TIME,I,Y(80),Q,QU,YU,YFAIL,
30052+  ZLS,ZLL,HS,PV,FPC,FDY,ICASE,NOBAR,AS(4),APS(4),D(4),DP(4),FDC,
30054+  EC,ES,R,ALP,ALP2,AREA,ZMASS,VFAIL,ZKLM,VL1,VL2,VS1,VS2,
30055+  MEMB,ASCL,ASCS,VCL,VCS,ASS,BSS,AF,BF,CF,NK,
30056+  W,P3,C3,L3C,S,ZLEN,CD,PS3,PD3,PR,PEXT,PC,TC,TO,DELAY,
30058+  NWIN,RHO3,V3,L1,AA(8,2),NN(8),AFRONT,ASIDE,G,G2,G3,G4,PP2,DT
30070 COMMON /SAR/ SAREAS,SAREAL
30100 REAL N,IC,IG,MM,KK1,KK2,KK3,MU(4),ICR(4),KT
30130 GOTO(4,500,45),IENTRY
30140 4 RETURN
30810C
30820C: *******************************************
30830C: * ENTRY 2: DETERMINE ELEMENT PROPERTIES *
30840C: *          DEPENDENT ON FDC, FDY, AND D *
30850C
30860 45 N=ES/EC
30870 FR=8.0*SQRT(FDC)
30880 IG=HS**3/12+(N-1)*(AS(1)*(D(1)-HS/   **2+APS(1)*(HS/2-DP(1))**2)
30900 MM=2.0*IG*FR/HS
30920 CALL MOMENT(FDC,FDY,ES,N,0,1.0,AS,4,HS,D,DP,MU,ICR,IC)
31490 GMU=MU(2)/MU(1)
31500C
31510C: DETERMINE POSITION OF YIELD LINES AND ULTIMATE RESISTANCE
31520C: COEFFICIENTS FOR TWO-WAY SLAB
31530 IF(ICASE.GT.4)GOTO 106
31540 ZI1=MU(4)/MU(2)
31550 ZI3=MU(3)/MU(1)
31560 GAMMA12=2.0*SQRT(1.0+ZI1)
31570 GAMMA34=2.0*SQRT(1.0+ZI3)
31580 GRAT=GAMMA12/GAMMA34
31590 B=SQRT(1+ZI1)*(GMU*ALP2/GAMMA34)*(SQRT(GRAT**2+3/
31600+  (GMU*ALP2))-GRAT)
31610C: IF BETA IS GREATER THAN 0.5 ASSUMPTION OF CRACK PATTERN A
31620C: NOT VALID AND CRACK PATTERN B IS ASSUMED TO OCCUR
31630 IF(B.LE.0.5)GOTO 105
31640C
31650C: CRACK PATTERN B
31660 KRACK=1
31670 B=SQRT(1.0+ZI3)*(SQRT(1.0/GRAT**2+GMU*ALP2*3.0)-1.0/GRAT)
31680+  /(GMU*ALP2*GAMMA12)
31690 OUTERM=6.0*GAMMA12**2*ALP*GMU/(SQRT(3+1/(GMU*ALP2*GRAT**2))
31700+  -1.0/(SQRT(GMU)*ALP*GRAT))**2
31705 SAREAS=0.5*ZLS*ZLL*(1.0-B) $ SAREAL=0.5*ZLS*ZLL*B
31710 GOTO 108
31720C
31730C: CRACK PATTERN A
31740 105 CONTINUE
31750 KRACK=0
31760 OUTERM=6.0*GAMMA34**2/(ALP*(SQRT(3+GMU*ALP2*GRAT**2)
31770+  -ALP*GRAT*SQRT(GMU))**2)
31775 SAREAS=0.5*ZLS*ZLL*B $ SAREAL=0.5*ZLS*ZLL*(1.0-B)
31780 GOTO 108
31790C
31800C: DETERMINE MOMENT AND DEFLECTION COEFFICIENTS
31810C: FOR CRACKED PORTION OF SLAB BEHAVIOR
31820 106 B=0
```

PROGRAM RCSLAB (CONTINUED)

```
      SAREAS=0.0 $ SAREAL=0.5*ZLS*ZLL
  108 CONTINUE
      GOTO(195,195,195,195,182,185,190),ICASE
  182 QUTERM=1.0/(BSS*ZLS)
      GOTO 195
  185 QUTERM=(MU(3)/MU(1)+1.0)/(BSS*ZLS)
      GOTO 195
  190 QUTERM=(0.5*MU(3)/MU(1)+1.0)/(BSS*ZLS)
  195 GOTO(200,210,210,210,200,210,210),ICASE
C
C: **************************************************************
C: * DETERMINE RESISTANCE CURVE FOR WALL                         *
C: * (Q IS IN UNITS OF PSI, KK IN LB/CU.IN., AND Y IN INCHES)    *
C: **************************************************************
C
C: CASES 1 AND 5
  200 Q1=MM/(BSS*ZLS*ZLS)
      KK1=EC*IG/(ASS*ZLS**4)
      Y1=Q1/KK1
      KK2=EC*IC/(ASS*ZLS**4)
      IF(ICASE.EQ.5)GOTO 205
      QU=QUTERM*MU(1)/AREA
      GOTO 208
  205 QU=QUTERM*MU(1)/ZLS
  208 YU=QU/KK2
      GOTO 280
C
C: CASES 2, 3, 4, 6, & 7
  210 Q1=MM/(BF*ZLS*ZLS)
      KK1=EC*IG/(AF*ZLS**4)
      Y1=Q1/KK1
      Q2=MU(NK)/(CF*ZLS*ZLS)
      KK2=EC*IC/(AF*ZLS**4)
      Y2=Q2/KK2
      KK3=EC*IC/(AFS*ZLS**4)
      IF(ICASE.GT.4)GOTO 215
      QU=QUTERM*MU(1)/AREA
      GOTO 220
  215 QU=QUTERM*MU(1)/ZLS
  220 YU=Y2+(QU-Q2)/KK3
  280 CONTINUE
      QFAIL=QU
      YT=999.9
C
C: CHECK FOR TYPE OF FAILURE - LIGHTLY REINFORCED OR CONVENTIONAL
      IF(MU(1).LT.1.5*MM)GOTO 288
C
C: CONVENTIONAL TYPE FAILURE
      IF(ICASE.EQ.1.OR.ICASE.EQ.5)GOTO 272
      YE=Y2+YU*(1.0-Q2/QU)
      GOTO 273
  272 YE=YU
  273 YFAIL=YE*0.1/(AS(1)/D(1))
C: DUCTILITY FACTOR MUST BE <= 30
      IF(YFAIL.GT.30.0*YE)YFAIL=30.0*YE
      GOTO 300
C
C: LIGHTLY REINFORCED TYPE OF FAILURE
C:  THE FOLLOWING EXPRESSION IS BASED ON A STEEL ELONGATION OF 20%
  288 UCOEF=30.0
      UPU=UCOEF*SQRT(FDC)
      ABAR=3.14159*(NOBAR/16.)**2
  290 YFAIL=SQRT((0.2*ABAR*FDY/UPU+ZLS/2.)**2-(ZLS/2.)**2)
C:
C  TENSILE MEMBRANE BEHAVIOR
  300 IF(MEMB.NE.1)GOTO 385
      TS=ASCS*FDY
      IF(ASCL.EQ.0)GOTO 312
      TL=ASCL*FDY
      C1=3.14159*SQRT(TS/TL)*ZLL/(2.0*ZLS)
      KT=0
      DO 310 JJ=1,13,4
      JJ2=JJ+2
      COSHJJ=0.5*(EXP(JJ*C1)+EXP(-JJ*C1))
```

PROGRAM RCSLAB (CONTINUED)

```
C0SHJJ2=0.5*(EXP(JJ2*C1)+EXP(-JJ2*C1))
C2=(1.0-1.0/C0SHJJ)/JJ**3-(1.0-1.0/C0SHJJ2)/JJ2**3
310 KT=KT+C2
KT=1.5*3.14159**3/(4.0*KT)
G0T0 314
312 KT=8.0*1.5
314 YT=QU*ZLS*ZLS/(KT*TS)
QT=QU
IF(YT.LE.YFAIL)G0T0 316
YT=YFAIL
QT=YT*KT*TS/(ZLS*ZLS)
316 YFAIL=0.15*ZLS
QFAIL=YFAIL*KT*TS/(ZLS*ZLS)
C
C * ADJUST L0AD-DEFLECTI0N CURVE F0R SLAB DEAD L0AD
285 QDL=150.0*HS/1728.0
IF(QDL.GT.Q1)G0T0 292
YDL=QDL/KK1
G0T0 295
292 G0T0(293,294,294,294,293,294,294),ICASE
293 YDL=Y1+(QDL-Q1)*(YU-Y1)/(QU-Q1)
IF(QDL.LT.QU)G0T0 295
PRINT,*QDL =*,QDL,*  QU =*,QU $ ST0P
294 YDL=Y1+(QDL-Q1)*(YU-Y1)/(QU-Q1)
IF(QDL.LT.Q2)G0T0 295 $ PRINT,*QDL =*,QDL,*  Q2 =*,Q2 $ ST0P
295 C0NTINUE
Y1=Y1-YDL$ Y2=Y2-YDL$ YU=YU-YDL$ YT=YT-YDL$ YFAIL=YFAIL-YDL
Q1=Q1-QDL$ Q2=Q2-QDL$ QU=QU-QDL$ QT=QT-QDL$ QFAIL=QFAIL-QDL
IF(KRAND.NE.1)PRINT 633,QDL,YDL
C
C: 0UTPUT L0AD-DEFLECTI0N CURVE
IF(KRAND.EQ.1)G0T0 335
PRINT 650
IF(ICASE.EQ.1.0R.ICASE.EQ.5)G0T0 320
PRINT 660,Q1,Y1,Q2,Y2
G0T0 330
320 PRINT 660,Q1,Y1
330 IF(MEMB.EQ.1)G0T0 332
PRINT 660,QU,YU,QFAIL,YFAIL
G0T0 335
332 IF(QT.NE.QU)G0T0 333
PRINT 660,QU,YU,QT,YT,QFAIL,YFAIL
G0T0 335
333 PRINT 660,QU,YU,QU,YT,QT,YT,QFAIL,YFAIL
335 C0NTINUE
C
CALL TRANS (R,ZLS,ZLL,ICASE,KRAK,ZKLMSE,ZKLMFE,ZKLMP,VL1S,VL2S,
+  VS1S,VS2S,VL1F,VL2F,VS1F,VS2F,VL1P,VL2P,VS1P,VS2P)
QSHRL=VCL*D(1)*ZLL/((VL1S+VL2S)*AREA)
IF(ICASE.GT.4)G0T0 340
QSHRS=VCS*D(2)*ZLS/((VS1S+VS2S)*AREA)
IF(KRAND.NE.1)PRINT 690,QSHRL,QSHRS
G0T0 345
340 IF(KRAND.NE.1)PRINT 695,QSHRL
345 C0NTINUE
RETURN
C
C: *****************************************************
C: * ENTRY 3: DETERMINE THE RESISTANCE (PER UNIT AREA) *
C: *            0F THE WALL AS A FUNCTI0N 0F Y(1)        *
C: *****************************************************
C
500 IF(Y(1).GE.YFAIL)G0T0 560
IF(Y(1).GT.YU)G0T0 540
G0T0(501,520,520,520,501,520,520),ICASE
501 C0NTINUE
C
C: ELASTIC RANGE -- CASES 1 AND 5
ZKLM=ZKLMSE
VL1=VL1S $ VL2=VL2S
VS1=VS1S $ VS2=VS2S
IF(Y(1).GT.Y1)G0T0 510
C
C: UNCRACKED P0RTI0N -- ALL CASES
505 Q=Y(1)*KK1
```

PROGRAM RCSLAB (CONTINUED)

```
33130 RETURN
33190C
33200C: CRACKED PØRTIØN -- CASES 1 AND 5
33210 510 Q=Q1+(Y(I)-Y1)*(QU-Q1)/(YU-Y1)
33220 RETURN
33230C
33240 520 IF(Y(I).GT.Y2)GØTØ 530
33250C
33260C: ELASTIC RANGE -- CASES 2,3,4,6,7
33270 ZKLM=ZKLMEF
33280 VL1=VL1F $ VL2=VL2F
33290 VS1=VS1F $ VS2=VS2F
33300 IF(Y(I).LT.Y1)GØTØ 505
33310C: CRACKED PØRTIØN -- CASES 2,3,4,6,7
33320 Q=Q1+(Y(I)-Y1)*(Q2-Q1)/(Y2-Y1)
33325 RETURN
33330C
33340C: ELASTØ-PLASTIC RANGE -- CASES 2,3,4,6,7
33350 530 ZKLM=ZKLMSE
33360 VL1=VL1S $ VL2=VL2S
33370 VS1=VS1S $ VS2=VS2S
33380 Q=Q2+K3*(Y(I)-Y2)
33390 RETURN
33400C
33410C: PLASTIC RANGE -- ALL CASES
33420 540 ZKLM=ZKLMP
33430 VL1=VL1P $ VL2=VL2P
33440 VS1=VS1P $ VS2=VS2P
33450 IF(Y(I).GT.YT)GØTØ 550
33460 Q=QU
33470 RETURN
33480C
33490C  TENSILE MEMBRANE RANGE -- ALL CASES
33500 550 Q=QT+(Y(I)-YT)*(QFAIL-QT)/(YFAIL-YT)
33510 RETURN
33520C
33530C: ELEMENT CØLLAPSED - NØ RESISTANCE (TØ AVØID NUMERICAL DIFFICULTIES
33540C: FØR CERTAIN CASES SET RESISTANCE EQUAL TØ VERY SMALL VALUE)
33550 560 Q=1E-10
33560 RETURN
33570C
33683 633 FØRMAT(/*QDL =*,F6.2,* PSI      YDL =*,F8.4,* IN.*)
33700 650 FØRMAT(//*LØAD-DEFLECTIØN CURVE*,/,3X,*Q (PSI)     Y (IN.)*)
33710 660 FØRMAT(F9.2,F12.4)
33750 690 FØRMAT(/*QSHRL =*,F9.2,* PSI      QSHRS =*,F9.2,* PSI*)
33760 695 FØRMAT(/*QSHRL =*,F9.2,* PSI*)
33810 END
35000 SUBRØUTINE MØMENT(FDC,FDY,ES,N,PV,B,AS,APS,D,DP,MU,ICR,IC)
35010C THIS SUBRØUTINE DETERMINES THE ULTIMATE MØMENT CAPACITY AND
35020C CRACKED MØMENT ØF INERTIA  FØR REQUIRED SECTIØNS
35040 REAL K1,K2,K3,KUD,N,IC,ICTØT,MU(4),ICR(4),AS(4),APS(4),D(4),DP(4)
35050C
35060C: DETERMINE VALUES ØF CØNCRETE PARAMETERS
35070 45 K1=0.94-FDC/26E3
35080 K2=0.50-FDC/8E4
35090 K3=(3900.0+0.35*FDC)/(3E3+0.82*FDC-FDC*FDC/26E3)
35100 EPSC=0.004-FDC/65E5
35150C: **************************************************
35160C: * DETERMINE ULTIMATE MØMENT CAPACITY AND CRACKED *
35170C: * MØMENT ØF INERTIA FØR REQUIRED SECTIØNS        *
35180C: **************************************************
35190C
35200 II=0$ ICTØT=0
35210 DØ 170 I=1,4
35220 IF(AS(I).EQ.0)GØTØ 170
35230 II=II+1
35240C: ALL PRØPERTIES ARE CØMPUTED FØR A WIDTH B
35250 TENS=AS(I)*FDY*PV
35260 IF(APS(I).LE.0)GØTØ 150
35270C
35280C: WALL HAS CØMPRESSIØN REINFØRCEMENT
35290 C=K1*K3*FDC*B*DP(I)
35300 TERM1=0.5*(TENS/APS(I)+ES*EPSC)
35310 TERM2=ES*EPSC*(TENS-C)/APS(I)
35320C: DETERMINE LØCATIØN ØF NEUTRAL AXIS
```

PROGRAM RCSLAB (CONTINUED)

```
IF(TENS.LE.C)GOTO 140
C
C: KUD > D'
FPS=TERM1+K3*FDC/2.0-SQRT((TERM1+K3*FDC/2.0)**2
+  -(TERM2+ES*EPSC*K3*FDC))
C: F'S MUST BE <= FDY
IF(FPS.LT.FDY)GOTO 130
FPS=FDY
130 TPS=APS(I)*(FPS-K3*FDC)
KUD=(TENS-TPS)/(K1*K3*FDC*B)
MU(I)=(TENS-TPS)*(D(I)-K2*KUD)+TPS*(D(I)-DP(I))
ICR(I)=B*KUD**3/3.0+N*AS(I)*(D(I)-KUD)**2
+  +(N-1)*APS(I)*(KUD-DP(I))**2
GOTO 152
C
C: KUD < D'
140 FPS=-TERM1+SQRT(TERM1**2-TERM2)
C: F'S MUST BE <= FDY
IF(FPS.LT.FDY)GOTO 145
FPS=FDY
145 TERM3=TENS+APS(I)*FPS
KUD=TERM3/(K1*K3*FDC*B)
MU(I)=TERM3*(D(I)-K2*KUD)-APS(I)*FPS*(D(I)-DP(I))
ICR(I)=B*KUD**3/3+N*AS(I)*(D(I)-KUD)**2+N*APS(I)*(DP(I)-KUD)**2
GOTO 152
C
C: WALL HAS NO COMPRESSION REINFORCEMENT
150 KUD=TENS/(K1*K3*FDC*B)
MU(I)=TENS*(D(I)-K2*KUD)
ICR(I)=B*KUD**3/3.0+N*AS(I)*(D(I)-KUD)**2
C
152 ICTOT=ICTOT+ICR(I)
170 CONTINUE
C
C: DETERMINE AVERAGE CRACKED MOMENT OF INERTIA
175 IC=ICTOT/II
RETURN
END
SUBROUTINE TRANS (B,ZLV,ZLH,ICASE,KRAK,ZKLMSE,ZKLMFE,ZKLMP,VL1S,
+  VL2S,VS1S,VS2S,VL1F,VL2F,VS1F,VS2F,VL1P,VL2P,VS1P,VS2P)
C
C: THIS SUBROUTINE DETERMINES LOAD AND MASS TRANSFORMATION FACTORS
C: AND DYNAMIC REACTION COEFFICIENTS FOR TWO-WAY WALLS.
C
C: DETERMINE LOAD AND MASS TRANSFORMATION FACTORS
      B2=B*B
      B3=B*B2
      B4=B2*B2
      B5=B2*B3
      B6=B3*B3
C
C: CASES 1 & 5 -- ELASTIC RANGE
330   ZKMSE1=20.48*B3*(1./12.-B2/7.5+B3/21+B4/14-B5/15+B6/90)
      ZKMSE2=0.5038-0.7066*B
      ZKLSE1=6.4*B2*(1./6.-B2/10.+B3/30.)
      ZKLSE2=0.64-0.8134*B
      BARS1=B*(1./12.-B2/15.+B3/42.)/(1./6.-B2/10.+B3/30.)
      BARS2=(0.127083-0.184524*B)/(0.4-0.508333*B)
      ZKMSE=ZKMSE1+ZKMSE2
      ZKLSE=ZKLSE1+ZKLSE2
      IF(KRAK.EQ.1)GOTO 335
C: CRACK PATTERN A
      CVS=0.5*B
      CVL=0.5*(1.0-B)
      XP=ZLH*B/3.0
      XBARS=BARS1*ZLH
      ZP=ZLV*(1.0-4.0*B/3.0)/(4.0*(1.0-B))
      ZBARS=BARS2*ZLV
      XBARP=0.5*B*ZLH
      ZBARP=ZLV*(1./24.-B/16.)/(1./8.-B/6.)
      GOTO 338
C: CRACK PATTERN B
335   CVS=0.5*(1.0-B)
      CVL=0.5*B
      XP=ZLH*(1.0-4.0*B/3.0)/(4.0*(1.0-B))
```

```
      XBARS=BARS2*ZLH
      ZP=ZLV*B/3.0
      ZBARS=BARS1*ZLV
      XBARP=ZLH*(1./24.-B/16.)/(1./8.-B/6.)
      ZBARP=0.5*B*ZLV
  338 ZKLMSE=ZKMSE/ZKLSE
      GOTO(390,340,350,360,390,340,470),ICASE
C
C: CASES 2, 3, & 4 -- ELASTIC RANGE
  350 IF(KRAK.EQ.1)GOTO 365
      GOTO 340
  360 IF(KRAK.EQ.0)GOTO 365
C: CASES 2A, 2B, 3A, 4B, & 6
  340 ZKMFE1=512.0*B5*(1.0/30.-B/10.5+3.*B2/28.-B3/18.+B4/90.)
      ZKLFE1=32.0*B3*(1./12.-B/10.+B2/30.)
      BARF1=B*(.05-B/15.+B2/42.)/(1./12.-B/10.+B2/30.)
      GOTO(370,365,370,370,370,365),ICASE
C: CASES 2A, 2B, 3B, 4A, & 6
  365 ZKMFE2=0.4065-0.6144*B
      ZKLFE2=0.5344-0.7328*B
      BARF2=(.091667-.138095*B)/(.266667-.366667*B)
      GOTO(375,368,375,375,375,368),ICASE
C: CASES 2A & 2B
  368 ZKMFE=ZKMFE1+ZKMFE2
      ZKLFE=ZKLFE1+ZKLFE2
      GOTO 380
C:  CASES 3A & 4B
  370 ZKMFE=ZKMFE1+ZKMSE2
      ZKLFE=ZKLFE1+ZKLSE2
      GOTO 380
C: CASES 3B, 4A, & 6
  375 ZKMFE=ZKMSE1+ZKMFE2
      ZKLFE=ZKLSE1+ZKLFE2
  380 ZKLMFE=ZKMFE/ZKLFE
      GOTO 390
C: CASE 7
  470 ZKLMFE=0.78
C
C: ALL CASES -- PLASTIC RANGE
  390 ZKMP=(1.0-B)/3.0
      ZKLP=0.5-B/3.0
      ZKLMP=ZKMP/ZKLP
C
C
C: DETERMINE DYNAMIC REACTION COEFFICIENTS FOR SHORT (VS) AND
C: LONG (VL) EDGES
C
      IF(ICASE.LT.5)GOTO 395
      XBARS=1E-10$ BARF1=1E-10$ XBARP=1E-10
  395 CONTINUE
      GOTO(450,400,400,420,450,400,445),ICASE
  400 IF(KRAK.EQ.1)GOTO 410
      XBARF=BARF1*ZLH
      IF(ICASE.EQ.3)GOTO 430
  405 ZBARF=BARF2*ZLV
      GOTO 440
  410 XBARF=BARF2*ZLH
      IF(ICASE.EQ.3)GOTO 435
  415 ZBARF=BARF1*ZLV
      GOTO 440
  420 IF(KRAK.EQ.1)GOTO 425
      XBARF=BARS1*ZLH
      GOTO 405
  425 XBARF=BARS2*ZLH
      GOTO 415
  430 ZBARF=BARS2*ZLV
      GOTO 440
  435 ZBARF=BARS1*ZLV
  440 CONTINUE
C
C: CASES 2, 3, 4, & 6 -- ELASTIC RANGE
      VS1F=CVS*(1.0-XP/XBARF)
      VS2F=CVS*(XP/XBARF)
      VL1F=CVL*(1.0-ZP/ZBARF)
      VL2F=CVL*(ZP/ZBARF)
```

PROGRAM RCSLAB (CONTINUED)

```
      GOTO 450
C
C: CASE 7 -- ELASTIC RANGE
 445  VS1F=0
      VL1F=0.459
      VL2F=0.165
C
C: CASE 1 & 5 -- ELASTIC RANGE
 450  VS1S=CVS*(1.0-XP/XBARS)
      VS2S=CVS*(XP/XBARS)
      VL1S=CVL*(1.0-ZP/ZBARS)
      VL2S=CVL*(ZP/ZBARS)
C
C: ALL CASES -- PLASTIC RANGE
 460  VS1P=CVS*(1.0-XP/XBARP)
      VS2P=CVS*(XP/XBARP)
      VL1P=CVL*(1.0-ZP/ZBARP)
      VL2P=CVL*(ZP/ZBARP)
      RETURN
      END
      SUBROUTINE RANDOM (IENTRY)
C THIS SUBROUTINE INPUTS MEAN AND STANDARD DEVIATIONS FOR RANDOM
C VARIABLES;  GENERATES RANDOM VALUES;  AND CONTROLS REQUIRED
C NUMBER OF CASES TO BE RUN;  AND OUTPUTS FINAL RESULTS AND SUMMARY
C
 COMMON KINC,LDTYPE,KRF,KRAND,TIME,I,Y(30),Q,QU,YU,YFAIL,
+  ZLS,ZLL,HS,PV,FPC,FDY,ICASE,NOBAR,AS(4),APS(4),D(4),DP(4),FDC,
+  EC,ES,R,ALP,ALP2,AREA,ZMASS,VFAIL,ZKLM,VL1,VL2,VS1,VS2,
+  MEMB,ASCL,ASCS,VCL,VCS,ASS,BSS,AF,BF,CF,NX,
+  W,PO,CO,LOC,S,ZLEN,CD,PSO,PDO,PR,PEXT,PC,TC,TO,DELAY,
+  NWIN,RHOO,V3,L1,AA(8,2),NN(8),AFRONT,ASIDE,G,G2,G3,G4,PP2,DT
       COMMON /RAND/ TIMEC
       DIMENSION CHI25(7),CHI975(7),TDIST(7)
C
C VALUES FOR 97.5%  (F=19,24,29,34,39,44,49)
      DATA CHI25/.4688,.5167,.5533,.5825,.6065,.6267,.6440/
      DATA CHI975/1.7295,1.6402,1.5766,1.5284,1.4903,1.4591,1.4331/
      DATA TDIST/2.093,2.064,2.045,2.032,2.022,2.016,2.010/
C
      GOTO(5,50,70),IENTRY
    5 XDUMMY=XNORM1(-1.0,0.0,1.0)
C INITIALIZE RANDOM NUMBER GENERATOR
      PRINT,/,*INPUT NRAND*,
      READ,NRAND
      DO 47 I=1,NRAND
      XDUMMY=XNORM1(0.0,0.0,1.0)
   47 CONTINUE
      INDEX=0$  SPSO=0$  SSPSO=0
      ICHECK=20
C
C  INPUT MEAN AND STANDARD DEVIATION FOR RANDOM VARIABLES
      IF(LOC.EQ.2)GOTO 30
      PRINT 87
      READ,SMEAN,SSD
C  REINFORCED CONCRETE WALLS
 30   PRINT 86
      READ,FDYMEAN,FDYSD
      IF(LOC.EQ.1)PRINT 96
      IF(LOC.NE.1)PRINT 95
      RETURN
C
C  GENERATE RANDOM VALUES
 50   FDY=XNORM1(0.0,FDYMEAN,FDYSD)
      IF(FDY.LE.0)GOTO 50
      IF(LOC.EQ.2.OR.SMEAN.EQ.0)GOTO 65
 60   S=XNORM1(0.0,SMEAN,SSD)
      IF(S.LE.0)GOTO 60
 65   INDEX=INDEX+1
      RETURN
C SUM VALUES OF PSO AND PSO**2 FOR USE IN STATISTICAL ANALYSIS
 70   SPSO=SPSO+PSO
      SSPSO=SSPSO+PSO*PSO
C
C OUTPUT FINAL RESULTS
 76   IF(LOC.EQ.1)PRINT 92,FDY,S,PSO,TIMEC
      IF(LOC.NE.1)PRINT 90,FDY,PSO,TIMEC
```

```
   80  IF(INDEX.LT.ICHECK)RETURN
C
C DETERMINE MEAN, STANDARD DEVIATION, AND STANDARD ERROR FOR PSO
       ZNO=INDEX
       ZMEAN=SPSO/ZNO
       SD=SQRT((SSPSO-ZNO*ZMEAN*ZMEAN)/ZNO)
       STDERR=SD/(SQRT(ZNO-1))
C CHECK IF MAXIMUM OF 50 PSO SAMPLES OBTAINED
       IF(INDEX.EQ.50)GOTO 62
C CHECK IF 95% CONFIDENCE INTERVAL FOR MEAN PSO VALUE IS
       IF(STDERR*TDIST((INDEX-15)/5)/ZMEAN.GT.0.10)GOTO 61
C
C CONFIDENCE INTERVAL IS WITHIN 10% -- DETERMINE UPPER LIMIT OF
C 95% CONFIDENCE INTERVAL FOR STANDARD DEVIATION
C PROBABILITY VALUE AND ITS 95% CONFIDENCE INTERVAL UPPER LIMIT
    62 SDU=SD/(SQRT(CHI25((INDEX-15)/5)))
C CHECK IF MAXIMUM OF 50 PSO SAMPLES OBTAINED
       IF(INDEX.EQ.50)GOTO 53
C CHECK IF UPPER VALUE OF 95% CONFIDENCE INTERVAL FOR STANDARD
C DEVIATION IS WITHIN 0.10*MEAN OF THE STANDARD DEVIATION
       IF(((SDU-SD)/ZMEAN).GT.0.10)GOTO 61
C
C 95% CONFIDENCE INTERVAL IS WITHIN 10% FOR BOTH MEAN AND 90%
C PROBABILITY VALUE -- THEREFORE SUFFICIENT SAMPLES OBTAINED
C DETERMINE 95% CONFIDENCE INTERVALS FOR MEAN, STANDARD DEVIATION
,
C AND 10% AND 90% PROBABILITY VALUES
    53 ZMEANL=ZMEAN-STDERR*TDIST((INDEX-15)/5)
       ZMEANU=ZMEAN+STDERR*TDIST((INDEX-15)/5)
       SDL=SD/(SQRT(CHI975((INDEX-15)/5)))
       P10=ZMEAN-1.282*SD
       P10L=ZMEAN-1.282*SDU
       P10U=ZMEAN-1.282*SDL
       P90=ZMEAN+1.282*SD
       P90L=ZMEAN+1.282*SDL
       P90=ZMEAN+1.282*SD
       P90U=ZMEAN+1.282*SDU
       P90U=ZMEAN+1.282*SDU
C
C OUTPUT STATISTICAL PARAMETERS OF INCIPIENT COLLAPSE PRESSURE
       PRINT 100,ZMEAN,ZMEANL,ZMEANU,SD,SDL,SDU,P10,P10L,P10U,
+        P90,P90L,P90U
       PRINT 105,INDEX,STDERR
       GOTO 999
C
C 95% CONFIDENCE INTERVAL IS NOT WITHIN 10% FOR BOTH MEAN AND 90
C
C VALUES -- THEREFORE OBTAIN 5 ADDITIONAL SAMPLES
    61 ICHECK=ICHECK+5
       RETURN
C
   86 FORMAT(/*INPUT MEAN AND STANDARD DEVIATION FOR FDY*,*)
   87 FORMAT(/*INPUT MEAN AND STANDARD DEVIATION FOR S*,*)
   90 FORMAT(F9.1,F10.2,F14.3)
   92 FORMAT(F9.1,F11.2,F10.2,F14.3)
   95 FORMAT(///,5X,*FDY*7X,*PSO*,6X,*COLLAPSE TIME*)
   96 FORMAT(///,5X,*FDY*,9X,*S*,8X,*PSO*,6X,*COLLAPSE TIM
  100 FORMAT(///,11X,*STATISTICAL PROPERTIES OF INCIPIENT F  )*,
+        //,39X,*95% CONFIDENCE LIMITS*,/,7X,*ITEM*,19X,
+        *VALUE         LOWER         UPPER*,//,* MEAN*,F29.2,
+        2F12.2,//,* STANDARD DEVIATION*,F15.2,2F12.2,//,
+        * 10% PROBABILITY VALUE*,3F12.2,//,
+        * 90% PROBABILITY VALUE*,3F12.2)
  105 FORMAT(//,5X,*NUMBER OF OBSERVATIONS  *,I3,/,5X,
+        *STANDARD ERROR =*,F5.2)
C
  999 STOP$ END
FUNCTION XNORM1(X,A,B)
IF(X)10,20,20
10 X0=RANF(-1.0)
20 X1=RANF(0.0)
X2=RANF(0.0)
Y=SQRT(-2.0*ALOG(X1))*(C .(6.283184*X2))
XNORM1=A+Y*B
RETURN
END
```

RESTRAN

Restrained Reinforced Concrete Slab

PROGRAM RESTRAN

```
01000 PROGRAM RES (INPUT,OUTPUT)
01010C  THIS PROGRAM CALCULATES THE RESISTANCE OF A REINFORCED CONCRETE
01020C  SLAB RESTRAINED AGAINST LATERAL MOVEMENT AT THE EDGES
01030C
01050 COMMON KINC,LDTYPE,KRF,KRAND,TIME,I,Y(100),Q,QU,YU,YFAIL,FDC,
01052+  ZLS,HS,FDY,AREA,ZMASS,ZKLM,VS1,VS2,PS0,PD0,PF,PEXT,PC,TC,TO,
01054+  PO,DELAY,S,FPC,FY
01060 DIMENSION A(80),V(80),T(80),VS(80),QQ(80),PN(80)
01080 COMMON /RAND/ TIMEC
01100C
01104C  PRINT PROGRAM TITLE, DATE, AND TIME
01105 5 DA=DATER(IDATE) $ CL=CLOCK(ICLOCK)
01106 PRINT 603,IDATE,ICLOCK
01107 603 FORMAT(////*PROGRAM RESTRAN (REVISED 12/22/73)*,5X,A9,5X,A9)
01108C
01110C * READ TITLE AND CONTROL PARAMETERS *
01120 PRINT 67
01130 READ 68,TITLE
01140 PRINT 85
01150 READ,KINC,LDTYPE,KRF,KRAND
01155C
01160 DELAY=0
01180 CALL RESIST(1)
01182 IF(KRAND.NE.1)CALL RESIST(2)
01185 IF(LDTYPE.EQ.0)GOTO 50
01190 CALL FORCE(1)
01200 IF(KRF.EQ.0)GOTO 14
01210 CALL FILL(PINT,1)
01260 14 IF(KRAND.NE.1)GOTO 13
01270 CALL FORCE(4)
01280 CALL RANDOM(1)
01290 34 CALL RANDOM(2)
01300 35 CALL RESIST(2)
01310C
01320C: MINIMUM, MAXIMUM, AND STARTING VALUES ARE DETERMINED FOR CASES
01330C: WHERE THE LOAD CAUSING INCIPIENT COLLAPSE IS TO BE FOUND
01340 13    IF(KINC.EQ.0)GOTO 23
01350        PF=QU
01360        PFMAX=0
01370        PFMIN=PF/2.0
01380       GOTO 20
01390 16     PF=(PFMIN+PFMAX)/2.0
01400 20    CALL FORCE(2)
01410 23    IF(KRF.EQ.0)GOTO 24
01420       CALL FILL(PINT,2)
01430C
01440C:  INITIALIZE VALUES FOR BETA METHOD (BETA=1/6) AND COMPUTE VALUES
01450C:  FOR FIRST TIME INTERVAL ASSUMING ELEMENT INITIALLY AT REST
01460 24     I=1
01470        TIME=0
01480       V(1)=0 $ Y(1)=0
01490        DELTA=0.001
01500       IF(KRF.NE.1)GOTO 30
01510 27    IF(TIME.GE.(DELAY-.00001))GOTO 30
01520       TIME=TIME+DELTA
01530       CALL FILL(PINT,3)
01540       GOTO 27
01640 30 CALL RESIST(3)
01650       A(1)=0.0 $ VS(1)=0.0
01660       T(1)=TIME
01680C
01690C: PROCEDURE FOR ALL SUBSEQUENT TIME INTERVALS
01700 1      I=I+1
01710       IF(I.LT.81)GOTO 11
01720       PRINT 98,TIME
01730 98 FORMAT(/*I=81)  TIME =*,F6.3,*:  FAILURE ASSUMED TO NOT OCCUR*)
01740       GOTO 6
01750 11    TIME=TIME+DELTA
01760        T(I)=TIME
01770        A(I)=A(I-1)
01775       IF(KRF.NE.0)GOTO 10
01780       CALL FORCE(3)
01790       PN(I)=PEXT
01800       GOTO 2
```

PROGRAM RESTRAN (CONTINUED)

```
10   CALL FILL(PINT,3)
     PN(I)=PINT
2    CONTINUE
     DO 8 JJ=1,10
      Y(I)=Y(I-1)+DELTA*V(I-1)+DELTA*DELTA*(A(I-1)/3.+A(I)/6.)
CALL RESIST(3)
4    ANEW=AREA*(PN(I)-Q)/(ZMASS*ZKLM)
      ADELTA=ANEW-A(I)
      A(I)=ANEW
IF(ANEW.EQ.O)PRINT,*1985*,TIME,PN(I),Q,ZMASS,ZKLM,Y(I),A(I-1)
      IF(ABS(ADELTA/(ANEW+1E-10)).LT.0.01)GOTO 9
8     CONTINUE
      A(I)=ANEW-ADELTA/2.0
PRINT 80,TIME,PF,A(I),Y(I)
9     CONTINUE
      Y(I)=Y(I-1)+DELTA*V(I-1)+DELTA*DELTA*(A(I-1)/3.+A(I)/6.)
      V(I)=V(I-1)+DELTA*(A(I)+A(I-1))/2.0
     VS(I)=4.0*AREA*(VS1*PN(I)+VS2*Q)
     QQ(I)=Q
C
C: CHECK FOR MAXIMUM DEFLECTION OR FAILURE
C:   IF MAXIMUM DEFLECTION REACHED, WALL DID NOT FAIL
     IF(Y(I).LE.Y(I-1).AND.PN(I).LE.PN(I-1))GOTO 6
     IF(Y(I).LT.0)GOTO 6
     IF(Y(I).GE.YFAIL)GOTO 7
     IF(TIME-DELAY.GE.0.010)DELTA=0.002
     IF(TIME-DELAY.GE.0.020)DELTA=0.005
     IF(TIME-DELAY.GE.0.100)DELTA=0.010
     IF(TIME-DELAY.GE.0.500)DELTA=0.050
C:   IF FAILURE DEFLECTION REACHED, ELEMENT FAILED
      GOTO 1
C
C: INTERVAL HALVING PROCEDURE TO DETERMINE LOAD CAUSING INCIPIENT
C: COLLAPSE FOR CASES WHERE DESIRED
C:   ELEMENT DID NOT FAIL -- SET PFMIN TO PF
6    CONTINUE
     IF(KINC.EQ.0)GOTO 18
36   PFMIN=PF
IF(PFMAX.GT.0)GOTO 16
      PF=2.0*PF
      GOTO 20
C:   ELEMENT FAILED -- SET PFMAX TO PF
7    CONTINUE
     TIMEC=TIME
     IF(KINC.EQ.0)GOTO 18
37   PFMAX=PF
C:   CHECK TO SEE IF LOAD RANGE IS WITHIN DESIRED ACCURACY
17    IF((PFMAX-PFMIN)/PFMIN.GT.0.01)GOTO 16
     IF(KRAND.NE.1)GOTO 18
     CALL RANDOM(3)
     GOTO 34
C
C: OUTPUT DATA INCLUDES THE MAXIMUM DEFLECTION AND TIME OF
C: OCCURANCE FOR A NON-FAILING ELEMENT OR THE TIME AND VELOCITY
C: AT COLLAPSE FOR A FAILING ELEMENT.  OPTIONAL OUTPUT IS THE
C: ENTIRE BEHAVIOR TIME-HISTORY.
C
C: OUTPUT LOAD DATA
18    CALL FORCE(4)
C
C: OUTPUT FINAL RESULTS
IF(Y(I).LT.YFAIL)PRINT 70,Y(I),T(I)
IF(Y(I).GE.YFAIL)PRINT 71,T(I),V(I)
GOTO 42
C
C CHECK TO SEE IF ENTIRE TIME-HISTORY IS DESIRED
42 PRINT 78
READ,IPRINT
IF(IPRINT.EQ.0)GOTO 25
PRINT 76,(T(J),PN(J),A(J),V(J),Y(J),QQ(J),VS(J),J=1,I)
25 PRINT 77
GOTO 5
C
67 FORMAT(//*INPUT TITLE*,!)
```

PROGRAM RESTRAN (CONTINUED)

```
02730 68 FORMAT(A59)
02740 70 FORMAT(//*NO FAILURE - MAX DEFLECTION OF*,F6.2,
02750+    * IN. REACHED AT*F7.3,* SEC*)
02760 71 FORMAT(/*FAILURE OCCURRED AT*,F7.3,* SEC (FINAL VELOCITY =*,
02770+    F7.2* IN./SEC)*)
02780 72 FORMAT(/*IS TIME HISTORY DESIRED (YES=1, NO=0)*,+)
02830 76 FORMAT(/*  TIME  PRESSURE  ACCELERATION  VELOCITY  *
02840+    *DISPLACEMENT    QQ       VS*,/,
02850+    (F6.3,F9.3,F12.1,F12.2,F12.4,F10.2,F9.0))
02860 77     FORMAT(///,7(*----------*))
02870 80 FORMAT(/*ACCELERATION NOT CONVERGING AT TIME =*,F6.3,
02880+            * SEC (PF =*,F7.3,* PSI)*/*     A(I) SET EQUAL TO*,
02890+            F8.1,* (AVG OF LAST 2 ITERATIONS)*/*     Y(I) =*,
02900+            F8.4,* IN.*)
02910 85 FORMAT(/*INPUT KINC,LDTYPE,KRF,KRAND(1=RANDOM)*,+)
02950C
03000C  OUTPUT RESISTANCE FUNCTION (LDTYPE=0)
03010 50 I=1
03020 PRINT 60
03030 52 READ,YSTART,YEND,YINC
03040 IF(YINC.EQ.0)GOTO 25
03050 PRINT 62
03060 Y(I)=YSTART
03070 54 CALL RESIST(3)
03080 PRINT 63,Y(I),Q
03090 Y(I)=Y(I)+YINC
03100 IF(Y(I).LE.YEND)GOTO 54
03110 PRINT 61
03120 GOTO 52
03130C
03200 60 FORMAT(//*IF INTERMEDIATE VALUES OF RESISTANCE FUNCTION ARE *,
03210+    *TO BE PRINTED,*/*INPUT STARTING, ENDING, AND INCREMENTAL *,
03220+    *DEFLECTION VALUES*/*(IF NO INTERMEDIATE VALUES ARE *
03230+    *TO BE PRINTED, INPUT ZEROS)*)
03240 61 FORMAT(/*MORE VALUES*,+)
03250 62 FORMAT(//*DEFLECTION (IN.)*,5X,*RESISTANCE (PSI)*)
03260 63 FORMAT(F11.4,F21.2)
03270C
03500 999 STOP
03510 END
10000 SUBROUTINE FORCE(IENTRY)
10010C  THIS SUBROUTINE INPUTS THE LOAD PARAMETERS AND DETERMINES
10020C  THE LOAD AT A GIVEN TIME FOR THE FOLLOWING LOAD TYPES:
10030C      1. IDEALIZED BLAST LOAD (FRONT OR SIDE FACE)
10040C
10050 COMMON KINC,LDTYPE,KRF,KRAND,TIME,I,Y(100),Q,QU,YU,YFAIL,FDC,
10052+  ZLS,HS,FDY,AREA,ZMASS,ZKLM,VS1,VS2,PSO,PDO,PR,P,PC,TC,TO,
10054+  PO,DELAY,S,FPC,FY
10060 DIMENSION TT(20),PP(20)
10120C
10130 IF(LDTYPE.EQ.5)GOTO 500
10140C
10150 GOTO(100,200,300,4),IENTRY
10160C
10170C * INPUT LOAD PARAMETERS *
10180C: LOCATION 1. FRONT FACE LOADING (USED IN ROOM-FILLING PROCEDURE)
10190 100 V=1000.0 $ PO=14.7 $ CO=1120.0
10200 IF(KRF.NE.1)GOTO 102
10210 LOC=1
10220 IF(KRAND.EQ.1)RETURN
10230 PRINT 600
10240 READ,S
10250 GOTO 105
10260C: LOCATION 2. TOP FACE LOADING
10265 102 CD=0 $ LOC=2
10270 ZLEN=ZLS/12.0
10275 105 IF(KINC.EQ.1)RETURN
10280 PRINT 610
10285 READ,PSO
10290 PR=2.0*PSO*(7.0*PO+4.0*PSO)/(7.0*PO+PSO)
10295 GOTO 215
11000C
11010C  CALCULATE LOAD PROPERTIES FOR GIVEN PEAK PRESSURE
11030 200 GOTO(205,210),LOC
```

PROGRAM RESTRAN (CONTINUED)

```
205 PSO=(PR-14.0*PO+SQRT(196.0*PO*PO+196.0*PO*PR+PR*PR))/16.0
GOTO 215
210 PSO=PR
215 PDO=2.5*PSO*PSO/(7.0*PO+PSO)
U=CO*SQRT(1.0+(6.0*PSO)/(7.0*PO))
TO=W**0.3333/(2.2399+0.1865*PSO)
GOTO(220,225),LOC
220 TC=3.0*S/U
PC=PSO*(1-TC/TO)*EXP(-TC/TO)+PDO*(1-TC/TO)**2*EXP(-2*TC/TO)
CD=1.0
RETURN
225 TA=ZLEN/U
TA2=TA/2.0
TA2TO=TA2/TO
PA=PSO*(1-TA2TO)*EXP(-TA2TO)+CD*PDO*(1-TA2TO)**2*EXP(-2*TA2TO)
RETURN
C
C  CALCULATE LOAD
300 GOTO(305,310),LOC
305 TTO=TIME/TO
IF(TIME.GT.TC)GOTO 320
P=PC+(TC-TIME)*(PR-PC)/TC
RETURN
310 TTO=(TIME-TA2)/TO
IF(TIME.GT.TA)GOTO 320
P=PA*TIME/TA
RETURN
320 IF(TTO.GE.1.0)GOTO 330
P=PSO*(1-TTO)*EXP(-TTO)+CD*PDO*(1-TTO)**2*EXP(-2*TTO)
RETURN
330 P=0
RETURN
C
C  PRINT LOAD DATA
4 IF(KINC.EQ.0)GOTO 400
PRINT 640,LDTYPE
GOTO 410
400 PRINT 645,LDTYPE
410 CONTINUE
415 GOTO(420,425),LOC
420 PRINT 650
GOTO 430
425 PRINT 655
430 PRINT 660,W,PO,CO
IF(KRAND.NE.0)RETURN
GOTO(435,440),LOC
435 PRINT 665,S,TC,PR
GOTO 445
440 PRINT 670,ZLEN,TA,PA
445 PRINT 675,U,TO,CD,PSO,PDO
RETURN
C
C  LOAD TYPE 5 -- ARBITRARY LOAD SHAPE
500 GOTO(510,520,530,540),IENTRY
C
C  INPUT LOAD DATA
510 PRINT 680
READ,NPOINT,(TT(J),PP(J),J=1,NPOINT)
FACTOR=1.0
IF(KINC.EQ.0)GOTO 518
PMAX=PP(1)
DO 515 J=2,NPOINT
515 IF(PP(J).GT.PMAX)PMAX=PP(J)
518 PX=PP(2)-PP(1)
TX=TT(2)-TT(1)
JJ=1
RETURN
C
C  CALCULATE MAXIMUM LOAD
520 FACTOR=PR/PMAX
GOTO 518
RETURN
C
C  CALCULATE LOAD
```

```
530 IF(TIME.LE.TT(JJ+1))GOTO 535
JJ=JJ+1
PX=PP(JJ+1)-PP(JJ)
TX=TT(JJ+1)-TT(JJ)
IF(TX.EQ.0)TX=1E-10
GOTO 530
535 P=FACTOR*(PP(JJ)+(TIME-TT(JJ))*PX/TX)
RETURN
C
C  PRINT LOAD DATA
540 IF(KINC.EQ.1)PRINT 640,LDTYPE
IF(KINC.EQ.0)PRINT 645,LDTYPE
PRINT 690
DO 545 J=1,NPOINT
P=FACTOR*PP(J)
545 PRINT 695,TT(J),P
RETURN
C
600 FORMAT(//*INPUT S*,!)
610 FORMAT(/*INPUT PSO*,!)
640 FORMAT(/*LOAD CAUSING INCIPIENT FAILURE IS AS FOLLOWS:*,
+    /,5X,*LOAD TYPE NUMBER*,I2)
645 FORMAT(/*PROPERTIES OF LOAD ACTING ON WALL ARE AS FOLLOWS:*,
+    /,5X,*LOAD TYPE NUMBER*,I2)
650 FORMAT(8X,*(FRONT FACE)*)
655 FORMAT(5X,*(SIDE OR TOP FACE)*)
660 FORMAT(10X,*W =*,F8.1,* KT    PO =*,F6.2,* PSI      CO =*,
+    F7.1,* FPS*)
665 FORMAT(10X,*S =*,F6.1,* FT      TC =*,F6.3,* SEC      PR =*,
+    F7.3,* PSI*)
670 FORMAT(10X,*L =*,F6.1,* FT      TA =*,F6.3,* SEC      PA =*,
+    F7.3,* PSI*)
675 FORMAT(10X,*U =*,F7.1,* FPS    TO =*,F6.3,* SEC      CD =*,
+    F5.1,/,8X,*PSO =*,F7.3,* PSI  PDO =*,F7.3,* PSI*)
680 FORMAT(//*INPUT NUMBER OF LOAD POINTS AND THE TIME AND *
+    *PRESSURE AT EACH POINT*)
690 FORMAT(/10X,*TIME       PRESSURE*)
695 FORMAT(F15.3,F12.2)
END
SUBROUTINE FILL(P3,IENTRY)
RETURN
END
SUBROUTINE RESIST (IENTRY)
C  THIS SUBROUTINE DETERMINES THE RESISTANCE FUNCTION,
C  TRANSFORMATION FACTORRS, AND REACTION COEFFICIENTS
C  FOR A LONGITUDINALLY RESTRAINED REINFORCED CONCRETE SLAB
C
COMMON KINC,LDTYPE,KRF,KRAND,TIME,I,Y(100),Q,QU,YU,YFAIL,FDC,
+ ZLS,HS,FDY,AREA,ZMASS,ZKLM,VS1,VS2,PSO,PDO,PR,PEXT,PC,TC,TO,
+ PO,DELAY,S,FPC,FY
COMMON /MOM/ AS(4),D(4),APS(4),DP(4),MU(4),R(4),
+ K1,K2,K3,EPSC,ES,FPS,KUD,E(4)
REAL MU,K1,K2,K3,KUD,N,NUXO,NUXBLL,NUXAVG,NUZO,NUZLSR,NUZAVG
REAL MU1BAR,MU2BAR,MU3BAR,MU4BAR,MU1,MU3,NB,KT
C
GOTO (1,2,3), IENTRY
C
C  ENTRY 1. INPUT AND ECHO SLAB AND REINFORCEMENT PHOPERTIES
1 PRINT 600
READ,ZLS,ZLL,HS,FPC,FY,DIF
PRINT 670
DO 100 I=1,4
PRINT 610,I
P(I)=0.0
READ,AS(I),D(I),APS(I),DP(I)
C  SLAB ASSUMED TO BE RESTRAINED AT CENTERLINE OF CROSS SECTION
E(I)=0.0
100 CONTINUE
PRINT 711
READ,ASCZ,ASCX
ASCZ=ASCZ/12.0 $ ASCX=ASCX/12.0
PRINT 602
READ,SZ,SX
C
```

```
C  CALCULATE VARIOUS CONSTANTS
DIF=1.0+DIF/100.0
FDC=DIF*FPC $ FDY=DIF*FY
ALP=ZLS/ZLL
ALP2=ALP*ALP
ES=29E6
AREA=ZLS*ZLL
ZMASS=150.0*AREA*HS/(386.07*1728.0)
C
C  OUTPUT SLAB PROPERTIES
PRINT 620,ZLS,ZLL,HS,FDC,FDY
PRINT 630
DO 104 I=1,4
IF(AS(I).EQ.0)GOTO 104
P=AS(I)/(12.0*D(I)) $ PP=APS(I)/(12.0*D(I))
PRINT 640,I,AS(I),P,D(I),APS(I),PP,DP(I)
C  CHANGE UNITS OF REINFORCEMENT FROM SQ.IN./FT TO SQ.IN./IN.
AS(I)=AS(I)/12.0 $ APS(I)=APS(I)/12.0
104 CONTINUE
RETURN
C
C  ENTRY 2.  DETERMINE PROPERTIES DEPENDENT UPON FDY AND FDC
2 EC=57619.03*SQRT(FPC)
N=ES/EC
FDC=DIF*FPC $ FDY=DIF*FY
K1=0.94-FDC/26E3
K2=0.50-FDC/8E4
K3=(3900.0+0.35*FDC)/(3E3+0.82*FDC-FDC*FDC/26E3)
EPSC=0.004-FDC/65E6
DO 110 I=1,4
110 CALL MOMENT (0.0,I,FDY,FDC,HS)
IF(MU(1).GT.0.0)GOTO 120
QUTERM=0.0
B=0.5*(SQRT(3.0*ALP2+ALP2**2)-ALP2)
GOTO 108
120 GMU=MU(2)/MU(1)
C
C: DETERMINE POSITION OF YIELD LINES AND ULTIMATE RESISTANCE
C: COEFFICIENTS FOR TWO-WAY WALLS
ZI1=MU(4)/MU(2)
ZI3=MU(3)/MU(1)
GAMMA12=2.0*SQRT(1.0+ZI1)
GAMMA34=2.0*SQRT(1.0+ZI3)
GRAT=GAMMA12/GAMMA34
B=SQRT(1+ZI1)*(GMU*ALP2/GAMMA34)*(SQRT(GRAT**2+3/
+  (GMU*ALP2))-GRAT)
C: IF BETA IS GREATER THAN 0.5 ASSUMPTION OF CRACK PATTERN A
C: NOT VALID AND CRACK PATTERN B IS ASSUMED TO OCCUR
IF(B.LE.0.5)GOTO 105
C
C: CRACK PATTERN B
KRAK=1
B=SQRT(1.0+ZI3)*(SQRT(1.0/GRAT**2+GMU*ALP2*3.0)-1.0/GRAT)
+  /(GMU*ALP2*GAMMA12)
QUTERM=6.0*GAMMA12**2*ALP*GMU/(SQRT(3+1/(GMU*ALP2*GRAT**2))
+  -1.0/(SQRT(GMU)*ALP*GRAT))**2
IF(KRAND.NE.1)PRINT 623
623 FORMAT(/*CRACK PATTERN B -- RESULTS NOT FULLY CHECKED OUT*)
GOTO 108
C
C: CRACK PATTERN A
105 CONTINUE
KRAK=0
QUTERM=6.0*GAMMA34**2/(ALP*(SQRT(3+GMU*ALP2*GRAT**2)
+  -ALP*GRAT*SQRT(GMU))**2)
C
C  SECONDARY RESISTANCE
108 QS=QUTERM*MU(1)/(ZLS*ZLL)
C
C  DETERMINE TRANSFORMATION FACTORS AND DYNAMIC REACTION COEFFICIENS
C  TRANSFORMATION FACTORS AND DYNAMIC REACTION COEFFICIENTS
CALL TRANS (B,ZLS,ZLL,KRAK,ZKLMP,VL1P,VL2P,VS1P,VS2P)
DO 200 I=1,4
R(I)=(AS(I)-APS(I))*FDY/(K3*FDC)
```

PROGRAM RESTRAN (CONTINUED)

```
200 CONTINUE
C  SET ELASTIC STRAIN TO ZERO
EAX=0.0 $ EAZ=0.0
SXP=SX+EAX $ SZP=SZ+EAZ
C
C  CALCULATE DEFLECTION AT ULTIMATE RESISTANCE
IYU=1
C  Z-DIRECTION STRIPS
TERM1=HS+(R(3)-R(1))/K1
TERM2=0.5*ZLS*ZLS*(EPSC*(1.0+SZ)+SZP)
GOTO 206
C  X-DIRECTION STRIPS
202 IYU=2
TERM1=HS+(R(4)-R(2))/K1
TERM2=B*ZLL*ZLL*(EPSC*(2.0*B+SX-2.0*B*EAX+EAX)+SXP)
206 IF(TERM2.GT.TERM1*TERM1)GOTO 210
YU=TERM1-SQRT(TERM1**2-TERM2)
IF(YU.LT.0.42*HS)GOTO 235
C  EMPIRICAL UPPER BOUND OF 0.42 TIMES SLAB THICKNESS FOR YU
210 YU=0.42*HS
235 CONTINUE
C
C  CALCULATE IN-PLANE COMPRESSIVE FORCES
NUZO=0.5*K3*FDC*((HS-0.25*SZP*ZLS*ZLS/YU)*K1-R(1)-R(3))
IF(NUZO.LT.0)NUZO=0
IF(NUZO.EQ.0.AND.IYU.EQ.1)GOTO 202
NUZLS2=NUZO-0.25*K3*FDC*K1*YU
IF(NUZLS2.LT.0)NUZLS2=0
NUZAVG=0.5*(NUZO+NUZLS2)
NUXO=0.5*K3*FDC*((HS-0.5*B*SXP*ZLL*ZLL/YU)*K1-R(2)-R(4))
IF(NUXO.LT.0)NUXO=0
NUXBLL=NUXO-0.25*K3*FDC*K1*YU
IF(NUXBLL.LT.0)NUXBLL=0
NUXAVG=0.5*(NUXO+NUXBLL)
C
C  CALCULATE ULTIMATE COMPRESSIVE MEMBRANE RESISTANCE
CALL MOMENT (NUXAVG,2,FDY,FDC,HS)
MU2BAR=MU(2)
CALL MOMENT (NUXAVG,4,FDY,FDC,HS)
MU4BAR=MU(4)
CALL MOMENT (NUZAVG,1,FDY,FDC,HS)
MU1BAR=MU(1)
CALL MOMENT (NUZAVG,3,FDY,FDC,HS)
MU3BAR=MU(3)
CALL MOMENT (NUZLS2,1,FDY,FDC,HS)
MU1=MU(1)
CALL MOMENT (NUZO,3,FDY,FDC,HS)
MU3=MU(3)
QU=12.0/(ZLS*ZLS*(3.0-2.0*B))*(((ZLS/ZLL)**2/B)*(MU2BAR
+ +MU4BAR-YU*(2.0*NUXBLL+NUXO)/6.0)+4.0*B*(MU1BAR+MU3BAR
+ -YU*(2.0*NUZLS2+NUZO)/6.0)+2.0*(1.0-2.0*B)*(MU1+MU3-YU*NUZLS2))
C
C  TENSILE MEMBRANE BEHAVIOR
TZ=ASCZ*FDY
TX=ASCX*FDY
IF(TZ.EQ.0.0.OR.TX.EQ.0)GOTO 312
C1=3.14159*SQR (TZ/TX)*ZLL/(2.0*ZLS)
KT=0
DO 310 JJ=1,13,2
JJ2=JJ+2
COSHJJ=0.5*(EXP(JJ*C1)+EXP(-JJ*C1))
COSHJJ2=0.5*(EXP(JJ2*C1)+EXP(-JJ2*C1))
C2=(1.0-1.0/COSHJJ)/JJ**3-(1.0-1.0/COSHJJ2)/JJ2**3
310 KT=KT+C2
KT=1.5*3.14159**3/(4.0*KT)
GOTO 314
312 KT=1.5*8.0
IF(TZ.EQ.0)GOTO 316
C
C  SECONDARY AND TENSILE MEMBRANE DEFLECTION
314 YT=QS*ZLS*ZLS/(KT*TZ)
YFAIL=0.15*ZLS
QFAIL=YFAIL*KT*TZ/(ZLS*ZLS)
GOTO 317
```

PROGRAM RESTRAN (CONTINUED)

```
316 IF(TX.EQ.0)GOTO 318
C  NO TENSILE MEMBRANE REINFORCEMENT -- SET YS=YT=YFAIL=HS
YT=QS*ZLL*ZLL/(KT*TX)
YFAIL=0.15*ZLL
QFAIL=YFAIL*KT*TX/(ZLL*ZLL)
317 IF(YT.LT.3.0*YU)YS=YT
IF(YT.GT.3.0*YU)YS=3.0*YU
GOTO 320
C  NO TENSILE MEMBRANE REINFORCEMENT - SET YS=YFAIL=HS
318 YS=HS $ YT=HS $ YFAIL=HS $ QFAIL=0.0
C
C  ADJUST LOAD-DEFLECTION CURVE FOR SLAB DEAD LOAD
320 QDL=150.0*HS/172.0
YDL=YU-YU*(1.0-(QDL/QU)**1.8)**(1.0/1.8)
YU=YU-YDL $ YS=YS-YDL $ YT=YT-YDL $ YFAIL=YFAIL-YDL
QU=QU-QDL $ QS=QS-QDL $ QFAIL=QFAIL-QDL
IF(KRAND.NE.1)PRINT 633,QDL,YDL
C
C  OUTPUT LOAD-DEFLECTION CURVE
IF(KRAND.EQ.1)GOTO 335
IF(YS.EQ.YT)PRINT 660,QU,YU,QS,YS,QFAIL,YFAIL
IF(YS.NE.YT)PRINT 660,QU,YU,QS,YS,QS,YT,QFAIL,YFAIL
335 RETURN
C
600 FORMAT(/*INPUT LS,LL,HS,F'C,FY,DIF(%)*,+)
602 FORMAT(/*INPUT LONGITUDINAL EDGE DISPLACEMENT (PER UNIT *,
+     *LENGTH)*/*IN SHORT AND LONG DIRECTIONS*,+)
610 FORMAT(I6,+)
620 FORMAT(//*PROPERTIES OF LONGITUDINALLY RESTRAINED *
+     *REINFORCED CONCRETE SLAB*,/,*  LS =*,F6.1,* IN.     LL =*,
+     F6.1,* IN.*,6X,*HS =*,F6.1,* IN.*,/,* FDC =*,F7.1,* PSI*,
+     *   FDY =*,F8.1,* PSI*)
630 FORMAT(/*REINFORCEMENT VALUES*,  SECTION    AS       (P)*,
+     7X,*D*,8X,*A'S     (P')*,7X,*D'*,/,8X,*(SQ IN./FT)*,10X,
+     *(IN.)  (SQ IN./FT)*,10X,*(IN.)*)
633 FORMAT(/*QDL =*,F6.2,* PSI      YDL =*,F8.4,* IN.*)
640 FORMAT(I5,F11.4,*    (*,F5.4,*)*,F8.3,F10.4,*    (*,F5.4,*)*,
+     F8.3)
660 FORMAT(/*LOAD-DEFLECTION CURVE*,/,4X,*Q (PSI)    Y (IN.)*,
+     /,(F9.2,F12.4))
670 FORMAT(/*INPUT AS, D, APS, AND DP FOR FOLLOWING SECTIONS*)
711 FORMAT(/*INPUT CONTINUOUS TENSILE MEMBRANE REINFORCEMENT *,
+     *(SQ IN./FT)*/*IN SHORT AND LONG DIRECTIONS*,+)
C
C  ENTRY 3.  DETERMINE THE RESISTANCE (PER UNIT AREA)
C            OF THE SLAB AS A FUNCTION OF Y(I)
C
3 IF(Y(I).LT.0.OR.Y(I).GT.YFAIL)GOTO 530
IF(Y(I).GE.YU)GOTO 510
ZKLM=ZKLMP
C
C COMPRESSIVE MEMBRANE RESISTANCE
Q=QU*(1.0-(1.0-Y(I)/YU)**1.8)**(1.0/1.8)
RETURN
C
510 IF(Y(I).GE.YS)GOTO 520
C
C SECONDARY RESISTANCE
Q=0.5*(QU+QS+(QU-QS)*COS(3.1416*(Y(I)-YU)/(YS-YU)))
RETURN
C
520 IF(Y(I).GT.YT)GOTO 525
C
C  PLASTIC RESISTANCE
Q=QS
RETURN
C
C TENSILE MEMBRANE RESISTANCE
525 Q=QS+(Y(I)-YT)*(QFAIL-QS)/(YFAIL-YT)
RETURN
C
C FAILURE (SET RESISTANCE TO VERY SMALL VALUE)
530 Q=1E-11
RETURN
```

```
END
SUBROUTINE MOMENT (PV,I,FDY,FDC,HS)
C  THIS SUBROUTINE CALCULATES THE ULTIMATE MOMENT CAPACITY FOR
C  REQUIRED SECTIONS.  INCLUDED ARE THE EFFECT OF IN-PLANE FORCES
C  AT AN ECCENTRICITY E(I) FROM THE CROSS SECTION CENTERLINE.
COMMON /MOM/ AS(4),D(4),APS(4),DP(4),MU(4),R(4),
+  K1,K2,K3,EPSC,ES,FPS,KUD,E(4)
REAL MU,K1,K2,K3,KUD
C
IF(AS(I).EQ.0)GOTO 170
C: ALL PROPERTIES ARE COMPUTED FOR A UNIT WIDTH
TENS=AS(I)*FDY+PV
IF(APS(I).LE.0)GOTO 150
C
C  SECTION HAS COMPRESSION REINFORCEMENT
C=K1*K3*FDC*DP(I)
TERM1=0.5*(TENS/APS(I)+ES*EPSC)
TERM2=ES*EPSC*(TENS-C)/APS(I)
C: DETERMINE LOCATION OF NEUTRAL AXIS
IF(TENS.LE.C)GOTO 140
C
C: KUD > D'
FPS=TERM1+K3*FDC/2.0-SQRT((TERM1+K3*FDC/2.0)**2
+  -(TERM2+ES*EPSC*K3*FDC))
C: F'S MUST BE <= FDY
IF(FPS.LT.FDY)GOTO 130
FPS=FDY
130 TPS=APS(I)*(FPS-K3*FDC)
KUD=(TENS-TPS)/(K1*K3*FDC)
MU(I)=AS(I)*FDY*(D(I)-K2*KUD)+APS(I)*FPS*(K2*KUD-DP(I))
+       +PV*(0.5*HS-K2*KUD+E(I))
RETURN
C
C: KUD < D'
140 FPS=-TERM1+SQRT(TERM1**2-TERM2)
C: F'S MUST BE <= FDY
IF(FPS.LT.FDY)GOTO 145
FPS=FDY
145 TERM3=TENS+APS(I)*FPS
KUD=TERM3/(K1*K3*FDC)
MU(I)=AS(I)*FDY*(D(I)-K2*KUD)-APS(I)*FPS*(K2*KUD-DP(I))
+       +PV*(0.5*HS-K2*KUD+E(I))
RETURN
C
C  SECTION HAS NO COMPRESSION REINFORCEMENT
150 TERM1=0.5*(ES*EPSC+PV/AS(I))
FS=-TERM1+SQRT(TERM1**2+ES*EPSC*(K1*K3*FDC*D(I)-PV)/AS(I))
IF(FS.LT.FDY)GOTO 160
FS=FDY
160 IF(FS.LT.-FDY)FS=-FDY
KUD=(AS(I)*FS+PV)/(K1*K3*FDC)
IF(KUD.GT.HS)PRINT,*KUD IS GREATER THAN HS*
MU(I)=AS(I)*FS*(D(I)-K2*KUD)+PV*(0.5*HS-K2*KUD+E(I))
RETURN
C
C  SECTION HAS NO REINFORCEMENT
170 KUD=PV/(K1*K3*FDC)
MU(I)=PV*(0.5*HS-K2*KUD+E(I))
RETURN
680 FORMAT(*MU(*,I1,*) =*,F10.1,5X,*FPS =*,F10.1,5X,
+    *KUD =*,F10.3)
685 FORMAT(*MU(*,I1,*) =*,F10.1,6X,*FS =*,F10.1,5X,
+    *KUD =*,F10.3)
690 FORMAT(*MU(*,I1,*) =*,F10.1)
END
SUBROUTINE TRANS (B,ZLV,ZLH,KRAK,ZKLMP,VL1P,VL2P,VS1P,VS2P)
C
C  THIS SUBROUTINE DETERMINES LOAD AND MASS TRANSFORMATION FACTORS
C  AND DYNAMIC REACTION COEFFICIENTS FOR TWO-WAY RESTRAINED SLABS.
C  VALUES CORRESPONDING TO PLASTIC BEHAVIOR ARE USED.
C
C  LOAD AND MASS TRANSFORMATION FACTORS
      IF(KRAK.EQ.1)GOTO 335
C
```

```
C: CRACK PATTERN A
      CVS=0.5*B
      CVL=0.5*(1.0-B)
      XP=ZLH*B/3.0
      ZP=ZLV*(1.0-4.0*B/3.0)/(4.0*(1.0-B))
      XBARP=0.5*B*ZLH
      ZBARP=ZLV*(1./24.-B/16.)/(1./8.-B/6.)
      GOTO 390
C: CRACK PATTERN B
 335  CVS=0.5*(1.0-B)
      CVL=0.5*B
      XP=ZLH*(1.0-4.0*B/3.0)/(4.0*(1.0-B))
      ZP=ZLV*B/3.0
      XBARP=ZLH*(1./24.-B/16.)/(1./8.-B/6.)
      ZBARP=0.5*B*ZLV
C
C: ALL CASES -- PLASTIC RANGE
 390  ZKMP=(1.0-B)/3.0
      ZKLP=0.5-B/3.0
      ZKLMP=ZKMP/ZKLP
C
C
C  DYNAMIC REACTION COEFFICIENTS FOR SHORT (VS) AND LONG (VL) EDGES
C
C: ALL CASES -- PLASTIC RANGE
 460  VS1P=CVS*(1.0-XP/XBARP)
      VS2P=CVS*(XP/XBARP)
      VL1P=CVL*(1.0-ZP/ZBARP)
      VL2P=CVL*(ZP/ZBARP)
      RETURN
      END
      SUBROUTINE RANDOM (IENTRY)
C THIS SUBROUTINE INPUTS MEAN AND STANDARD DEVIATIONS FOR RANDOM
C VARIABLES; GENERATES RANDOM VALUES; CONTROLS REQUIRED
C NUMBER OF CASES TO BE RUN; AND OUTPUTS FINAL RESULTS AND SUMMARY
C
 COMMON KINC,LDTYPE,KRF,KRANE,TIME,I,Y(100),Q,QU,YU,YFAIL,FDC,
+ ZLS,HS,FDY,AREA,ZMASS,ZKLM,VS1,VS2,PSO,PDO,PR,PEXT,PC,TC,TO,
+ PO,DELAY,S,FPC,FY
     COMMON /RAND/ TIMEC
 DIMENSION CHI25(7),CHI975(7),TDIST(7)
C
C VALUES FOR 97.5% (F=19,24,29,34,39,44,49)
      DATA CHI25/.4688,.5167,.5533,.5825,.6065,.6267,.6440/
      DATA CHI975/1.7295,1.6402,1.5766,1.5284,1.4903,1.4591,1.4331/
      DATA TDIST/2.093,2.064,2.045,2.032,2.022,2.016,2.010/
C
      GOTO(5,50,70),IENTRY
C
C  ENTRY 1:  INITIALIZE RANDOM NUMBER GENERATOR
   5  XDUMMY=XNORM1(-1.0,0.0,1.0)
      PRINT,/,/,*INPUT NRAND*,
      READ,NRAND
      DO 47 I=1,NRAND
      XDUMMY=XNORM1(0.0,0.0,1.0)
  47  CONTINUE
      INDEX=0$  SPSO=0$  SSPSO=0
      ICHECK=20
C
C  INPUT MEAN AND STANDARD DEVIATION FOR RANDOM VARIABLES
      IF(KRF.EQ.0)GOTO 30
      PRINT 87
      READ,SMEAN,SSD
C  REINFORCED CONCRETE SLABS
  30  PRINT 86
      READ,FYMEAN,FYSD
      PRINT 85
      READ,FPCMEAN,FPCSD
      IF(KRF.EQ.1)PRINT 96
      IF(KRF.EQ.0)PRINT 97
      RETURN
C
C  ENTRY 2:  GENERATE RANDOM VALUES
  50  FY=XNORM1(0.0,FYMEAN,FYSD)
```

```
70580       IF(FY.LE.0)GOTO 50
70582  55   FPC=XNORM1(0.0,FPCMEAN,FPCSD)
70583       IF(FPC.LE.0)GOTO 55
70585       IF(KRF.EQ.0)GOTO 65
70590  60   S=XNORM1(0.0,SMEAN,SSD)
70600       IF(S.LE.0)GOTO 60
70610  65   INDEX=INDEX+1
70620       RETURN
70625C
70630C  ENTRY 3:  SUM VALUES OF PSO AND PSO**2 FOR USE IN
70635C            STATISTICAL ANALYSIS
70640  70   SPSO=SPSO+PSO
70650       SSPSO=SSPSO+PSO*PSO
70660C
70670C OUTPUT FINAL RESULTS
70730  76   IF(KRF.EQ.1)PRINT 92,FDY,FDC,S,PSO,TIMEC
70735       IF(KRF.EQ.0)PRINT 90,FDY,FDC,PSO,TIMEC
70740  80   IF(INDEX.LT.ICHECK)RETURN
70750C
70760C DETERMINE MEAN, STANDARD DEVIATION, AND STANDARD ERROR FOR PSO
70770       ZNO=INDEX
70780       ZMEAN=SPSO/ZNO
70790       SD=SQRT((SSPSO-ZNO*ZMEAN*ZMEAN)/ZNO-1
70800       STDERR=SD/(SQRT(ZNO-1))
70810C CHECK IF MAXIMUM OF 50 PSO SAMPLES OBTAINED
70820       IF(INDEX.EQ.50)GOTO 62
70830C CHECK IF 95% CONFIDENCE INTERVAL FOR MEAN PSO VALUE IS
70840       IF(STDERR*TDIST((INDEX-15)/5)/ZM    .GT.0.10)GOTO 61
70850C
70860C CONFIDENCE INTERVAL IS WITHIN 10% -- DETERMINE UPPER LIMIT OF
70870C 95% CONFIDENCE INTERVAL FOR STANDARD DEVIATION
70880C PROBABILITY VALUE AND ITS 95% CONFIDENCE INTERVAL UPPER LIMIT
70890   62 SDU=SD/(SQRT(CHI25((INDEX-15)/5)))
70900C CHECK IF MAXIMUM OF 50 PSO SAMPLES OBTAINED
70910       IF(INDEX.EQ.50)GOTO 53
70920C CHECK IF UPPER VALUE OF 95% CONFIDENCE INTERVAL FOR STANDARD
70930C DEVIATION IS WITHIN 0.10*MEAN OF THE STANDARD DEVIATION
70940       IF(((SDU-SD)/ZMEAN).GT.0.10)GOTO 61
70950C
70960C 95% CONFIDENCE INTERVAL IS WITHIN 10% FOR BOTH MEAN AND 90%
70970C PROBABILITY VALUE -- THEREFORE SUFFICIENT SAMPLES OBTAINED
70980C DETERMINE 95% CONFIDENCE INTERVALS FOR MEAN, STANDARD DEVIATION
70990,
71000C AND 10% AND 90% PROBABILITY VALUES
71010   53 ZMEANL=ZMEAN-STDERR*TDIST((INDEX-15)/5)
71020       ZMEANU=ZMEAN+STDERR*TDIST((INDEX-15)/5)
71030       SDL=SD/(SQRT(CHI975((INDEX-15)/5)))
71040       P10=ZMEAN-1.282*SD
71050       P10L=ZMEAN-1.282*SDU
71060       P10U=ZMEAN-1.282*SDL
71070       P90=ZMEAN+1.282*SD
71080       P90L=ZMEAN+1.282*SDL
71090       P90=ZMEAN+1.282*SD
71100       P90U=ZMEAN+1.282*SDU
71110       P90U=ZMEAN+1.282*SDU
71120C
71130C OUTPUT STATISTICAL PARAMETERS OF INCIPIENT COLLAPSE PRESSURE
71140       PRINT 100,ZMEAN,ZMEANL,ZMEANU,SD,SDL,SDU,P10,P10L,P10U,
71150+        P90,P90L,P90U
71160       PRINT 105,INDEX,STDERR
71170       GOTO 999
71180C
71190C 95% CONFIDENCE INTERVAL IS NOT WITHIN 10% FOR BOTH MEAN AND 90
71200B
71210C VALUES -- THEREFORE OBTAIN 5 ADDITIONAL SAMPLES
71220   61 ICHECK=ICHECK+5
71230       RETURN
71240C
71260   85 FORMAT(/*INPUT MEAN AND STANDARD DEVIATION FOR F'C*,:)
71270   86 FORMAT(/*INPUT MEAN AND STANDARD DEVIATION FOR FY*,:)
71280   87 FORMAT(/*INPUT MEAN AND STANDARD DEVIATION FOR S*,:)
71290   90 FORMAT(2F9.1,F10.2,F14.3)
71310   92 FORMAT(2F9.1,F11.2,F10.2,F14.3)
71350   96 FORMAT(///,5X,*FDY*,6X,*FDC*,8X,*S*,8X,*PSO*,6X,
```

PROGRAM RESTRAN (CONCLUDED)

```
        *COLLAPSE TIME*)
 97 FORMAT(///,4X,*FDY*,6X,*FDC*,8X,*PSO*,5X,*COLLAPSE TIME*)
 100 FORMAT(///,11X,*STATISTICAL PROPERTIES OF INCIPIENT PSO*,
        //,39X,*95% CONFIDENCE LIMITS*,/,7X,*ITEM*,18X,
        *VALUE        LOWER        UPPER*,//,* MEAN*,F29.2,
        2F12.2,//,* STANDARD DEVIATION*,F15.2,2F12.2,//,
        * 10% PROBABILITY VALUE*,3F12.2,//,
        * 90% PROBABILITY VALUE*,3F12.2)
 105 FORMAT(//,5X,*NUMBER OF OBSERVATIONS =*,I3,/,5X,
        *STANDARD ERROR =*,F5.2)
 999 STOP$ END
FUNCTION XNORM1(X,A,B)
IF(X)10,20,20
10 XO=RANF(-1.0)
20 X1=RANF(0.0)
X2=RANF(0.0)
Y=SQRT(-2.0*ALOG(X1))*(COS(6.283184*X2))
XNORM1=A+Y*B
RETURN
END
```

RCBEAM

Reinforced Concrete Support Beam

PROGRAM RCBEAM

```
PRØGRAM RCBEAM1(INPUT,ØUTPUT,TAPE1)
CALL RETR(7HRCBEAM2,7HRCBEAM2)
C * THIS PØRTIØN ØF THE PRØGRAM INPUTS THE REQUIRED ELEMENT
C:  AND LØAD DATA AND INITIALIZES CERTAIN PARAMETERS *
C
CØMMØN KINC,LDTYPE,KRF,KRAND,TIME,I,.(100),QT,QU,YU,YFAIL,
+  ZLB,BB,HB,FPC,FDY,ICASE,AS(4),APS(4),D(4),DP(4),FDC,
+  EC,ES,QAREA,PAREA,ZMASS,ZKLM,VL1,VL2,
+  MEMB,ASCS,VCL,QDLSLAB,KCØMP,HS,BS,NSLABS,NAMEF(2),KLØAD,
+  W,PØ,CØ,LØC,S,ZLEN,CD,PSØ,PDØ,PR,PEXT,PC,TC,TO,DELAY,
+  TT(80,2),PP(80,2),REAC(80,2),INDEX(2),BR(2),
+  NWIN,RHØØ,V3,L1,AA(8,2),NN(8),AFRØNT,ASIDE,G,G2,G3,G4,PP2,QT
LØGICAL L1
C
C * READ TITLE AND CØNTRØL PARAMETERS
PRINT 67
READ 68,TITLE
PRINT 720
READ,NSLABS,KLØAD
PAREA=0
IF(KLØAD.EQ.0)GØTØ 40
C
C:  INPUT SLAB REACTIØN DATA FILE DATA
PRINT 725
DØ 39 J=1,NSLABS
PRINT 735,J
READ,NAMEF(J),ISIDE
CALL PFUR(3HREX,1,NAMEF(J))
IF(ISIDE.EQ.1)READ(1,)SAREA,DUM,HS
IF(ISIDE.EQ.2)READ(1,)DUM,SAREA,HS
510 INDEX(J)=1
READ(1,)NPØINT
IF(ISIDE.EQ.2)GØTØ 520
515 READ(1,)(TT(JJ,J),PP(JJ,J),REAC(JJ,J),DUM,JJ=1,NPØINT)
GØTØ 525
520 READ(1,)(TT(JJ,J),PP(JJ,J),DUM,REAC(JJ,J),JJ=1,NPØINT)
525 BR(J)=(REAC(2,J)-REAC(1,J))/(TT(2,J)-TT(1,J))
BP=(PP(2,J)-PP(1,J))/(TT(2,J)-TT(1,J))
REWIND 1
CALL DRØP1(1)
QDLSLAB=QDLSLAB+150.0*SAREA*HS/1728.0
39 PAREA=PAREA+SAREA
KINC=0 $ LDTYPE=5 $ KRF=0 $ KRAND=0
GØTØ 45
C
C INPUT TRIBUTARY SLAB DATA
40 PRINT 730
DØ 42 J=1,NSLABS
PRINT 735,J
READ,SAREA,HS
QDLSLAB=QDLSLAB+150.0*SAREA*HS/12.0
42 PAREA=PAREA+SAREA*144.0
C
PRINT 85
READ,KINC,LDTYPE,KRF,KRAND
45 CØNTINUE
DELAY=0
67 FØRMAT(/*INPUT TITLE*,+)
68 FØRMAT(A59)
85 FØRMAT(/*INPUT KINC,LDTYPE,KRF,KRAND(1=RANDØM)*,+)
720 FØRMAT(/*INPUT NUMBER ØF SLABS SUPPØRTED BY BEAM, AND IF *,
+     *SLAB REACTIØNS*/*ARE TØ BE CALCULATED (0) ØR READ FRØM *,
+     *DATA FILE (1)*,+)
725 FØRMAT(/*INPUT REACTIØN DATA FILE NAME AND SIDE *,
+     *(1=SHØRT,2=LØNG)*)
730 FØRMAT(/*INPUT CØNTRIBUTARY AREA (SQ FT) AND THICKNESS (IN.)*)
735 FØRMAT(6X,*FØR SLAB NØ.*,I2,+)
C
C * INPUT AND ECHØ BEAM AND REINFØRCEMENT PRØPERTIES *
PRINT 615
READ,ZLB,BB,HB,FPC,FDY,ICASE,KCØMP
ICASE4=ICASE-4
FDC=1.25*FPC
EC=57619.0*SQRT(FPC)
```

PROGRAM RCBEAM (CONTINUED)

```
ES=29E6
QAREA=ZLB*BB
PAREA=PAREA+QAREA
ECKIP=EC/1000.0 $ ESKIP=ES/1000.0
11 PRINT 670
DØ 8 I=1,3,2
AS(I)=0
IF(ICASE.EQ.5.AND.I.GT.1)GØTØ 8
PRINT 625,I
READ,AS(I),D(I),APS(I),DP(I)
8 CØNTINUE
IF(KCØMP.EQ.0)GØTØ 9
PRINT 616
READ,SP,HS,ASSLAB,DSLAB
BS=0.25*ZLB
IF(BS.GT.(BB+16.0*HS))BS=BB+16.0*HS
IF(BS.GT.SP)BS=SP
9 PRINT 711
READ,MEMB
IF(MEMB.NE.1)GØTØ 15
13 PRINT 705
READ,ASCS
15 CØNTINUE
PRINT 620,ICASE,ZLB,BB,HB,FPC,FDC,ECKIP,FDY,ESKIP
IF(KCØMP.NE.0)PRINT 621,HS,BS,SP
PRINT 630
DØ 110 I=1,3,2
IF(ICASE.EQ.5.AND.I.GT.1)GØTØ 110
P=AS(I)/(BB*D(I)) $ PPR=APS(I)/(BB*D(I))
PRINT 640,I,AS(I),P,D(I),APS(I),PPR,DP(I)
110 CØNTINUE
IF(KCØMP.EQ.0)GØTØ 115
PRINT 641,ASSLAB,DSLAB
IF(ASSLAB.EQ.0)GØTØ 115
ASSLAB=BS*ASSLAB/12.0
DP(1)=(APS(1)*DP(1)+ASSLAB*DSLAB)/(APS(1)+ASSLAB)
APS(1)=APS(1)+ASSLAB
115 ZMASS=150.0*ZLB*HB*BB/(386.07*1728.0)
VCL=2.28*SQRT(FPC)/(1-2*D(1)/ZLB)+3000*(AS(1)/(BB*D(1)))/(1-D(1)/ZLB)
VCLMAX=3.5*SQRT(FPC)/(1.0-2.0*D(1)/ZLB)
IF(VCL.GT.VCLMAX)VCL=VCLMAX
PRINT 685,VCL
C
615 FØRMAT(/*INPUT LB,BB,HB,FPC,FDY,ICASE,KCØMP*,/)
616 FØRMAT(/*INPUT BEAM SPACING, HS, AS(SLAB), & D(SLAB)*,/)
625 FØRMAT(*INPUT AS,D,A'S,&D' FØR SECTIØN*,I2,/)
620 FØRMAT(//*PRØPERTIES ØF REINFØRCED CØNCRETE SUPPØRT BEAM --*,
+        * SUPPØRT TYPE NØ.*,I2,/,
+        *  LB =*,F6.1,* IN.      BB =*,F6.1,* IN.*,6X,*HB =*,
+        F6.1,* IN.*,/,* F'C =*,F7.1,* PSI    FDC =*,F7.1,
+        * PSI*,5X,*EC =*,F7.1,* KSI*,/,* FDY =*,F8.1,* PSI*,
+        3X,*ES =*,F8.1,* KSI*)
630 FØRMAT(//*REINFØRCEMENT VALUES*/* SECTIØN     AS      (P)*,
+     9X,*D*,8X,*A'S     (P')*,4X,*D'*/9X,*(SQ IN.)*,12X,
+     *(IN.)    (SQ IN.)*,12X,*(IN.)*)
670 FØRMAT(1H )
685 FØRMAT(/,*VCL =*,F6.1,* PSI*)
705 FØRMAT(/*INPUT CØNTINUØUS REINFØRCEMENT (SQ IN.)*,/)
711 FØRMAT(/*IS TENSILE MEMBRANE TØ BE INCLUDED (0=NØ)*,
+     *1=YES)*,/)
640 FØRMAT(I5,F11.4,* (*,F6.4,*)*,F9.3,F10.4,* (*,F6.4,*)*,F9.3)
621 FØRMAT(*  HS =*,F6.1,* IN.      BS =*,F5.1,* IN.*,
+     5X,*BEAM SPACING =*,F6.1,* IN.*)
641 FØRMAT(*  SLAB*,F10.4,* SQ IN./FT*,F9.3)
C
C  INPUT LØAD PARAMETERS
IF(LDTYPE.EQ.5)GØTØ 20
C  LØCATIØN 1.  FRØNT FACE LØADING (USED IN RØØM-FILLING PRØCEDURE)
IF(KLØAD.EQ.1)GØTØ 25
100 W=1000.0 $ PØ=14.7 $ CØ=1120.0
IF(KRF.NE.1)GØTØ 102
LØC=1
IF(KRAND.EQ.1)GØTØ 106
PRINT 600
```

PROGRAM RCBEAM (CONTINUED)

```
READ,S
GØTØ 103
C  LØCATIØN 2. TØP FACE LØADING
102 CD=0 $ LØC=2
ZLEN=ZLB/12.0
105 IF(KINC.EQ.1)GØTØ 106
PRINT 610
READ,PSØ
PR=2.0*PSØ*(7.0*PØ+4.0*PSØ)/(7.0*PØ+PSØ)
600 FØRMAT(/*INPUT S*,+)
610 FØRMAT(/*INPUT PSØ*,+)
C
C * INPUT RØØM-FILLING PARAMETERS *
106 IF(KRF.EQ.0)GØTØ 20
10 PRINT 700
RHØØ=0.076 $ L1=.FALSE.
DELAY=1E10
READ,NWIN,V3
AT=0$ AFRØNT=0$ ASIDE=0
DØ 18 I=1,NWIN
PRINT 710,I
READ,AA(I,1),NN(I),AA(I,2)
AA(I,2)=AA(I,2)/1000.0
AT=AT+AA(I,1)
M=NN(I)$ GØTØ(12,14,14),M
12 AFRØNT=AFRØNT+AA(I,1)
GØTØ 18
14 ASIDE=ASIDE+AA(I,1)
18 IF(AA(I,2).LT.DELAY)DELAY=AA(I,2)
AFRØNT=AFRØNT/AT$ ASIDE=ASIDE/AT
700 FØRMAT(/*INPUT NWIN AND RØØM VØLUME (CF)*,+)
710 FØRMAT(/*INPUT AREA (SQ FT),LØCATIØN CØDE & DELAY(MSEC)*
+    * FØR WINDØW*,I2,+)
G=1.4 $ G2=1./G $ G3=1.-G2 $ G4=2./G3 $ G5=G+1.
PP2=.1912
C=SQRT(G*PØ*32.*144./RHØØ)
TAU=2.*(V3**(1./3.))/C
DT=TAU/4.0
C
20 CØNTINUE
25 CALL CHAIN(RCBEAM2)
99 STØP
END
SEGMENT RCBEAM2 (INPUT,ØUTPUT,TAPE1)
C  THIS SEGMENT CALCULATES THE RESISTANCE FUNCTIØN AND
C  LØADING AND SØLVES THE DYNAMIC EQUATIØNS ØF MØTIØN
C
CØMMØN KINC,LDTYPE,KRF,KRAND,TIME,I,Y(100),QT,QU,YU,YFAIL,
+  ZLB,BB,HB,FPC,FDY,ICASE,AS(4),APS(4),D(4),DP(4),FDC,
+  EC,ES,QAREA,PAREA,ZMASS,ZKL4,VL1,VL2,
+  NØMB,ASCS,VCL,QØLSLAB,KCØMP,HS,BS,NSLABS,NAMEF(2),KLØAD,
+  ZK,PØ,CØ,LØC,S,ZLEN,CD,PSØ,PDØ,PF,PEXT,PC,TC,TO,DELAY,
+  TT(80,2),PP(80,2),REAC(80,2),INDEX(2),BR(2),
+  NWIN,RHØØ,V3,L1,AA(8,2),NN(8),AFRØNT,ASIDE,G,G2,G3,G4,PP2,DT
CØMMØN /RAND/ TIMEC
DIMENSIØN A(80),V(80),T(80),VS(80),VL(80),PV(80)
C
IF(KINC.NE.1.ØR.LDTYPE.EQ.3)CALL FØRCE(1)
14   IF(KRAND.NE.1)GØTØ 35
     CALL FØRCE(4)
     CALL RANDØM(1)
34   CALL RANDØM(2)
35   CALL RESIST(J)
C
C: MINIMUM, MAXIMUM, AND STARTING VALUES ARE DETERMIN   FØR CASES
C: WHERE THE LØAD CAUSING INCIPIENT CØLLAPSE IS TØ BE FØUND
13   IF(KINC.EQ.0)GØTØ 23
PF=4.0
      PFMAX=0
     PFMIN=0
     GØTØ 20
15    PF=(PFMIN+PFMAX)/2.0
20   CALL FØRCE(2)
23   IF(KRF.EQ.0)GØTØ 24
```

PROGRAM RCBEAM (CONTINUED)

```
      CALL FILL(PINT,2)
C
C:  INITIALIZE VALUES FØR BETA METHØD (BETA=1/6) AND CØMPUTE VALUES
C:  FØR FIRST TIME INTERVAL ASSUMING ELEMENT INITIALLY AT REST
24     I=1
       TIME=0
      V(1)=0 $ Y(1)=0
       DELTA=0.001
      IF(KRF.NE.1)GØTØ 30
27    IF(TIME.GE.(DELAY-.00001))GØTØ 30
      TIME=TIME+DELTA
      CALL FILL(PINT,3)
      GØTØ 27
30    CALL RESIST(2)
      A(1)=0.0 $ VS(1)=0.0 $ VL(1)=0.0
      T(1)=TIME
C
C: PRØCEDURE FØR ALL SUBSEQUENT TIME INTERVALS
1      I=I+1
      IF(I.LT.101)GØTØ 11
      PRINT 98,TIME
98 FØRMAT(/*I=101, TIME =*,F6.3,*, FAILURE ASSUMED TØ NØT ØCCUR*)
      GØTØ 6
11    TIME=TIME+DELTA
       T(I)=TIME
       A(I)=A(I-1)
      IF(KRF.NE.0)GØTØ 10
      CALL FØRCE(3)
      PN(I)=PEXT
      GØTØ 2
 DIMENSIØN A(100),V(100),T(100),VL(100),PN(100)
10    CALL FILL(PINT,3)
      PN(I)=PINT
2 IF(KLØAD.EQ.0)PT=PN(I)*PAREA
 IF(KLØAD.NE.0)PT=PN(I)
      DØ 8 JJ=1,10
       Y(I)=Y(I-1)+DELTA*V(I-1)+DELTA*DELTA*(A(I-1)/3.+A(I)/6.)
       CALL RESIST(2)
4 ANEW=(PT-QT)/(ZMASS*ZKLM)
       ADELTA=ANEW-A(I)
       A(I)=ANEW
 IF(ANEW.EQ.0)PRINT,*1985*,TIME,PT,QT,ZMASS,ZKLM,Y(I),A(I-1)
       IF(ABS(ADELTA/(ANEW+1E-10)).LT.0.01)GØTØ 9
8      CØNTINUE
       A(I)=ANEW-ADELTA/2.0
 PRINT 80,TIME,PF,A(I),Y(I)
9      CØNTINUE
       Y(I)=Y(I-1)+DELTA*V(I-1)+DELTA*DELTA*(A(I-1)/3.+A(I)/6.)
       V(I)=V(I-1)+DELTA*(A(I)+A(I-1))/2.0
 VL(I)=VL1*PT+VL2*QT
C
C: CHECK FØR MAXIMUM DEFLECTIØN ØR FAILURE
C:   IF MAXIMUM DEFLECTIØN REACHED, WALL DID NØT FAIL
       IF(Y(I).LE.Y(I-1).AND.PN(I).LE.PN(I-1))GØTØ 6
       IF(Y(I).LT.0)GØTØ 6
       IF(Y(I).GE.YFAIL)GØTØ 7
       IF(TIME-DELAY.GE.0.010)DELTA=0.002
       IF(TIME-DELAY.GE.0.020)DELTA=0.005
       IF(TIME-DELAY.GE.0.100)DELTA=0.010
       IF(TIME-DELAY.GE.0.500)DELTA=0.050
C:   IF FAILURE DEFLECTIØN REACHED, ELEMENT FAILED
        GØTØ 1
C
C: INTERVAL HALVING PRØCEDURE TØ DETERMINE LØAD CAUSING INCIPIENT
C: CØLLAPSE FØR CASES WHERE DESIRED
C:    ELEMENT DID NØT FAIL -- SET PFMIN TØ PF
6      CØNTINUE
       IF(KINC.EQ.0)GØTØ 18
36     PFMIN=PF
 IF(PFMAX.GT.0)GØTØ 16
       PF=2.0*PF
       GØTØ 20
C:   ELEMENT FAILED -- SET PFMAX TØ PF
7      CØNTINUE
```

PROGRAM RCBEAM (CONTINUED)

```
02350        TIMEC=TIME
02370        IF(KINC.EQ.0)GØTØ 18
02380 37     PFMAX=PF
02390C:    CHECK TØ SEE IF LØAD RANGE IS WITHIN DESIRED ACCURACY
02400 17     IF((PFMAX-PFMIN)/PFMAX.GT.0.01)GØTØ 16
02410        IF(KRAND.NE.1)GØTØ 18
02420        CALL RANDØM(3)
02430        GØTØ 34
02440C
02450C: ØUTPUT DATA INCLUDES THE MAXIMIM DEFLECTIØN AND TIME ØF
02460C: ØCCURANCE FØR A NØN-FAILING ELEMENT ØR THE TIME AND VELØCITY
02470C: AT CØLLAPSE FØR A FAILING ELEMENT.  ØPTIØNAL ØUTPUT IS THE
02480C: ENTIRE BEHAVIØR TIME-HISTØRY.
02490C
02500C: ØUTPUT LØAD DATA
02510 18       ALL FØRCE(4)
02520C
02530C: ØUTPUT FINAL RESULTS
02540 IF(Y(I).LT.YFAIL)PRINT 70,Y(I),T(I)
02550 IF(Y(I).GE.YFAIL)PRINT 71,T(I),V(I)
02577C
02578C CHECK TØ SEE IF ENTIRE TIME-HISTØRY IS DESIRED
02580 PRINT 72
02590 READ,M
02600          IF(M.EQ.0)GØTØ 25
02680 PRINT 76,(T(J),PN(J),A(J),V(J),Y(J),VL(J),J=1,I)
02690 25 PRINT 77
02710C
02740 70 FØRMAT(/*NØ FAILURE - MAX DEFLECTIØN ØF*,F6.2,
02750+     * IN. REACHED AT*F7.3,* SEC*)
02760 71 FØRMAT(/*FAILURE ØCCURRED AT*,F7.3,* SEC (FINAL VELØCITY =*,
02770+     F7.2* IN./SEC)*)
02780 72 FØRMAT(/*IS TIME HISTØRY DESIRED (YES=1, NØ=0)*,+)
02830 76 FØRMAT(/*  TIME  PRESSURE  ACCELERATIØN  VELØCITY  *
02840+     *DISPLACEMENT     VL*,/,
02850+     (F6.3,F9.3,F12.1,F12.2,F12.4,F11.0))
02860 77      FØRMAT(///,7(*-----------*))
02870 80 FØRMAT(/*ACCELERATIØN NØT CØNVERGING AT TIME =*,F6.3,
02880+             * SEC (PF =*,F7.3,* PSI)*/*      A(I) SET EQUAL TØ*,
02890+             F8.1,* (AVG ØF LAST 2 ITERATIØNS)*/*      Y(I) =*,
02900+             F8.4,* IN.*)
02950C
02960 999    STØP
02970        END
10000 SUBRØUTINE FØRCE(IENTRY)
10010C  THIS SUBRØUTINE INPUTS THE LØAD PARAMETERS AND DETERMINES
10020C  THE LØAD AT A GIVEN TIME FØR THE FØLLØWING LØAD TYPES:
10030C      1. IDEALIZED BLAST LØAD (FRØNT ØR SIDE FACE)
10050 CØMMØN KINC,LDTYPE,KRF,KRAND,TIME,I,Y(100),9T,9U,YU,YFAIL,
10052+   ZLB,BB,HB,FPC,FDY,ICASE,AS(4),APS(4),D(4),DP(4),FDC,
10054+   EC,ES,OAREA,PAREA,ZMASS,ZKLM,VL1,VL2,
10055+   MEMB,ASCS,VCL,QDLSLAB,KCØMP,HS,BS,NSLABS,NAMEF(2),KLØAD,
10056+   W,PØ,CØ,LØC,S,ZLEN,CD,PSØ,PDØ,PR,P,PC,TC,TO,DELAY,
10057+   TT(80,2),PP(80,2,,REAC(80,2),INDEX(2),BR(2),
10058+   NWIN,RHØØ,V3,L1,AA(8,2),NN(8),AFRØNT,ASIDE,G,G2,G3,G4,PP2,DT
10080C
10130 IF(LDTYPE.EQ.5)GØTØ 700
10135 IF(KLØAD.EQ.1)GØTØ 500
10140C
10150 GØTØ(215,200,300,4),IENTRY
11000C
11010C  CALCULATE LØAD PRØPERTIES FØR GIVEN PEAK PRESSURE
11030 200 GØTØ(205,210),LØC
11040 205 PSØ=(PR-14.0*PØ+SQRT(196.0*PØ*PØ+196.0*PØ*PR+PR*PR))/16.0
11050 GØTØ 215
11060 210 PSØ=PR
11070 215 PDØ=2.5*PSØ*PSØ/(7.0*PØ+PSØ)
11080 U=CØ*SQRT(1.0+(6.0*PSØ)/(7.0*PØ))
11090 TO=W**0.3333/(2.2399+0.1886*PSØ)
11100 GØTØ(220,225),LØC
11110 220 TC=3.0*S/U
11120 PC=PSØ*(1-TC/TO)*EXP(-TC/TO)+PDØ*(1-TC/TO)**2*EXP(-2*TC/TO)
11130 CD=1.0
11140 RETURN
```

PROGRAM RCBEAM (CONTINUED)

```
11150 225 TA=ZLEN/U
11160 TA2=TA/2.0
11170 TA2TO=TA2/TO
11180 PA=PS0*(1-TA2TO)*EXP(-TA2TO)+CD*PD0*(1-TA2TO)**2*EXP(-2*TA2TO)
11190 RETURN
12000C
12010C  CALCULATE LOAD
12030 300 GOTO(305,310),LOC
12040 305 TTO=TIME/TO
12050 IF(TIME.GT.TC)GOTO 320
12060 P=PC+(TC-TIME)*(PR-PC)/TC
12070 RETURN
12080 310 TTO=(TIME-TA2)/TO
12090 IF(TIME.GT.TA)GOTO 320
12100 P=PA*TIME/TA
12110 RETURN
12120 320 IF(TTO.GE.1.0)GOTO 330
12130 P=PS0*(1-TTO)*EXP(-TTO)+CD*PD0*(1-TTO)**2*EXP(-2*TTO)
12150 RETURN
12160 330 P=0
12170 RETURN
13000C
13010C  PRINT LOAD DATA
13020 4 IF(KINC.EQ.0)GOTO 400
13030 PRINT 640,LDTYPE
13040 GOTO 410
13050 400 PRINT 645,LDTYPE
13060 410 CONTINUE
13070 415 GOTO(420,425),LOC
13080 420 PRINT 650
13090 GOTO 430
13100 425 PRINT 655
13110 430 PRINT 660,W,P0,C0
13120 IF(KRAND.NE.0)RETURN
13130 GOTO(435,440),LOC
13140 435 PRINT 665,S,TC,PR
13150 GOTO 445
13160 440 PRINT 670,ZLEN,TA,PA
13170 445 PRINT 675,U,TO,CD,PS0,PD0
13180 RETURN
13500C
13510C  LOAD TYPE 5 -- ARBITRARY LOAD SHAPE
13520 700 GOTO(710,720,730,740),IENTRY
13530C
13540C  INPUT LOAD DATA
13550 710 PRINT 780
13560 READ,NPOINT,(TT(J,1),PP(J,1),J=1,NPOINT)
13570 FACTOR=1.0
13580 IF(KINC.EQ.0)GOTO 718
13590 PMAX=PP(1,1)
13600 DO 715 J=2,NPOINT
13610 715 IF(PP(J,1).GT.PMAX)PMAX=PP(J,1)
13620 718 PX=PP(2,1)-PP(1,1)
13630 TX=TT(2,1)-TT(1,1)
13640 JJ=1
13650 RETURN
13660C
13670C  CALCULATE MAXIMUM LOAD
13680 720 FACTOR=PR/PMAX
13690 GOTO 718
13700 RETURN
13710C
13720C  CALCULATE LOAD
13730 730 IF(TIME.LE.TT(JJ+1,1))GOTO 735
13740 JJ=JJ+1
13750 PX=PP(JJ+1,1)-PP(JJ,1)
13760 TX=TT(JJ+1,1)-TT(JJ)
13765 IF(TX.EQ.0)TX=1E-10
13770 GOTO 730
13780 735 P=FACTOR*(PP(JJ,1)+(TIME-TT(JJ,1))*PX/TX)
13790 RETURN
13800C
13810C  PRINT LOAD DATA
13815 740 IF(KINC.EQ.1)PRINT 640,LDTYPE
```

```
13820 IF(KINC.EQ.0)PRINT 645,LDTYPE
13825 PRINT 790
13830 DØ 745 J=1,NPØINT
13840 P=FACTØR*PP(J,1)
13850 745 PRINT 795,TT(J,1),P
13860 RETURN
14000C
14070 640 FØRMAT(/*LØAD CAUSING INCIPIENT FAILURE IS AS FØLLØWS:*,
14071+      /,5X,*LØAD TYPE NUMBER*,I2)
14080 645 FØRMAT(/*PRØPERTIES ØF LØAD ACTING ØN WALL ARE AS FØLLØWS:*,
14081+      /,5X,*LØAD TYPE NUMBER*,I2)
14090 650 FØRMAT(5X,*(FRØNT FACE)*)
14100 655 FØRMAT(5X,*(SIDE ØR TØP FACE)*)
14110 660 FØRMAT(10X,*W =*,F8.1,* KT     PØ =*,F6.2,* PSI      CØ =*,
14111+      F7.1,* FPS*)
14120 665 FØRMAT(10X,*S =*,F6.1,* FT       TC =*,F6.3,* SEC       PR =*,
14121+      F7.3,* PSI*)
14130 670 FØRMAT(10X,*L =*,F6.1,* FT       TA =*,F6.3,* SEC       PA =*,
14131+      F7.3,* PSI*)
14140 675 FØRMAT(10X,*U =*,F7.1,* FPS     TO =*,F6.3,* SEC      CD =*,
14141+      F5.1,/,8X,*PSØ =*,F7.3,* PSI   PDØ =*,F7.3,* PSI*)
14150 780 FØRMAT(/*INPUT NUMBER ØF LØAD PØINTS AND THE TIME AND *
14151+      *PRESSURE AT EACH PØINT*)
14160 790 FØRMAT(/10X,*TIME        PRESSURE*)
14170 795 FØRMAT(F15.3,F12.2)
15000C
15002C:   SLAB REACTIØN DATA
15010 500 GØTØ(510, 530,540, 560),IENTRY
15020 510 RETURN
15085C
15090 530 STØP
15095C
15100 540 P=0.0
15110 DØ 555 J=1,NSLABS
15115 545 JJ=INDEX(J)
15120 550 IF(TIME.LE.TT(JJ+1,J))GØTØ 555
15125 INDEX(J)=INDEX(J)+1 $ JJ=INDEX(J)
15130 IF(JJ.LT.NPØINT)GØTØ 552
15132 PRINT 690,TIME
15133 STØP
15135 552 BR(J)=(REAC(JJ+1,J)-REAC(JJ,J))/(TT(JJ+1,J)-TT(JJ,J))
15140 BP=(PP(JJ+1,J)-PP(JJ,J))/(TT(JJ+1,J)-TT(JJ,J))
15150 GØTØ 550
15160 555 P=P+REAC(JJ,J)+(TIME-TT(JJ,J))*BR(J)
15165 P=P+QAREA*(PP(JJ,J)+(TIME-TT(JJ,J))*BP)
15170 RETURN
15180 560 PRINT 680
15190 DØ 565 J=1,NSLABS
15200 565 PRINT 685,NAMEF(J)
15300 680 FØRMAT(/*BEAM LØADED WITH REACTIØNS FRØM FILE(S):*)
15310 685 FØRMAT(10X,A7)
15315 690 FØRMAT(/*END ØF FILE -- BEAM HAS NØT FAILED AT*,F6.3,* SEC*)
15320 RETURN
15330 END
20000 SUBRØUTINE FILL(P3,IENTRY)
20010C: CØMPUTES AVERAGE AIR PRESSURE IN RØØM DUE TØ BLAST WAVE
20020C: INCIDENT HEAD-ØN UPØN FRØNT WALL.
20030C
20050 CØMMØN KINC,LDTYPE,KRF,KRAND,TIME,II,Y(100),DT,DU,YU,YFAIL,
20052+   ZLB,BB,HB,FPC,FDY,ICASE,AS(4),APS(4),D(4),DP(4),FDC,
20054+   EC,ES,QAREA,PAREA,ZMASS,ZKLM,VL1,VL2,
20055+   MEMB,ASCS,VCL,QDLSLAB,KCØMP,HS,BS,NSLABS,NAMEF(2),KLØAD,
20056+   W,PØ,CØ,LØC,S,ZLEN,CD,PSØ,PDØ,PR,PEXT,PC,TC,TO,DELAY,
20057+   TT(80,2),PP(80,2),REAC(80,2),INDEX(2),BR(2),
20058+   NWIN,RHØØ,V3,L1,AA(8,2),NN(8),AFRØNT,ASIDE,G,G2,G3,G4,PP2,DT
20090 LØGICAL L1,L2,L3
20095C
20100 GØTØ(10,13,11),IENTRY
20110 10 RETURN
20310C
20320 13 P3Ø=P3
20330 TT=0.$ TØ=0.
20340 RHØ3Ø=RHØØ
20350 L2=.FALSE. $ L3=.FALSE.
```

PROGRAM RCBEAM (CONTINUED)

```
20360 RETURN
20370C
20380 11 IF(L1)GOTO 52
20385 IF(L2.A.L3)GOTO 9
20390 52 DDT=(TIME-T))*0.5
20395 ISTOP=2
20400 53 IF(DDT.LT.DT)GOTO 51
20410 50 DDT=0.5*DDT
20415 ISTOP=2*ISTOP
20420 GO TO 53
20430 51 CONTINUE
20440 DO 99 I=1,ISTOP
20450 TT=T0+I*DDT
20460 IF(TT.GT.TO)GO TO 99
20470 DM=0. $ WW=0. $ NW=0
20480 DO 500 K=1,NWIN
20490 M=NN(K) $ DLY=AA(K,2)+0.000001
20500 IF(DLY.GE.TT)GO TO 500
20510 GOTO(15,16,16),M
20520 15 CDF=1.0
20530 IF(TT-TC)20,20,21
20540 20  P11=(TC-TT)*(PR-PC)/TC+PC
20550 P11=P11+P0
20560 GO TO 30
20570 16 CDF=-0.4
20600 21  R=TT/TO $ RR=1.-R
20610 PD=PDO*RR*RR*EXP(-2.*R)
20620 PS=PSO*RR*EXP(-R)
20630 P11=PS+CDF*PD
20640 P11=P11+P0
20650 30  RHO1=RHO3*((P11/P3)**G2)
20660 IF(P11-P3)36,36,37
20670 36  ISIGN=-1
20680 L2=.TRUE.
20770 303  P2=P11
20780 RH32=((P2/P33)**G2)*RH33%
20790 X=P30/RH033
20800 GO TO 38
20810 37  ISIGN=+1
20820 306 P2=PP2*P11
20830 RH32=((P2/P11)**G2)*RH31
20840 X=P11/RH31
20850 38  U22=G4*(X-P2/RH32)*32.*144.
20860 IF(U22)40,39,39
20870 40  PRINT,*U22 NEGATIVE*,U22
20880 STOP
20890 39  U2=SQRT(U22)*ISIGN
20900 DDM=U2*RH32*AA(K,1)*DDT
20910 DM=DM+DDM
20920 WW=WW+P11*DDM/(G3*RH31)
20925C
20930 500 CONTINUE
20940 P33=P30+(G-1.)*WW/V3
20950 RH030=RH030+DM/V3
20960 99  CONTINUE
20970 T0=TT
20980 P3=P30-P0
20982 IF(TIME.GE.TC)L3=.TRUE.
20983 RETURN
20984 9 R=TIME/TO $ RR=1.0-R
20985 PD=PDO*RR*RR*EXP(-2.0*R)
20986 PS=PSO*RR*EXP(-R)
20987 P3=PS+PD*(AFRONT-0.4*ASIDE)
20990 999 RETURN
21020 END
30000 SUBROUTINE RESIST (IENTRY)
30010C: THIS SUBROUTINE DETERMINES THE RESISTANCE FUNCTION,
30020C: TRANSFORMATION FACTORS, AND REACTION COEFFICIENTS; AND
30030C: SUPPLIES THE REACTION VALUES FOR SPECIFIC DEFLECTIONS
30040C
30050 COMMON KINC,LDTYPE,KRF,KRAND,TIME,I,Y(100),Q,QU,YU,YFAIL,
30052+   ZL9,BB,H9,FPC,FDY,ICASE,AS(4),APS(4),D(4),DP(4),FDC,
30054+   EC,ES,OAREA,PAREA,ZMASS,ZKLM,VL1,VL2,
30055+   MEMB,ASCS,VCL,ODLSLAB,KCOMP,HS,BS,NSLABS,NAMEF(2),KLOAD,
```

PROGRAM RCBEAM (CONTINUED)

```
30056+   W,PO,CO,LOC,S,ZLEN,CD,PSO,PDO,PR,PEXT,PC,TC,TO,DELAY,
30057+   TT(80,2),PP(80,2),REAC(80,2),INDEX(2),BR(2),
30058+   NWIN,RHOO,V3,L1,AA(8,2),VN(8),AFRONT,ASIDE,G,G2,G3,G4,PP2,OT
30100 REAL N,IC,IG,MM,KK1,KK2,KK3,MU(4),ICR(4),KT,ICC,ICS
30130 GOTO(4,500,45),IENTRY
30140 4 RETURN
30810C
30830C * ENTRY 2:  DETERMINE RESISTANCE FUNCTION *
30840 45 N=ES/EC
30850 FR=8.0*SQRT(FDC)
30860 IG=BB*HB**3/12.+(N-1)*(AS(1)*(D(1)-HB/2.)**2
30861+     +APS(1)*(HB/2.-DP(1))**2)
30870 MM=2.0*IG*FR/HB
30872 IF(KCOMP.EQ.0)GOTO 50
30874 YBART=0.5*(HB*HB*BB+HS*HS*(BS-BB))/(HS*(BS-BB)+HB*BB)
30876 IG=IG+(0.5*HB-YBART)**2+HS**3*(BS-BB)/12.0
30877+     +HS*(BS-BB)*(YBART-0.5*HS)**2
30878 MM=IG*FR/(HB-YBART)
30879 50 IF(KCOMP.EQ.0)BS=BB
30880 CALL MOMENT(FDC,FDY,ES,N,0,BS,AS,APS,D,DP,MU,ICR,IC,KCOMP,HS,BB)
31800C: DETERMINE MOMENT AND DEFLECTION COEFFICIENTS
31810C: FOR CRACKED PORTION OF SLAB BEHAVIOR
31820 106 B=0
31830 ICASE4=ICASE-4
31832 ICC=ICR(1) $ ICS=ICR(3)
31834 CALL COEF(ICASE4,ASS,BSS,AF,BF,ICC,ICS)
31840 GOTO(182,185,190),ICASE4
31850 182 OUTERM=1.0/BSS
31860 GOTO 195
31870 185 OUTERM=(MU(3)/MU(1)+1.0)/BSS
31880 GOTO 195
31890 190 OUTERM=(0.5*MU(3)/MU(1)+1.0)/BSS
31900 195 GOTO(200,210,210),ICASE4
31910C
31920C: *************************************************************
31930C: * DETERMINE RESISTANCE (TOTAL) CURVE FOR WALL                *
31940C: * (Q IS IN UNITS OF LB, KK IN LB/IN., AND Y IN INCHES)       *
31950C: *************************************************************
31960C
31970C: CASE 5
31980 200 Q1=MM/(BSS*ZLB)
31990 KK1=EC*IG/(ASS*ZLB**3)
32000 Y1=Q1/KK1
32010 KK2=EC*ICR(1)/(ASS*ZLB**3)
32050 205 QU=OUTERM*MU(1)/ZLB
32060 208 YU=QU/KK2
32070 GOTO 280
32080C
32090C: CASES 6 AND 7
32100 210 Q1=MM/(BF*ZLB)
32110 KK1=EC*IG/(AF*ZLB**3)
32120 Y1=Q1/KK1
32130 Q2=MU(3)/(BF*ZLB)
32140 KK2=EC*ICR(3)/(AF*ZLB**3)
32150 Y2=Q2/KK2
32160 KK3=EC*ICR(3)/(ASS*ZLB**3)
32200 215 QU=OUTERM*MU(1)/ZLB
32210 220 YU=Y2+(QU-Q2)/KK3
32220 280 CONTINUE
32260 QFAIL=QU
32270 YT=999.9
32280C
32290C: CHECK FOR TYPE OF FAILURE - LIGHTLY REINFORCED OR CONVENTIONAL
32300 IF(MU(1).LT.1.5*MM)GOTO 288
32310C
32320C: CONVENTIONAL TYPE FAILURE
32322 IF(ICASE.EQ.1.OR.ICASE.EQ.5)GOTO 272
32324 YE=Y2+YU*(1.0-Q2/QU)
32326 GOTO 273
32328 272 YE=YU
32330 273 YFAIL=YE*0.1/(AS(1)/(BB*D(1)))
32340C: DUCTILITY FACTOR MUST BE <= 30
32350 IF(YFAIL.GT.30.0*YE)YFAIL=30.0*YE
32370 GOTO 300
```

PROGRAM RCBEAM (CONTINUED)

```
C
C: LIGHTLY REINFORCED TYPE OF FAILURE
C:   THE FOLLOWING EXPRESSION IS BASED ON A STEEL ELONGATION OF 20%
 298 UCOEF=30.
 PRINT 605
 READ, NOBAR
 UPU=UCOEF*SQRT(FDC)
 ABAR=3.14159*(NOBAR/16.)**2
 290 YFAIL=SQRT((0.2*ABAR*FDY/UPU+ZLB/2.)**2-(ZLB/2.)**2)
C
C  TENSILE MEMBRANE BEHAVIOR
 300 IF(MEMB.NE.1)GOTO 285
 312 KT=8.0*1.5
 TS=ASCS*FDY
 314 YT=QU*ZLB/(KT*TS)
 QT=QU
 IF(YT.LE.YFAIL)GOTO 316
 YT=YFAIL
 QT=YT*KT*TS/ZLB
 316 IF(YFAIL.LT.0.15*ZLB)YFAIL=0.15*ZLB
 QFAIL=0.15*KT*TS
C ADJUST LOAD-DEFLECTION CURVE FOR BEAM DEAD LOAD
 285 QDL=QDLSLAB+150.0*HB*BB*ZLB/1728.0
 IF(QDL.GT.Q1)GOTO 292
 YDL=QDL/KK1
 GOTO 295
 292 GOTO(293,294,294),ICASE4
 293 YDL=Y1+(QDL-Q1)*(YU-Y1)/(QU-Q1)
 291 IF(QDL.LT.QU)GOTO 295
 PRINT,*QDL =*,QDL,*  QU =*,QU $ STOP
 294 YDL=Y1+(QDL-Q1)*(Y2-Y1)/(Q2-Q1)
 IF(QDL.LT.Q2)GOTO 295
 YDL=Y2+(QDL-Q2)*(YU-Y2)/(QU-Q2) $ GOTO 291
 295 Y1=Y1-YDL$ Y2=Y2-YDL$ YU=YU-YDL$ YT=YT-YDL$ YFAIL=YFAIL-YDL
 Q1=Q1-QDL$ Q2=Q2-QDL$ QU=QU-QDL$ QT=QT-QDL$ QFAIL=QFAIL-QDL
 IF(KRAND.NE.1)PRINT 633,QDL,YDL
C
C: OUTPUT LOAD-DEFLECTION CURVE
 IF(KRAND.EQ.1)GOTO 335
 PRINT 650
 IF(ICASE.EQ.5)GOTO 320
 PRINT 660,Q1,Y1,Q2,Y2
 GOTO 330
 320 PRINT 660,Q1,Y1
 330 IF(MEMB.EQ.1)GOTO 332
 PRINT 660,QU,YU,QFAIL,YFAIL
 GOTO 335
 332 IF(QT.NE.QU)GOTO 333
 PRINT 660,QU,YU,QT,YT,QFAIL,YFAIL
 GOTO 335
 333 PRINT 660,QU,YU,QU,YT,QT,YT,QFAIL,YFAIL
 335 CONTINUE
C
 CALL TRANS(ICASE4,ZKLM,ZKLMSE,ZKLMFE,ZKLMP,VL1S,VL2S,VL1F,
+             VL2F,VL1P,VL2P)
 QSHRL=VCL*D(1)*BB
 340 IF(KRAND.NE.1)PRINT 695,QSHRL
 RETURN
C
C: ********************************************************
C: * ENTRY 3: DETERMINE THE RESISTANCE (PER UNIT AREA)  *
C: *          OF THE WALL AS A FUNCTION OF Y(I)          *
C: ********************************************************
C
 500 IF(Y(I).GE.YFAIL)GOTO 560
 IF(Y(I).GT.YU)GOTO 540
 GOTO(501,520,520,520,501,520,520),ICASE
 501 CONTINUE
C
C: ELASTIC RANGE -- CASES 1 AND 5
 ZKLM=ZKLMSE
 VL1=VL1S $ VL2=VL2S
 IF(Y(I).GT.Y1)GOTO 510
C
```

PROGRAM RCBEAM (CONTINUED)

```
C: UNCRACKED PORTION -- ALL CASES
 505 Q=Y(I)*KK1
 RETURN
C
C: CRACKED PORTION -- CASES 1 AND 5
 510 Q=Q1+(Y(I)-Y1)*(QU-Q1)/(YU-Y1)
 RETURN
C
 520 IF(Y(I).GT.Y2)GOTO 530
C
C: ELASTIC RANGE -- CASES 2,3,4,6,7
 ZKLM=ZKLMEF
 VL1=VL1F $ VL2=VL2F
 IF(Y(I).LT.Y1)GOTO 505
C: CRACKED PORTION -- CASES 2,3,4,6,7
 Q=Q1+(Y(I)-Y1)*(Q2-Q1)/(Y2-Y1)
 RETURN
C
C: ELASTO-PLASTIC RANGE -- CASES 2,3,4,6,7
 530 ZKLM=ZKLMSE
 VL1=VL1S $ VL2=VL2S
 Q=Q2+KK3*(Y(I)-Y2)
 RETURN
C
C: PLASTIC RANGE -- ALL CASES
 540 ZKLM=ZKLMP
 VL1=VL1P $ VL2=VL2P
 IF(Y(I).GT.YT)GOTO 550
 Q=QU
 RETURN
C
C  TENSILE MEMBRANE RANGE -- ALL CASES
 550 Q=QT+(Y(I)-YT)*(QFAIL-QT)/(YFAIL-YT)
 RETURN
C
C: WALL COLLAPSED - NO RESISTANCE (TO AVOID NUMERICAL DIFFICULTIEES
C: FOR CERTAIN CASES SET RESISTANCE EQUAL TO VERY SMALL VALUE)
 560 Q=1E-10
 RETURN
C
 605 FORMAT(/*INPUT BAR NUMBER OF REINFORCEMENT*,:)
 633 FORMAT(/*QDL =*,F10.2,* LB    YDL =*,F8.4,* IN.*)
 650 FORMAT(//*LOAD-DEFLECTION CURVE*,/,4X,*QT (LB)        Y (IN.)*)
 660 FORMAT(F10.2,F12.4)
 695 FORMAT(/*QSHRL =*,F11.2,* LB*)
 END
 SUBROUTINE MOMENT(FDC,FDY,ES,N,PV,B,AS,APS,D,DP,MU,ICR,IC,
+                   KCOMP,HS,BP)
C THIS SUBROUTINE DETERMINES THE ULTIMATE MOMENT CAPACITY AND
C CRACKED MOMENT OF INERTIA  FOR REQUIRED SECTIONS
 REAL K1,K2,K3,KUD,N,IC,ICTOT,MU(4),ICR(4),AS(4),APS(4),D(4),DP(4)
C
C: DETERMINE VALUES OF CONCRETE PARAMETERS
 45 K1=0.94-FDC/26E3
 K2=0.50-FDC/8E4
 K3=(3900.0+0.35*FDC)/(3E3+0.82*FDC-FDC*FDC/26E3)
 EPSC=0.004-FDC/65E5
C: ************************************************
C: * DETERMINE ULTIMATE MOMENT CAPACITY AND CRACKED *
C: * MOMENT OF INERTIA FOR REQUIRED SECTIONS       *
C: ************************************************
C
 II=0$ ICTOT=0
 DO 170 I=1,4
 IF(AS(I).EQ.0)GOTO 170
 II=II+1
C: ALL PROPERTIES ARE COMPUTED FOR A WIDTH B
 TENS=AS(I)*FDY+PV
 IF(APS(I).LE.0)GOTO 150
C
C: WALL HAS COMPRESSION REINFORCEMENT
 C=K1*K3*FDC*B*DP(I)
 TERM1=0.5*(TENS/APS(I)+ES*EPSC)
 TERM2=ES*EPSC*(TENS-C)/APS(I)
```

PROGRAM RCBEAM (CONTINUED)

```
C: DETERMINE LOCATION OF NEUTRAL AXIS
IF(TENS.LE.C)GOTO 140
C
C: KUD > D'
FPS=TERM1+K3*FDC/2.0-SQRT((TERM1+K3*FDC/2.0)**2
+   -(TERM2+ES*EPSC*K3*FDC))
C: F'S MUST BE <= FDY
IF(FPS.LT.FDY)GOTO 130
FPS=FDY
130 TPS=APS(I)*(FPS-K3*FDC)
KUD=(TENS-TPS)/(K1*K3*FDC*B)
MU(I)=(TENS-TPS)*(D(I)-K2*KUD)+TPS*(D(I)-DP(I))
ICR(I)=B*KUD**3/3.0+N*AS(I)*(D(I)-KUD)**2
+  +(N-1)*APS(I)*(KUD-DP(I))**2
GOTO 152
C
C: KUD < D'
140 FPS=-TERM1+SQRT(TERM1**2-TERM2)
C: F'S MUST BE <= FDY
IF(FPS.LT.FDY)GOTO 145
FPS=FDY
145 TERM3=TENS+APS(I)*FPS
KUD=TERM3/(K1*K3*FDC*B)
MU(I)=TERM3*(D(I)-K2*KUD)-APS(I)*FPS*(D(I)-DP(I))
ICR(I)=B*KUD**3/3+N*AS(I)*(D(I)-KUD)**2+N*APS(I)*(DP(I)-KUD)**2
GOTO 152
C
C: WALL HAS NO COMPRESSION REINFORCEMENT
150 KUD=TENS/(K1*K3*FDC*B)
MU(I)=TENS*(D(I)-K2*KUD)
ICR(I)=B*KUD**3/3.0+N*AS(I)*(D(I)-KUD)**2
C
152 IF(KCOMP.EQ.0)GOTO 155
IF(I.EQ.3)GOTO 170
IF(KUD.LT.HS)GOTO 170
C
C * TEE BEAM -- NEUTRAL AXIS OUTSIDE FLANGE *
C   (USE EQUIVALENT RECTANGULAR STRESS BLOCK)
ASF=0.85*FDC*(B-BP)*HS/FDY
KUD=(AS(I)-ASF)*FDY/(0.85*FDC*BP)
MU(I)=ASF*FDY*(D(I)-0.5*HS)+(AS(I)-ASF)*FDY*(D(I)-0.5*KUD)
ICR(I)=BP*KUD**3/3.0+(B-BP)*HS**3/12.0+HS*(B-BP)*(KUD-0.5*HS)**2
+         +N*AS(I)*(D(I)-KUD)**2
155 ICTOT=ICTOT+ICR(I)
170 CONTINUE
C
C: DETERMINE AVERAGE CRACKE    ENT OF INERTIA
175 IC=ICTOT/II
RETURN
END
SUBROUTINE COEF (ICASE4,ASS,BSS,AF,BF,ICEN,ISUP)
C
C  THIS SUBROUTINE DETERMINES DEFLECTION AND MOMENT COEFFICIENTS
C  FOR TEE BEAMS WITH VARIABLE MOMENT OF INERTIA
C
REAL ICEN,ISUP
C:  CASE 5. ONE-WAY SIMPLY SUPPORTED WALL
      ASS=5.0/384.0
       BSS=0.125
RI=ISUP/ICEN
      GOTO(270,60,70),ICASE4
C
C:  CASE 6. ONE-WAY FIXED END WALL
60 BF=(.00953+.03213*RI)/(.211+.289*RI)
AF=.00132+.01169*RI-BF*(.0223+.1028*RI)
RETURN
C
C:  CASE 7. ONE-WAY PROPPED CANTILEVER WALL
70 BF=(.0109+.0308*RI)/(.1926+.1407*RI)
AF=-.00439-.00342*RI+BF*(.08333+.02083*RI)
270  RETURN
     END
SUBROUTINE TRANS (ICASE4,ZKLM,ZKLMSE,ZKLMFE,ZKLMP,VL1S,VL2S,
+     VL1F,VL2F,VL1P,VL2P)
```

PROGRAM RCBEAM (CONTINUED)

```
50020C
50030C  THIS SUBRØUTINE DETERMINES TRANSFØRMATIØN FACTØRS AND
50040C  DYNAMIC REACTIØN CØEFFICIENTS FØR ØNE-WAY BEAMS
50050C
50060C  ALL CASES
50070 ZKLMSE=0.78
50080 ZKLMP=0.66
50090 VL1S=0.107
50100 VL2S=0.393
50110 VL1P=0.125
50120 VL2P=0.375
50130 GØTØ(300,310,320),ICASE4
50140C
50150C  CASE 5
50160 300 ZKLM=ZKLMSE
50170 VL1=VL1S
50180 VL2=VL2S
50190 RETURN
50200C
50210C  CASE 6
50220 310 ZKLMFE=0.77
50230 VL1F=0.136
50240 VL2F=0.364
50250 GØTØ 330
50260C
50270C  CASE 7
50280 320 ZKLMFE=0.78
50290 VL1F=0.165
50300 VL2F=0.459
50310 330 ZKLM=ZKLMFE
50320 VL1=VL1F
50330 VL2=VL2F
50340 RETURN
50350 END
70000       SUBRØUTINE RANDØM (IENTRY)
70010C THIS SUBRØUTINE INPUTS MEAN AND STANDARD DEVIATIØNS FØR RANDØM
70020C VARIABLES;  GENERATES RANDØM VALUES;  AND CØNTRØLS REQUIRED
70030C NUMBER ØF CASES TØ BE RUN;  AND ØUTPUTS FINAL RESULTS AND SUMMARY
70040C
70050 CØMMØN KINC,LDTYPE,KRF,KRAND,TIME,II,Y(100),QT,QU,YU,YFAIL,
70052+  ZLB,BB,HB,FPC,FDY,ICASE,AS(4),APS(4),D(4),DP(4),FDC,
70054+  EC,ES,QAREA,PAREA,ZMASS,ZKLM,VL1,VL2,
70055+  MEMB,ASCS,VCL,QDLSLAP,KCØMP,HS,BS,NSLABS,NAMEF(2),KLØAD,
70056+  W,PØ,CØ,LØC,S,ZLEN,CO,PS3,PDØ,PR,PEXT,PC,TC,TO,DELAY,
70057+  TT(80,2),PP(80,2),REAC(80,2),INDEX(2),BR(2),
70058+  NWIN,RHØØ,V3,L1,AA(8,2),NN(8),AFRØNT,ASIDE,G,G2,G3,G4,PP2,DT
70080      CØMMØN /RAND/ TIMEC
70090      DIMENSIØN CHI25(7),CHI975(7),TDIST(7)
70100C
70110C VALUES FØR 97.5%  (F=19,24,29,34,39,44,49)
70120      DATA CHI25/.4088,.5167,.5533,.5825,.6065,.6267,.6440/
70130      DATA CHI975/1.7295,1.6402,1.5766,1.5284,1.4903,1.4591,1.4331/
70140      DATA TDIST/2.093,2.064,2.045,2.032,2.022,2.016,2.010/
70150C
70160      GØTØ(5,50,70),IENTRY
70170    5 XDUMMY=XNØRM1(-1.0,0.0,1.0)
70180C INITIALIZE RANDØM NUMBER GENERATØR
70190      PRINT,/,*INPUT NRAND*,
70200      READ,NRAND
70210      DØ 47 I=1,NRAND
70220      XDUMMY=XNØRM1(0.0,0.0,1.0)
70230   47 CØNTINUE
70240      INDEX=0$  SPSØ=0$  SSPSØ=0
70250      ICHECK=20
70260C
70270C  INPUT MEAN AND STANDARD DEVIATIØN FØR RANDØM VARIABLES
70275      IF(LØC.EQ.2)GØTØ 30
70280      PRINT 87
70290      READ,SMEAN,SSD
70410C  REINFØRCED CØNCRETE WALLS
70420   30 PRINT 86
70430      READ,FDYMEAN,FDYSD
70440      IF(LØC.EQ.1)PRINT 94
70445      IF(LØC.NE.1)PRINT 95
```

PROGRAM RCBEAM (CONTINUED)

```
      RETURN
C
C  GENERATE RANDØM VALUES
   50 FDY=XNØRM1(0.0,FDYMEAN,FDYSD)
      IF(FDY.LE.0)GØTØ 50
      IF(LØC.EQ.2.ØR.SMEAN.EQ.0)GØTØ 65
   60 S=XNØRM1(0.0,SMEAN,SSD)
      IF(S.LE.0)GØTØ 60
   65 INDEX=INDEX+1
      RETURN
C SUM VALUES ØF PSØ AND PSØ**2 FØR USE IN STATISTICAL ANALYSIS
   70 SPSØ=SPSØ+PSØ
      SSPSØ=SSPSØ+PSØ*PSØ
C
C ØUTPUT FINAL RESULTS
   76 IF(LØC.EQ.1)PRINT 92,FDY,S,PSØ,TIMEC
      IF(LØC.NE.1)PRINT 90,FDY,PSØ,TIMEC
   80 IF(INDEX.LT.ICHECK)RETURN
C
C DETERMINE MEAN, STANDARD DEVIATIØN, AND STANDARD ERRØR FØR PSØ
      ZNØ=INDEX
      ZMEAN=SPSØ/ZNØ
      SD=SQRT((SSPSØ-ZNØ*ZMEAN*ZMEAN)/ZNØ)
      STDERR=SD/(SQRT(ZNØ-1))
C CHECK IF MAXIMUM ØF 50 PSØ SAMPLES ØBTAINED
      IF(INDEX.EQ.50)GØTØ 62
C CHECK IF 95% CØNFIDENCE INTERVAL FØR MEAN PSØ VALUE IS
      IF(STDERR*TDIST((INDEX-15)/5)/ZMEAN.GT.0.10)GØTØ 61
C
C CØNFIDENCE INTERVAL IS WITHIN 10% -- DETERMINE UPPER LIMIT ØF
C 95% CØNFIDENCE INTERVAL FØR STANDARD DEVIATIØN
C PRØBABILITY VALUE AND ITS 95% CØNFIDENCE INTERVAL UPPER LIMIT
   62 SDU=SD/(SQRT(CHI25((INDEX-15)/5)))
C CHECK IF MAXIMUM ØF 50 PSØ SAMPLES ØBTAINED
      IF(INDEX.EQ.50)GØTØ 53
C CHECK IF UPPER VALUE ØF 95% CØNFIDENCE INTERVAL FØR STANDARD
C DEVIATIØN IS WITHIN 0.10*MEAN ØF THE STANDARD DEVIATIØN
      IF(((SDU-SD)/ZMEAN).GT.0.10)GØTØ 61
C
C 95% CØNFIDENCE INTERVAL IS WITHIN 10% FØR BØTH MEAN AND 90%
C PRØBABILITY VALUE -- THEREFØRE SUFFICIENT SAMPLES ØBTAINED
C DETERMINE 95% CØNFIDENCE INTERVALS FØR MEAN, STANDARD DEVIATIØN
.
C AND 10% AND 90% PRØBABILITY VALUES
   53 ZMEANL=ZMEAN-STDERR*TDIST((INDEX-15)/5)
      ZMEANU=ZMEAN+STDERR*TDIST((INDEX-15)/5)
      SDL=SD/(SQRT(CHI975((INDEX-15)/5)))
      P10=ZMEAN-1.282*SD
      P10L=ZMEAN-1.282*SDU
      P10U=ZMEAN-1.282*SDL
      P90=ZMEAN+1.282*SD
      P90L=ZMEAN+1.282*SDL
      P90=ZMEAN+1.282*SD
      P90U=ZMEAN+1.282*SDU
      P90U=ZMEAN+1.282*SDU
C
C ØUTPUT STATISTICAL PARAMETERS ØF INCIPIENT CØLLAPSE PRESSURE
      PRINT 100,ZMEAN,ZMEANL,ZMEANU,SD,SDL,SDU,P10,P10L,P10U,
     +  P90,P90L,P90U
      PRINT 105,INDEX,STDERR
      GØTØ 999
C
C 95% CØNFIDENCE INTERVAL IS NØT WITHIN 10% FØR BØTH MEAN AND 90
B
C VALUES -- THEREFØRE ØBTAIN 5 ADDITIØNAL SAMPLES
   61 ICHECK=ICHECK+5
      RETURN
C
   86 FØRMAT(/*INPUT MEAN AND STANDARD DEVIATIØN FØR FDY*,*)
   87 FØRMAT(/*INPUT MEAN AND STANDARD DEVIATIØN FØR S*,*)
   90 FØRMAT(F9.2,F10.2,F14.3)
   92 FØRMAT(F9.1,F11.2,F10.2,F14.3)
   95 FØRMAT(///,5X,*FDY*,7X,*PSØ*,6X,*CØLLAPSE TIME*)
   96 FØRMAT(///,5X,*FDY*,9X,*S*,9X,*PSØ*,6X,*CØLLAPSE TIME*)
```

PROGRAM RCBEAM (CONCLUDED)

```
  100 FORMAT(///,11X,*STATISTICAL PROPERTIES OF INCIPIENT PSO*,
     //,39X,*95% CONFIDENCE LIMITS*,/,7X,*ITEM*,18X,
     *VALUE        LOWER        UPPER*,//,* MEAN*,F29.2,
     2F12.2,//,* STANDARD DEVIATION*,F15.2,2F12.2,//,
     * 10% PROBABILITY VALUE*,3F12.2,//,
     * 90% PROBABILITY VALUE*,3F12.2)
  105 FORMAT(//,5X,*NUMBER OF OBSERVATIONS =*,I3,/,5X,
     *STANDARD ERROR =*,F5.2)
C
  999 STOP$ END
FUNCTION XNORM1(X,A,B)
IF(X)10,20,20
10 X1=RANF(-1.0)
20 X1=RANF(0.0)
X2=RANF(0.0)
Y=SQRT(-2.0*ALOG(X1))*(COS(6.283184*X2))
XNORM1=A+Y*B
RETURN
END
```

STBEAM

Steel Support Beam

PROGRAM STBEAM

```
PROGRAM STBEAM1(INPUT,OUTPUT,TAPE1)
CALL RETR(7HSTBEAM2,7HSTBEAM2)
C * THIS PORTION OF THE PROGRAM INPUTS THE REQUIRED ELEMENT
C:  AND LOAD DATA AND INITIALIZES CERTAIN PARAMETERS *
C
COMMON KINC,LDTYPE,KRF,KRAND,TIME,I,Y(100),DT,DU,YU,YFAIL,
+   ZLB,HB,BFLG,TFLG,TW,FDYS,ICASE,KPLATE,KCAMP,HP,BP,ES,
+   HPC,HS,FPC,FDC,EC,BS,FDYR,AS,D,APS,DP,QAREA,PAREA,ZMASS,
+   ZKLM,VL1,VL2,NSLABS,NAMEF(2),KLOAD,QDLSLAB,
+   W,PO,CO,LOC,S,ZLEN,CD,PSO,PDO,PR,PEXT,PC,TC,TO,DELAY,
+   TT(80,2),PP(80,2),REAC(80,2),INDEX(2),BR(2),
+   VMIN,RHOM,V3,L1,AA(8,2),VV(8),AFRONT,ASIDE,G,G2,G3,G4,PPR,DT
LOGICAL L1
C
C * READ TITLE AND CONTROL PARAMETERS
PRINT 67
READ 68,TITLE
PRINT 720
READ,NSLABS,KLOAD
PAREA=0 $ QDLSLAB=0
IF(KLOAD.EQ.0)GOTO 40
C
C:  INPUT SLAB REACTION DATA FILE DATA
PRINT 725
DO 39 J=1,NSLABS
PRINT 735,J
READ,NAMEF(J),ISIDE
CALL PFUR(3HRET,1,NAMEF(J))
IF(ISIDE.EQ.1)READ(1,)SAREA,DUM,HS
IF(ISIDE.EQ.2)READ(1,)DUM,SAREA,HS
510 INDEX(J)=1
READ(1,)NPOINT
IF(ISIDE.EQ.2)GOTO 520
515 READ(1,)(TT(JJ,J),PP(JJ,J),REAC(JJ,J),DUM,JJ=1,NPOINT)
GOTO 525
520 READ(1,)(TT(JJ,J),PP(JJ,J),DUM,REAC(JJ,J),JJ=1,NPOINT)
525 BR(J)=(REAC(2,J)-REAC(1,J))/(TT(2,J)-TT(1,J))
BP=(PP(2,J)-PP(1,J))/(TT(2,J)-TT(1,J))
REWIND 1
CALL DROP1(1)
QDLSLAB=QDLSLAB+150.0*SAREA*HS/1728.0
39 PAREA=PAREA+SAREA
KINC=0 $ LDTYPE=5 $ KRF=0 $ KRAND=0
GOTO 45
C
C INPUT TRIBUTARY SLAB DATA
40 PRINT 730
DO 42 J=1,NSLABS
PRINT 735,J
READ,SAREA,HS
QDLSLAB=QDLSLAB+150.0*SAREA*HS/12.0
42 PAREA=PAREA+SAREA*144.0
C
PRINT 85
READ,KINC,LDTYPE,KRF,KRAND
45 CONTINUE
DELAY=0
67 FORMAT(/*INPUT TITLE*,+)
68 FORMAT(A59)
85 FORMAT(/*INPUT KINC,LDTYPE,KRF,KRAND(1=RANDOM)*,+)
720 FORMAT(/*INPUT NUMBER OF SLABS SUPPORTED BY BEAM, AND IF *,
+     *SLAB REACTIONS*/*ARE TO BE CALCULATED (0) OR READ FROM *,
+     *DATA FILE (1)*,+)
725 FORMAT(/*INPUT REACTION DATA FILE NAME AND SIDE *,
+     *(1=SHORT,2=LONG)*)
730 FORMAT(/*INPUT CONTRIBUTARY AREA (SQ FT) AND THICKNESS (IN.)*)
735 FORMAT(6X,*FOR SLAB NO.*,I2,+)
C
C  INPUT LOAD PARAMETERS
IF(LDTYPE.EQ.5)GOTO 20
C  LOCATION 1.  FRONT FACE LOADING (USED IN ROOM-FILLING PROCEDURE)
IF(KLOAD.EQ.1)GOTO 25
100 W=1000.0 $ PO=14.7 $ CO=1120.0
IF(KRF.NE.1)GOTO 102
```

PROGRAM STBEAM (CONTINUED)

```
LOC=1
IF(KRAND.EQ.1)GOTO 106
PRINT 600
READ,S
GOTO 105
C  LOCATION 2. TOP FACE LOADING
102 CD=0 $ LOC=2
ZLEN=ZLB/12.0
105 IF(KINC.EQ.1)GOTO 106
PRINT 610
READ,PSO
PR=2.0*PSO*(7.0*PO+4.0*PSO)/(7.0*PO+PSO)
600 FORMAT(/*INPUT S*,+)
610 FORMAT(/*INPUT PSO*,+)
C
C * INPUT ROOM-FILLING PARAMETERS *
106 IF(KRF.EQ.0)GOTO 20
10 PRINT 700
RHOO=0.076 $ L1=.FALSE.
DELAY=1E10
READ,NWIN,V3
AT=0$ AFRONT=0$ ASIDE=0
DO 18 I=1,NWIN
PRINT 710,I
READ,AA(I,1),NN(I),AA(I,2)
AA(I,2)=AA(I,2)/1000.0
AT=AT+AA(I,1)
M=NN(I)$ GOTO(12,14,14),M
12 AFRONT=AFRONT+AA(I,1)
GOTO 18
14 ASIDE=ASIDE+AA(I,1)
18 IF(AA(I,2).LT.DELAY)DELAY=AA(I,2)
AFRONT=AFRONT/AT$ ASIDE=ASIDE/AT
700 FORMAT(/*INPUT NWIN AND ROOM VOLUME (CF)*,+)
710 FORMAT(/*INPUT AREA (SQ FT),LOCATION CODE & DELAY(MSEC)*
+      * FOR WINDOW*,I2,+)
G=1.4 $ G2=1./G $ G3=1.-G2 $ G4=2./G3 $ G5=G+1.
PP2=.1912
C=SQRT(G*PO*32.*144./RHOO)
TAU=2.*(V3**(1./3.))/C
DT=TAU/4.0
C
20 CONTINUE
25 CALL CHAIN(STREAM2)
99 STOP
END
SEGMENT STREAM2(INPUT,OUTPUT,TAPE1)
C  THIS SEGMENT CALCULATES THE RESISTANCE FUNCTION AND
C  LOADING AND SOLVES THE DYNAMIC EQUATIONS OF MOTION
C
COMMON KINC,LOTYPE,KRF,KRAND,TIME,I,Y(100),DT,DU,YU,YFAIL,
+  ZLB,HB,BFLG,TFLG,TW,FDYS,ICASE,KPLATE,KCOMP,NP,BP,ES,
+  HPC,HS,FPC,FDC,EC,BS,FDYR,AS,D,APS,DF,OAREA,PAREA,ZMASS,
+  ZKLM,VL1,VL2,NSLABS,NAMEF(2),KLOAD,OULSLAB,
+  W,PO,CO,LOC,S,ZLEN,CD,PSO,POO,PF,PEXT,PC,TC,TO,DELAY,
+  TT(80,2),PP(80,2),REAC(80,2),INDEX(2),BR(2),
+  NWIN,RHOO,V3,L1,AA(8,2),NN(8),AFRONT,ASIDE,G,G2,G3,G4,PP2,DT
COMMON /RAND/ TIMEC
DIMENSION A(100),V(100),T(100),VS(100),VL(100),PV(100)
C
      CALL RESIST(1)
IF(KINC.NE.1.OR.LOTYPE.EQ.5)CALL FORCE(1)
14    IF(KRAND.NE.1)GOTO 35
      CALL FORCE(4)
      CALL RANDOM(1)
34    CALL RANDOM(2)
35    CALL RESIST(3)
C
C: MINIMUM, MAXIMUM, AND STARTING VALUES ARE DETERMINED FOR CASES
C: WHERE THE LOAD CAUSING INCIPIENT COLLAPSE IS TO BE FOUND
13    IF(KINC.EQ.0)GOTO 23
PF=4.0
       PFMAX=0
       PFMIN=PF/2.0
```

PROGRAM STBEAM (CONTINUED)

```
      GØTØ 20
16     PF=(PFMIN+PFMAX)/2.0
20    CALL FØRCE(2)
23    IF(KRF.EQ.0)GØTØ 24
      CALL FILL(PINT,2)
C
C:  INITIALIZE VALUES FØR BETA METHØD (BETA=1/6) AND CØMPUTE VALUES
C:  FØR FIRST TIME INTERVAL ASSUMING ELEMENT INITIALLY AT REST
24     I=1
       TIME=0
      V(1)=0 $ Y(1)=0
       DELTA=0.001
      IF(KRF.NE.1)GØTØ 30
27    IF(TIME.GE.(DELAY-.00001))GØTØ 30
      TIME=TIME+DELTA
      CALL FILL(PINT,3)
      GØTØ 2
30    CALL RESIST(2)
      A(1)=0.0 $ VS(1)=0.0 $ VL(1)=0.0
      T(1)=TIME
C
C: PRØCEDURE FØR ALL SUBSEQUENT TIME INTERVALS
1      I=I+1
      IF(I.LT.101)GØTØ 11
      PRINT 98,TIME
98 FØRMAT(/*I=101: TIME =*,F6.3,*: FAILURE ASSUMED TØ NØT ØCCUR*)
      GØTØ 6
11    TIME=TIME+DELTA
       T(I)=TIME
       A(I)=A(I-1)
      IF(KRF.NE.0)GØTØ 10
      CALL FØRCE(2)
      PV(I)=PEXT
      GØTØ 2
10    CALL FILL(PINT,3)
      PV(I)=PINT
2 IF(KLØAD.EQ.0)PT=PV(I)*PAREA
IF(KLØAD.NE.0)PT=PV(I)
      DØ 8 J=1,10
       Y(I)=Y(I-1)+DELTA*V(I-1)+DELTA*DELTA*(A(I-1)/3.+A(I)/6.)
       CALL RESIST(2)
 0 ANEW=(PT-QT)/(ZMASS*ZKLM)
       ADELTA=ANEW-A(I)
       A(I)=ANEW
IF(ANEW.EQ.0)PRINT,*1985*,TIME,PT,QT,ZMASS,ZKLM,Y(I),A(I-1)
      IF(ABS(ADELTA/(ANEW+1E-10)).LT.0.01) GØTØ 9
8      CØNTINUE
       A(I)=ANEW-ADELTA/2.0
PRINT 50,TIME,PF,A(I),Y(I)
9      CØNTINUE
       Y(I)=Y(I-1)+DELTA*V(I-1)+DELTA*DELTA*(A(I-1)/3.+A(I)/6.)
       V(I)=V(I-1)+DELTA*(A(I)+A(I-1))/2.0
VL(I)=VL1*PT+VL2*QT
C
C: CHECK FØR MAXIMUM DEFLECTIØN ØR FAILURE
C:    IF MAXIMUM DEFLECTIØN REACHED, WALL DID NØT FAIL
       IF(Y(I).LE.Y(I-1).AND.PV(I).LE.PV(I-1))GØTØ 6
       IF(Y(I).LT.0)GØTØ 6
       IF(Y(I).GE.YFAIL)GØTØ 7
       IF(TIME-DELAY.GE.0.010)DELTA=0.002
       IF(TIME-DELAY.GE.0.020)DELTA=0.005
       IF(TIME-DELAY.GE.0.100)DELTA=0.010
       IF(TIME-DELAY.GE.0.500)DELTA=0.050
C:    IF FAILURE DEFLECTIØN REACHED, ELEMENT FAILED
        GØTØ 1
C
C: INTERVAL HALVING PRØCEDURE TØ DETERMINE LØAD CAUSING INCIPIENT
C: CØLLAPSE FØR CASES WHERE DESIRED
C:    ELEMENT DID NØT FAIL -- SET PFMIN TØ PF
6      CØNTINUE
       IF(KINC.EQ.0)GØTØ 18
36     PFMIN=PF
IF(PFMAX.GT.0)GØTØ 16
        PF=2.0*PF
```

PROGRAM STBEAM (CONTINUED)

```
02320         GOTO 20
02330C:    ELEMENT FAILED -- SET PFMAX TO PF
02340 7     CONTINUE
02350       TIMEC=TIME
02370       IF(KINC.EQ.0)GOTO 18
02380 17    PFMAX=PF
02390C:    CHECK TO SEE IF LOAD RANGE IS WITHIN DESIRED ACCURACY
02400 11     IF((PFMAX-PFMIN)/PFMIN.GT.0.01)GOTO 16
02410       IF(KRAND.NE.1)GOTO 18
02420       CALL RANDOM(3)
02430       GOTO 34
02440C
02450C: OUTPUT DATA INCLUDES THE MAXIMUM DEFLECTION AND TIME OF
02460C: OCCURANCE FOR A NON-FAILING ELEMENT OR THE TIME AND VELOCITY
02470C: AT COLLAPSE FOR A FAILING ELEMENT.  OPTIONAL OUTPUT IS THE
02480C: ENTIRE BEHAVIOR TIME-HISTORY.
02490C
02500C: OUTPUT LOAD DATA
02510 18     CALL FORCE(4)
02520C
02530C: OUTPUT FINAL RESULTS
02540 IF(Y(I).LT.YFAIL)PRINT 70,Y(I),T(I)
02550 IF(Y(I).GE.YFAIL)PRINT 71,T(I),V(I)
02577C
02578C CHECK TO SEE IF ENTIRE TIME-HISTORY IS DESIRED
02580 PRINT 72
02590 READ,M
02600         IF(M.EQ.0)GOTO 25
02680 PRINT 76,(T(J),PV(J),A(J),V(J),Y(J),VL(J),J=1,I)
02690 25 PRINT 77
02710C
02740 70 FORMAT(/*NO FAILURE - MAX DEFLECTION OF*,F6.2,
02750+    * IN. REACHED AT*F7.3,* SEC*)
02760 71 FORMAT(/*FAILURE OCCURRED AT*,F7.3,* SEC (FINAL VELOCITY =*,
02770+    F7.2* IN./SEC)*)
02780 72 FORMAT(/*IS TIME HISTORY DESIRED (YES=1, NO=0)*,+)
02830 76 FORMAT(/*  TIME   PRESSURE  ACCELERATION  VELOCITY  *
02840+    *DISPLACEMENT     VL*,/,
02850+    (F6.3,F9.3,F12.1,F12.2,F12.4,F11.0))
02860 77      FORMAT(///,7(*----------*))
02870 80 FORMAT(/*ACCELERATION NOT CONVERGING AT TIME =*,F6.3,
02880+           * SEC (PF =*,F7.3,* PSI)*/*      A(I) SET EQUAL TO*,
02890+           F8.1,* (AVG OF LAST 2 ITERATIONS)*/*      Y(I) =*,
02900+           F8.4,* IN.*)
02950C
02960 999    STOP
02970        END
10000 SUBROUTINE FORCE(IENTRY)
10010C  THIS SUBROUTINE INPUTS THE LOAD PARAMETERS AND DETERMINES
10020C  THE LOAD AT A GIVEN TIME FOR THE FOLLOWING LOAD TYPES:
10030C      1. IDEALIZED BLAST LOAD (FRONT OR SIDE FACE)
10050 COMMON KINC,LDTYPE,KRF,KRAND,TIME,I,Y(100),DT,QU,YU,YFAIL,
10052+  ZLB,HB,BFLG,TFLG,TW,FDYS,ICASE,KPLATE,KCOMP,HP,BP,ES,
10054+  HPC,HS,FPC,FDC,EC,BS,FDYR,AS,D,APS,DP,QAREA,PAREA,ZMASS,
10055+  ZKLM,VL1,VL2,NSLABS,NAMEF(2),KLOAD,ODLSLAB,
10056+  W,P0,C0,LSC,S,ZLEN,CD,PS0,PD0,PR,P,PC,TC,TO,DELAY,
10057+  TT(80,2),PP(80,2),REAC(80,2),,NDEX(2),BR(2)
10058+  NWIN,PHOO,V3,L1,AA(8,2),NN(8),AFRONT,ASIDE,G,G2,G3,G4,PP2,DT
10080C
10130 IF(LDTYPE.EQ.5)GOTO 700
10135 IF(KLOAD.EQ.1)GOTO 500
10140C
10150 GOTO(215,200,300,4),IENTRY
11000C
11010C  CALCULATE LOAD PROPERTIES FOR GIVEN PEAK PRESSURE
11030 200 GOTO(205,210),LOC
11040 205 PSO=(PR-14.0*P3+SQRT(196.0*P3*P3+196.0*P3*PR+PR*PR))/16.0
11050 GOTO 215
11060 210 PSO=PR
11070 215 PDO=2.5*PSO*PSO/(7.0*P3+PSO)
11080 U=C0*SQRT(1.0+(6.0*PSO)/(7.0*P0))
11090 TO=W**0.3333/(2.2399+0.1886*PSO)
11100 GOTO(220,225),LOC
11110 220 TC=3.0*S/U
```

PROGRAM STBEAM (CONTINUED)

```
PC=PSØ*(1-TC/TO)*EXP(-TC/TO)+PDØ*(1-TC/TO)**2*EXP(-2*TC/TO)
CD=1.0
RETURN
225 TA=ZLEN/U
TA2=TA/2.0
TA2TO=TA2/TO
PA=PSØ*(1-TA2TO)*EXP(-TA2TO)+CD*PDØ*(1-TA2TO)**2*EXP(-2*TA2TO)
RETURN
C
C  CALCULATE LØAD
300 GØTØ(305,310),LØC
305 TTO=TIME/TO
IF(TIME.GT.TC)GØTØ 320
P=PC+(TC-TIME)*(PR-PC)/TC
RETURN
310 TTO=(TIME-TA2)/TO
IF(TIME.GT.TA)GØTØ 320
P=PA*TIME/TA
RETURN
320 IF(TTO.GE.1.0)GØTØ 330
P=PSØ*(1-TTO)*EXP(-TTO)+CD*PDØ*(1-TTO)**2*EXP(-2*TTO)
RETURN
330 P=0
RETURN
C
C  PRINT LØAD DATA
4 IF(KINC.EQ.0)GØTØ 400
PRINT 640,LDTYPE
GØTØ 410
400 PRINT 645,LDTYPE
410 CØNTINUE
415 GØTØ(420,425),LØC
420 PRINT 650
GØTØ 430
425 PRINT 655
430 PRINT 660,W,PØ,CØ
IF(KRAND.NE.0)RETURN
GØTØ(435,440),LØC
435 PRINT 665,S,TC,PR
GØTØ 445
440 PRINT 670,ZLEN,TA,PA
445 PRINT 675,U,TO,CD,PSØ,PDØ
RETURN
C
C  LØAD TYPE 5 -- ARBITRARY LØAD SHAPE
700 GØTØ(710,720,730,740),IENTRY
C
C  INPUT LØAD DATA
710 PRINT 780
READ,NPØINT,(TT(J,1),PP(J,1),J=1,NPØINT)
FACTØR=1.0
IF(KINC.EQ.0)GØTØ 718
PMAX=PP(1,1)
DØ 715 J=2,NPØINT
715 IF(PP(J,1).GT.PMAX)PMAX=PP(J,1)
718 PX=PP(2,1)-PP(1,1)
TX=TT(2,1)-TT(1,1)
JJ=1
RETURN
C
C  CALCULATE MAXIMUM LØAD
720 FACTØR=PR/PMAX
GØTØ 718
RETURN
C
C  CALCULATE LØAD
730 IF(TIME.LE.TT(JJ+1,1))GØTØ 735
JJ=JJ+1
PX=PP(JJ+1,1)-PP(JJ,1)
TX=TT(JJ+1,1)-TT(JJ)
IF(TX.EQ.0)TX=1E-10
GØTØ 730
735 P=FACTØR*(PP(JJ,1)+(TIME-TT(JJ,1))*PX/TX)
RETURN
```

PROGRAM STBEAM (CONTINUED)

```
C
C  PRINT LOAD DATA
740 IF(KINC.EQ.1)PRINT 640,LDTYPE
IF(KINC.EQ.0)PRINT 645,LDTYPE
PRINT 790
DØ 745 J=1,NPØINT
P=FACTØR*PP(J,1)
745 PRINT 795,TT(J,1),P
RETURN
C
640 FØRMAT(/*LØAD CAUSING INCIPIENT FAILURE IS AS FØLLØWS:*,
+     /,5X,*LØAD TYPE NUMBER*,I2)
645 FØRMAT(/*PRØPERTIES ØF LØAD ACTING ØN WALL ARE AS FØLLØWS:*,
+     /,5X,*LØAD TYPE NUMBER*,I2)
650 FØRMAT(8X,*(FRØNT FACE)*)
655 FØRMAT(5X,*(SIDE ØR TØP FACE)*)
660 FØRMAT(10X,*W =*,F8.1,* KT   PØ =*,F6.2,* PSI      CØ =*,
+     F7.1,* FPS*)
665 FØRMAT(10X,*S =*,F6.1,* FT      TC =*,F6.3,* SEC      PR =*,
+     F7.3,* PSI*)
670 FØRMAT(10X,*L =*,F6.1,* FT      TA =*,F6.3,* SEC      PA =*,
+     F7.3,* PSI*)
675 FØRMAT(10X,*U =*,F7.1,* FPS   TO =*,F6.3,* SEC      CD =*,
+     F5.1,/,8X,*PSØ =*,F7.3,* PSI  PDØ =*,F7.3,* PSI*)
780 FØRMAT(/*INPUT NUMBER ØF LØAD PØINTS AND THE TIME AND *
+     *PRESSURE AT EACH PØINT*)
790 FØRMAT(/10X,*TIME        PRESSURE*)
795 FØRMAT(F15.3,F12.2)
C
C:  SLAB REACTIØN DATA
500 GØTØ(510,530,540,560),IENTRY
510 RETURN
C
530 STØP
C
540 P=0.0
DØ 555 J=1,NSLABS
545 JJ=INDEX(J)
550 IF(TIME.LE.TT(JJ+1,J))GØTØ 555
INDEX(J)=INDEX(J)+1 $ JJ=INDEX(J)
IF(JJ.LT.NPØINT)GØTØ 552
PRINT 690,TIME
STØP
552 BR(J)=(REAC(JJ+1,J)-REAC(JJ,J))/(TT(JJ+1,J)-TT(JJ,J))
BP=(PP(JJ+1,J)-PP(JJ,J))/(TT(JJ+1,J)-TT(JJ,J))
GØTØ 550
555 P=P+REAC(JJ,J)+(TIME-TT(JJ,J))*BR(J)
P=P+DAREA*(PP(JJ,J)+(TIME-TT(JJ,J))*BP)
RETURN
560 PRINT 680
DØ 565 J=1,NSLABS
565 PRINT 685,NAMEF(J)
680 FØRMAT(/*BEAM LØADED WITH REACTIØNS FRØM FILE(S):*)
685 FØRMAT(10X,A7)
690 FØRMAT(/*END ØF FILE -- BEAM HAS NØT FAILED AT*,F6.3,* SEC*)
RETURN
END
SUBRØUTINE FILL(P3,IENTRY)
C: CØMPUTES AVERAGE AIR PRESSURE IN RØØM DUE TØ BLAST WAVE
C: INCIDENT HEAD-ØN UPØN FRØNT WALL.
C
CØMMØN KINC,LDTYPE,KRF,KRAND,TIME,I,Y(100),DT,QU,YU,YFAIL,
+   ZLB,HB,BFLG,TFLG,TW,FDYS,ICASE,KPLATE,KCØMP,HP,BP,ES,
+   HPC,HS,FPC,FDC,EC,BS,FDYR,AS,D,APS,DP,DAREA,PAREA,ZMASS,
+   ZKLM,VL1,VL2,NSLABS,NAMEF(2),KLØAD,DDLSLAB,
+   W,P3,CØ,LØC,S,ZLEN,CD,PSØ,PDØ,PR,PEXT,PC,TC,TO,DELAY,
+   TT(80,2),PP(80,2),REAC(80,2),INDEX(2),BR(2),
+   NWIN,RHØØ,V3,L1,AA(8,2),NN(8),AFRØNT,ASIDE,G,G2,G3,G4,PP2,DT
LØGICAL L1,L2,L3
C
GØTØ(10,13,11),IENTRY
10 RETURN
C
13 P3Ø=P3
```

```
TT=0.$ T0=0.
RH033=RH30
L2=.FALSE. $ L3=.FALSE.
RETURN
C
11 IF(L1)GOTO 52
IF(L2.A.L3)GOTO 9
52 DDT=(TIME-T0)*0.5
ISTOP=2
53 IF(DDT.LT.DT)GOTO 51
50 DDT=0.5*DDT
ISTOP=2*ISTOP
GO TO 53
51 CONTINUE
DO 99 I=1,ISTOP
TT=T0+I*DDT
IF(TT.GT.TO)GO TO 99
DM=0. $ WW=0. $ NW=0
DO 500 K=1,NWIN
M=NN(K) $ DLY=AA(K,2)+0.000001
IF(DLY.GE.TT)GO TO 500
GOTO(15,16,16),M
15 CDF=1.0
IF(TT-TC)20,20,21
20 P11=(TC-TT)*(PR-PC)/TC+PC
P11=P11+P0
GO TO 30
16 CDF=-0.4
21 R=TT/TO $ RR=1.-R
PD=PD0*RR*RR*EXP(-2.*R)
PS=PS0*RR*EXP(-R)
P11=PS+CDF*PD
P11=P11+P0
30 RH01=RH30*((P11/P3)**G2)
IF(P11-P30)36,36,37
36 JSIGN=-1
L2=.TRUE.
303 P2=P11
RH32=((P2/P30)**G2)*RH333
X=P33/RH333
GO TO 38
37 JSIGN=+1
306 P2=PP2*P11
RH32=((P2/P11)**G2)*RH31
X=P11/RH31
38 U22=G4*(X-P2/RH32)*32.*144.
IF(U22)40,39,39
40 PRINT,*U22 NEGATIVE*,U22
STOP
39 U2=SQRT(U22)*JSIGN
DDM=U2*RH32*AA(K,1)*DDT
DM=DM+DDM
WW=WW+P11*DDM/(G3*RH31)
C
500 CONTINUE
P30=P30+(G-1.)*WW/V3
RH030=RH330+DM/V3
99 CONTINUE
T0=TT
P3=P30-P0
IF(TIME.GE.TC)L3=.TRUE.
RETURN
9 R=TIME*1.0-R
PD=PD0*RR*RR*EXP(-2.0*R)
PS=PS0*RR*EXP(-R)
P3=PS+PD*(AFRONT-0.4*ASIDE)
999 RETURN
END
SUBROUTINE RESIST (IENTRY)
C: THIS SUBROUTINE DETERMINES THE RESISTANCE FUNCTION,
C: TRANSFORMATION FACTORS, AND REACTION COEFFICIENTS: AND
C: SUPPLIES THE REACTION VALUES FOR SPECIFIC DEFLECTIONS
C
COMMON KINC,LDTYPE,KRF,KRAND,TIME,I,Y(100),QT,QU,YU,YFAIL,
```

PROGRAM STBEAM (CONTINUED)

```
30052+   ZLB,HB,BFLG,TFLG,TW,FDYS,ICASE,KPLATE,KCOMP,HP,BP,ES,
30054+   HPC,HS,FPC,FDC,EC,BS,FDYR,AS,D,APS,DP,QAREA,PAREA,ZMASS,
30055+   ZKLM,VL1,VL2,NSLABS,NAMEF(2),KLOAD,ODLFLAG,
30056+   W,PO,CO,LOC,S,ZLEN,CD,PSO,POO,PR,PEXT,PC,TC,TO,DELAY,
30057+    U(80,2),PP(80,2),REAC(80,2),INDEX(2),BR(2),
30058+          RHOO,V3,L1,AA(8,2),NN(8),AFRONT,ASIDE,G,G2,G3,G4,PP2,DT
30100 REAL      B,ISTEEL,ICC,ICS,IAVG,KK1,KK2,KS1,KS2
30120 GOTO(4,500,45),IENTRY
30130C
30140C * ENTRY 1:  INPUT BEAM DATA *
30150 4 PRINT 615
30160 READ,ZLB,HB,BFLG,TFLG,TW,FDYS,ICASE,KPLATE,KCOMP
30165 ZLEN=ZLB/12.0
30170 ICASE4=ICASE-4
30180 ES=29E6
30185 BP=0 $ HP=0
30190 IF(KPLATE.EQ.0)GOTO 11
30200 PRINT 616
30210 READ,BP,HP
30220 11 IF(KCOMP.EQ.0)GOTO 13
30230 PRINT 618
30240 READ,HBP,HS,FPC
30250 FDC=1.25*FPC
30260 EC=57619.0*SQRT(FPC) $ ECKIP=EC/1000.0
30270 N=ES/EC
30280C * EFFECTIVE WIDTH OF CONCRETE *
30290 BS=ZLB/4.0
30300 IF(BS.GT.BFLG+16.0*HS)BS=BFLG+16.0*HS
30310 IF(ICASE.EQ.5)GOTO 13
30320 PRINT 619
30330 READ,FDYR,AS,D,APS,DP
30340 13 CONTINUE
30350 QAREA=ZLB*BFLG
30360 PAREA=PAREA+QAREA
30370C
30380C * OUTPUT BEAM DATA *
30390 PRINT 620,ICASE,ZLB,HB,BFLG,TFLG,TW,FDYS
30400 IF(KPLATE.NE.0)PRINT 625,BP,HP
30410 IF(KCOMP.EQ.0)GOTO 15
30420 PRINT 630,HBP,HS,BS,FPC,FDC,ECKIP
30430 IF(ICASE.GT.5)PRINT 635,AS,D,FDYR,APS,DP
30440 AS=AS*BS/12.0 $ APS=APS*BS/12.0
30450 15 CONTINUE
30460C
30470C * DETERMINE BEAM PROPERTIES INDEPENDENT OF FDY *
30500 CALL TRANS(ICASE4,ZKLM,ZKLMSE,ZKLMFE,ZKLMP,VL1S,VL2S,
30510+             VL1F,VL2F,VL1P,VL2P)
30530C *  PROPERTIES FOR STEEL SECTION *
30540 AFLG=BFLG*TFLG
30550 AB=2.0*AFLG+TW*(HB-2.0*TFLG)
30560 AP=HP*BP
30570 ASTEEL=AB+AP
30575 ZMASS=490.0*ASTEEL*ZLB/1728.0
30580 IB=2.0*(BFLG*TFLG**3/12.0+AFLG*(0.5*(HB-TFLG))**2)
30590+    +TW*(HB-2.0*TFLG)**3/12.0
30600 YS=0.5*(HB+HP)*AP/ASTEEL
30610 YBS=0.5*HB+HP-YS
30620 YTS=0.5*HB+YS
30630 H1=HBP+YTS
30640 ISTEEL=IB+AB*YS*YS+BP*HP**3/12.0+AP*(YBS-0.5*HP)**2
30650 ZS=AFLG*(HB-TFLG)+TW*((0.5*HB-TFLG)**2-YS*YS)+AP*(YBS-0.5*HP)
30670 IF(KCOMP.EQ.1)RETURN
30680 ZCC=ZS $ ZCS=ZS
30690 ICC=ISTEEL $ ICS=ISTEEL
30700 RETURN
30710C
30750C * ENTRY 3: DETERMINE BEAM PROPERTIES DEPENDENT ON FDY *
30760 45 IF(KCOMP.NE.1)GOTO 78
30770C *  PROPERTIES FOR COMPOSITE SECTION *
30780 ASR=AS*FDYR/FDYS $ APSR=APS*FDYR/FDYS
30790C
30800C *  NEGATIVE MOMENT SECTION *
30810 IF(ICASE.EQ.5)GOTO 50
30820 YC=(ASR*(HS-D)+APSR*(HS-DP)+ASTEEL*H1)/(ASR+APSR+ASTEEL)
```

```
30830 ICS=ASR*(YC-HS+D)**2+APSR*(YC-HS+DP)**2+ISTEEL
30840+      +ASTEEL*(HBP+YTS-YC)**2
30850 ASTELP=0.5*(ASTEEL-ASR-APSR)
30860 ZTC=TFLG+(ASTELP-AFLG)/TW
30870 E=YTS+HBP-HS+D
30880 EP=YTS+HBP-HS+DP
30890 EPP=YTS-0.5*TFLG
30900 EPPP=EPP-0.5*ZTC
30910 ZCS=ASR*E+APSR*EP+2.0*AFLG*EPP+2.0*(ASTELP-AFLG)*EPPP
30930C
30940C *  PØSITIVE MØMENT SECTIØN *
30950C *    ELASTIC MØMENT ØF INERTIA *
30960 50 TERM1=N*ASTEEL/BS
30970 YC=-TERM1+SQRT(TERM1*TERM1+2.0*TERM1*H1)
30980 IF(YC.GT.HS)GØTØ 55
30990C *     NEUTRAL AXIS IN SLAB *
31000 ICC=ISTEEL+ASTEEL*(H1-YC)**2+BS*YC**3/(3.0*N)
31010 GØTØ 60
31020C *     NEUTRAL AXIS BELØW SLAB *
31030 55 YC=(BS*HS*HS+2.0*N*ASTEEL*H1)/(2.0*(BS*HS+N*ASTEEL))
31040 ICC=ISTEEL+ASTEEL*(H1-YC)**2+BS*HS**3/(12.0*N)
31050+     +BS*HS*(YC-0.5*HS)**2/N
31060C *    PLASTIC SECTIØN MØDULUS *
31070 60 BSU=0.85*FDC*BS/FDYS
31090 ZC=ASTEEL/BSU
31100 IF(ZC.GT.HS)GØTØ 65
31110 IF(ZC.GT.HBP)GØTØ 62
31120C *     NEUTRAL AXIS IN SLAB (ABØVE BEAM FLANGE) *
31130 ZCC=ASTEEL*(YTS+HBP-0.5*ZC)
31140 GØTØ 75
31150C *     NEUTRAL AXIS BELØW BEAM FLANGE (ENCASED BEAM) *
31160 62 ACU=BSU*HBP
31170 E=YTS+0.5*HBP
31180 GØTØ 68
31190C *     NEUTRAL AXIS BELØW SLAB *
31200 65 ACU=BSU*HS
31210 E=YTS+HBP-0.5*HS
31220 68 ASTELP=0.5*(ASTEEL-ACU)
31230 IF(ASTELP.GT.AFLG)GØTØ 70
31240C *      NEUTRAL AXIS IN BEAM FLANGE *
31250 ZC=HBP+ASTELP/BFLG
31260 ZCC=ACU*E+2.0*ASTELP*(YTS-0.5*(ZC-HBP))
31270 GØTØ 75
31280C *      NEUTRAL AXIS IN BE : WEB *
31290 70 ZC=(ASTELP-AFLG)/TW+HB -TFLG
31300 EP=YTS-0.5*TFLG
31310 EPP=EP-0.5*(ZC-HBP)
31320 ZCC=ACU*E+2.0*AFLG*EP+. .0*( '' ,ELP-AFLG)*EPP
31330 75 CØNTINUE
31350C
31360C *  DETERMINE RESISTANCE ' '' S (IØTAL) FØR BEAM *
31370C *  (Q IS IN UNITS ØF LB, K. IN LB/IN. AND Y IN INCHES ) *
31375 78 CALL CØEF(ICASE4,ASS,PSS,AF,BF,ICC,ICS)
31380 IF(ICASE.GT.5)GØTØ 80
31390C
31400C CASE 5
31410 QU=ZCC*FDYS/(BSS*ZLB)
31420 KK1=ES*ICC/(ASS*ZLB**3)
31430 YU=QU/KK1
31440 QFAIL=ZS*FDYS/(BSS*ZLB)
31450 YFAIL=26.4*QFAIL/(ES*ISTEEL/(ASS*ZLB**3))
31460 GØTØ 100
31470C
31480C CASES 6 AND 7
31490 80 Q1=ZCS*FDYS/(BF*ZLB)
31500 IAVG=0.5*(ICC+ICS)
31510 KK1=ES*ICS/(AF*ZLB**3)
31520 Y1=Q1/KK1
31530 IF(ICASE.EQ.7)GØTØ 85
31540 QU=FDYS*(ZCC+ZCS)/(BSS*ZLB)
31550 QFAIL=2.0*FDYS*ZS/(BSS*ZLB)
31555 QS1=FDYS*ZS*12.0/ZLB
31556 KS1=ES*ISTEEL*384.0/ZLB**3
31560 GØTØ 90
```

PROGRAM STBEAM (CONTINUED)

```
85 QU=FDYS*(ZCC+0.5*ZCS)/(BSS*ZLB)
QFAIL=1.5*FDYS*ZS/(BSS*ZLB)
QS1=FDYS*8.0/ZLB
KS1=ES*ISTEEL*185.0/ZLB**3
90 KK2=ES*IAVG/(ASS*ZLB**3)
YU=Y1+(QU-Q1)/KK2
YS1=QS1/KS1
KS2=ES*ISTEEL/(ASS*ZLB**3)
YSU=YS1+(QFAIL-QS1)/KS2
YFAIL=26.4*(YS1+YSU*(1.0-QS1/QFAIL))
C
C * ADJUST RESISTANCE CURVE FOR SLAB DEAD LOAD *
100 QDL=QDLSLAB+490.0*ASTEEL*ZLB/1728.0
YDL=QDL/KK1
IF(KRAND.NE.1)PRINT 633,QDL,YDL
Y1=Y1-YDL$ YU=YU-YDL$ YFAIL=YFAIL-YDL
Q1=Q1-QDL$ QU=QU-QDL$ QFAIL=QFAIL-QDL
C
C * OUTPUT RESISTANCE CURVE *
IF(KRAND.EQ.1)GOTO 335
PRINT 650
IF(ICASE.EQ.5)GOTO 320
PRINT 660,Q1,Y1
320 PRINT 660,QU,YU,QFAIL,YFAIL
335 CONTINUE
RETURN
C
C * ENTRY 2: DETERMINE THE RESISTANCE (TOTAL) OF THE BEAM
C              AS A FUNCTION OF Y(I) *
C
500 IF(Y(I).GT.YFAIL)GOTO 560
IF(Y(I).GT.YU)GOTO 540
IF(ICASE.GT.5)GOTO 520
C
C * ELASTIC RANGE - CASE 5 *
ZKLM=ZKLMSE
VL1=VL1S $ VL2=VL2S
QT=Y(I)*KK1
RETURN
C
520 IF(Y(I).GT.Y1)GOTO 530
C
C * ELASTIC RANGE - CASES 6 & 7 *
ZKLM=ZKLMFE
VL1=VL1F $ VL2=VL2F
QT=Y(I)*KK1
RETURN
C
C * ELASTØ-PLASTIC RANGE - CASES 6 & 7 *
530 ZKLM=ZKLMSE
VL1=VL1S $ VL2=VL2S
QT=Q1+KK2*(Y(I)-Y1)
RETURN
C
C * PLASTIC RANGE - ALL CASES *
540 ZKLM=ZKLMP
VL1=VL1P $ VL2=VL2P
QT=QU+(Y(I)-YU)*(QFAIL-QU)/(YFAIL-YU)
RETURN
C
C * BEAM CØLLAPSED (SET RESISTANCE TO SMALL VALUES) *
560 QT=1E-10
RETURN
C
615 FØRMAT(/*INPUT LB,HB,BF,TF,TW,FDYS,ICASE,KPLATE,KCOMP*)
616 FØRMAT(/*INPUT BP & HP (FOR BOTTOM COVER PLATE)*,+)
618 FØRMAT(/*INPUT HBP,HS,F'C (FOR COMPOSITE BEAM)*,+)
619 FØRMAT(/*INPUT FDYR & AS,D,APS,DP AT BEAM SUPPORT*,+)
620 FØRMAT(//*PRØPERTIES ØF STEEL SUPPØRT BEAM -- SUPPØRT TYPE NO.*,
+     I2,/,*  LB =*,F6.1,* IN.*,5X,*HB =*,F6.2,* IN.*,6X,*BF =*,
      F7.3,* IN.*,/,*  TF =*,F6.3,* IN.*,5X,*TW =*,F6.3,* IN.*,4X,
+     *FDYS =*,F8.1,* PSI*)
625 FØRMAT(/*  BP =*,F6.2,* IN.*,5X,*HP =*,F6.3,* IN.*)
630 FØRMAT(/* HBP =*,F6.2,* IN.*,5X,*HS =*,F6.2,* IN.*,4X,
```

PROGRAM STBEAM (CONTINUED)

```
32340+     *  BS =*,F6.2,* IN.*,/,* F'C =*,F7.1,* PSI   FDC =*,
32350+     F7.1,* PSI*,5X,*EC =*,F7.1,* KSI*)
32360 635 FORMAT(/*  AS =*,F7.4,* SQ IN./FT*,6X,*D =*,F7.3,* IN.*,
32370+     4X,*FDYR =*,F8.1,* PSI*,/,* A'S =*,F7.4,* SQ IN./FT*,5X,
32380+     *D' =*,F7.3,* IN.*)
32390 633 FORMAT(/*QDL =*,F10.2,* LB    YDL =*,F8.4,* IN.*)
32400 650 FORMAT(//*LOAD-DEFLECTION CURVE*,/,5X,*QT (LB)      Y (IN.)*)
32410 660 FORMAT(F12.2,F12.4)
32420 END
40000 SUBROUTINE COEF (ICASE4,ASS,BSS,AF,BF,ICEN,ISUP)
40010C
40020C  THIS SUBROUTINE DETERMINES DEFLECTION AND MOMENT COEFFICIENTS
40030C  FOR TEE BEAMS WITH VARIABLE MOMENT OF INERTIA
40040C
40050 REAL ICEN,ISUP
40060C:  CASE 5. ONE-WAY SIMPLY SUPPORTED WALL
40070      ASS=5.0/384.0
40080       BSS=0.125
40090 RI=ISUP/ICEN
40100      GOTO(270,60,70),ICASE4
40110C
40120C:  CASE 6. ONE-WAY FIXED END WALL
40130 60 BF=(.00953+.03213*RI)/(.211+.289*RI)
40140 AF=.00132+.01169*RI-BF*(.0223+.1025*RI)
40150 RETURN
40160C
40170C:  CASE 7. ONE-WAY PROPPED CANTILEVER WALL
40180 70 BF=(.0109+.0305*RI)/(.1925+.1407*RI)
40190 AF=-.00439-.00342*RI+BF*(.08333+.02083*RI)
40200 270  RETURN
40210      END
50000 SUBROUTINE TRANS (ICASE4,ZKLM,ZKLMSE,ZKLMFE,ZKLMP,VL1S,VL2S,
50010+     VL1F,VL2F,VL1P,VL2P)
50020C
50030C  THIS SUBROUTINE DETERMINES TRANSFORMATION FACTORS AND
50040C  DYNAMIC REACTION COEFFICIENTS FOR ONE-WAY BEAMS
50050C
50060C  ALL CASES
50070 ZKLMSE=0.78
50080 ZKLMP=0.66
50090 VL1S=0.107
50100 VL2S=0.393
50110 VL1P=0.125
50120 VL2P=0.375
50130 GOTO(300,310,320),ICASE4
50140C
50150C  CASE 5
50160 300 ZKLM=ZKLMSE
50170 VL1=VL1S
50180 VL2=VL2S
50190 RETURN
50200C
50210C  CASE 6
50220 310 ZKLMFE=0.77
50230 VL1F=0.136
50240 VL2F=0.364
50250 GOTO 330
50260C
50270C  CASE 7
50280 320 ZKLMFE=0.78
50290 VL1F=0.165
50300 VL2F=0.459
50310 330 ZKLM=ZKLMFE
50320 VL1=VL1F
50330 VL2=VL2F
50340 RETURN
50350 END
70000       SUBROUTINE RANDOM (IENTRY)
70010C THIS SUBROUTINE INPUTS MEAN AND STANDARD DEVIATIONS FOR RANDOM
70020C VARIABLES; GENERATES RANDOM VALUES; AND CONTROLS REQUIRED
70030C NUMBER OF CASES TO BE RUN; AND OUTPUTS FINAL RESULTS AND SUMMARY
70040C
70050 COMMON KINC,LDTYPE,KRF,KRAND,TIME,I,Y(100),QT,QU,YU,YFAIL,
70052+  ZLB,HB,BFLG,TFLG,TW,FDYS,ICASE,KPLATE,KCOMP,HP,BP,ES,
```

PROGRAM STBEAM (CONTINUED)

```
+     HPC,HS,FPC,FDC,SC,PS,FDYR,AS,D,APS,DP,DAREA,PAREA,ZMASS,
+     ZKL4,VL1,VL2,NSLABS,NAMEF(2),KLOAD,DOLSLAB,
+     W,PJ,CO,LOC,S,ZLEN,CD,PSO,PDO,PR,PEXT,PC,TC,TO,DELAY,
+     TT(80,2),PP(80,2),REAC(80,2),INDEX(2),BR(2),
+     NWIN,RHO,V3,L1,AA(8,2),NN(8),AFRONT,ASIDE,G,G2,G3,G4,PP2,DT
        COMMON /RAND/ TIMEC
        DIMENSION CHI25(7),CHI975(7),TDIST(7)
C
C VALUES FOR 97.5% (F=19,24,29,34,39,44,49)
        DATA CHI25/.4688,.5167,.5533,.5825,.6065,.6267,.6440/
        DATA CHI975/1.7295,1.6402,1.5766,1.5284,1.4903,1.4591,1.4331/
        DATA TDIST/2.093,2.064,2.045,2.032,2.022,2.016,2.010/
C
        GOTO(5,50,70),IENTRY
      5 XDUMMY=XNORM1(-1.0,0.0,1.0)
C INITIALIZE RANDOM NUMBER GENERATOR
        PRINT,/,*INPUT NRAND*,
        READ,NRAND
        DO 47 I=1,NRAND
        XDUMMY=XNORM1(0.0,0.0,1.0)
     47 CONTINUE
        INDEX=0$  SPSO=0$  SSPSO=0
        ICHECK=20
C
C  INPUT MEAN AND STANDARD DEVIATION FOR RANDOM VARIABLES
        IF(LOC.EQ.2)GOTO 30
        PRINT 87
        READ,SMEAN,SSD
C  STEEL (NON-COMPOSITE AND COMPOSITE) SUPPORT BEAMS
   30   PRINT 86
        READ,FDYSMEN,FDYSSD
        IF(KCOMP.EQ.1.AND.ICASE.GT.5)GOTO 40
        IF(LOC.EQ.1)PRINT 96
        IF(LOC.NE.1)PRINT 95
        RETURN
   40   PRINT 88
        READ,FDYRMEN,FDYRSD
        IF(LOC.EQ.1)PRINT 97
        IF(LOC.NE.1)PRINT 98
        RETURN
C
C  GENERATE RANDOM VALUES
   50   FDYS=XNORM1(0.0,FDYSMEN,FDYSSD)
        IF(FDYS.LE.0)GOTO 50
        IF(KCOMP.NE.1.OR.ICASE.EQ.5)GOTO 58
   55   FDYR=XNORM1(0.0,FDYRMEN,FDYRSD)
        IF(FDYR.LE.0.AND.FDYRMEN.NE.0)GOTO 55
   58   CONTINUE
        IF(LOC.EQ.2.OR.SMEAN.EQ.0)GOTO 65
   60   S=XNORM1(0.0,SMEAN,SSD)
        IF(S.LE.0)GOTO 60
   65   INDEX=INDEX+1
        RETURN
C SUM VALUES OF PSO AND PSO**2 FOR USE IN STATISTICAL ANALYSIS
   70   SPSO=SPSO+PSO
        SSPSO=SSPSO+PSO*PSO
C
C OUTPUT FINAL RESULTS
   76   IF(KCOMP.EQ.1.AND.ICASE.GT.5)GOTO 78
        IF(LOC.EQ.1)PRINT 92,FDYS,S,PSO,TIMEC
        IF(LOC.NE.1)PRINT 90,FDYS,PSO,TIMEC
        GOTO 80
   78   IF(LOC.EQ.1)PRINT 93,FDYS,FDYR,S,PSO,TIMEC
        IF(LOC.NE.1)PRINT 94,FDYS,FDYR,PSO,TIMEC
   80   IF(INDEX.LT.ICHECK)RETURN
C
C DETERMINE MEAN, STANDARD DEVIATION, AND STANDARD ERROR FOR PSO
        ZNO=INDEX
        ZMEAN=SPSO/ZNO
        SD=SQRT((SSPSO-ZNO*ZMEAN*ZMEAN)/ZNO)
        STDERR=SD/(SQRT(ZNO-1))
C CHECK IF MAXIMUM OF 50 PSO SAMPLES OBTAINED
        IF(INDEX.EQ.50)GOTO 62
C CHECK IF 95% CONFIDENCE INTERVAL FOR MEAN PSO VALUE IS
```

PROGRAM STBEAM (CONCLUDED)

```
      IF(STDERR*TDIST((INDEX-15)/5)/ZMEAN.GT.0.10)GOTO 61
C
C CONFIDENCE INTERVAL IS WITHIN 10% -- DETERMINE UPPER LIMIT OF
C 95% CONFIDENCE INTERVAL FOR STANDARD DEVIATION
C PROBABILITY VALUE AND ITS 95% CONFIDENCE INTERVAL UPPER LIMIT
   62 SDU=SD/(SQRT(CHI25((INDEX-15)/5)))
C CHECK IF MAXIMUM OF 50 PSI SAMPLES OBTAINED
      IF(INDEX.EQ.50)GOTO 53
C CHECK IF UPPER VALUE OF 95% CONFIDENCE INTERVAL FOR STANDARD
C DEVIATION IS WITHIN 0.10*MEAN OF THE STANDARD DEVIATION
      IF(((SDU-SD)/ZMEAN).GT.0.10)GOTO 61
C
C 95% CONFIDENCE INTERVAL IS WITHIN 10% FOR BOTH MEAN AND 90%
C PROBABILITY VALUE -- THEREFORE SUFFICIENT SAMPLES OBTAINED
C DETERMINE 95% CONFIDENCE INTERVALS FOR MEAN, STANDARD DEVIATION
C
C AND 10% AND 90% PROBABILITY VALUES
   53 ZMEANL=ZMEAN-STDERR*TDIST((INDEX-15)/5)
      ZMEANU=ZMEAN+STDERR*TDIST((INDEX-15)/5)
      SDL=SD/(SQRT(CHI975((INDEX-15)/5)))
      P10=ZMEAN-1.282*SD
      P10L=ZMEAN-1.282*SDU
      P10U=ZMEAN-1.282*SDL
      P90=ZMEAN+1.282*SD
      P90L=ZMEAN+1.282*SDL
      P90=ZMEAN+1.282*SD
      P90U-ZMEAN+1.282*SDU
      P90U=ZMEAN+1.282*SDU
C
C OUTPUT STATISTICAL PARAMETERS OF INCIPIENT COLLAPSE PRESSURE
      PRINT 100,ZMEAN,ZMEANL,ZMEANU,SD,SDL,SDU,P10,P10L,P10U,
     *  P90,P90L,P90U
      PRINT 105,INDEX,STDERR
      GOTO 999
C
C 95% CONFIDENCE INTERVAL IS NOT WITHIN 10% FOR BOTH MEAN AND 90%
C
C VALUES -- THEREFORE OBTAIN 5 ADDITIONAL SAMPLES
   61 ICHECK=ICHECK+5
      RETURN
C
   86 FORMAT(/*INPUT MEAN AND STANDARD DEVIATION FOR FOYS*,F)
   87 FORMAT(/*INPUT MEAN AND STANDARD DEVIATION FOR S*,F)
   88 FORMAT(/*INPUT MEAN AND STANDARD DEVIATION FOR FOYR*,F)
   90 FORMAT(F9.2,F10.2,F14.3)
   92 FORMAT(F9.1,F11.2,F10.2,F14.3)
   93 FORMAT(2F9.1,F11.2,F10.2,F14.3)
   94 FORMAT(2F9.1,F10.2,F14.3)
   95 FORMAT(///,4X,*FOYS*,7X,*PSI*,6X,*COLLAPSE TIME*)
   96 FORMAT(///,4X,*FOYS*,7X,*S*,8X,*PSI*,6X,*COLLAPSE TIME*)
   97 FORMAT(///,4X,*FOYS*,7X,*FOYR*,8X,*S*,8X,*PSI*,6X,
     *  *COLLAPSE TIME*)
   98 FORMAT(///,4X,*FOYS*,7X,*FOYR*,7X,*PSI*,6X,*COLLAPSE TIME*)
  100 FORMAT(///,11X,*STATISTICAL PROPERTIES OF INCIPIENT PSI*,
     *   //,39X,*95% CONFIDENCE LIMITS*,/,7X,*ITEM*,18X,
     *   *VALUE        LOWER        UPPER*,//,* MEAN*,F23.2,
     *   2F12.2,//,* STANDARD DEVIATION*,F15.2,2F12.2,//,
     *   * 10% PROBABILITY VALUE*,3F12.2,//,
     *   * 90% PROBABILITY VALUE*,3F12.2)
  105 FORMAT(//,5X,*NUMBER OF OBSERVATIONS =*,I3,/,5X,
     *   *STANDARD ERROR =*,F5.2)
C
  999 STOP$ END
      FUNCTION XNORM(X,A,B)
      IF(X)10,20,20
   10 X0=RANF(-1.0)
   20 X1=RANF(0.0)
      X2=RANF(0.0)
      Z=SQRT(-2.0*ALOG(X1))*(COS(6.283184*X2))
      XNORM=A+Z*B
      RETURN
      END
```

FLAT

Flat Slab or Flat Plate

PROGRAM FLAT

```
PROGRAM FLAT(INPUT,OUTPUT)
C:  THIS ROUTINE IS THE CONTROLLING ROUTINE FOR THE PROGRAM
C:  USED IN THE ANALYSIS OF REINFORCED CONCRETE FLAT SLABS
C
COMMON KINC,LDTYPE,KRF,KRAND,TIME,I,Y(100),Q,QU,YU,YFAIL,
+  ZLS,HS,FDY,AREA,ZMASS,ZKLM,VS1,VS2,PS0,PD0,PF,PEXT,PC,TC,TO,
+  P0,DELAY,S
DIMENSION A(60),V(80),T(80),VS(80),QQ(80),PN(80)
COMMON /SHEAR/ ISHEAR,JSHEAR,V. .EAR,MEMB
COMMON /RAND/ TIMEC
C
C * READ TITLE AND CONTROL PARAMETERS *
5 PRINT 67
READ 68,TITLE
PRINT 85
READ,KINC,LDTYPE,KRF,KRAND
DELAY=0
CALL RESIST(1)
CALL FORCE(1)
IF(KRF.EQ.0)GOTO 14
CALL FILL(PINT,1)
14    IF(KRAND.NE.1)GOTO 35
      CALL FORCE(4)
      CALL RANDOM(1)
34    CALL RANDOM(2)
35    CALL RESIST(3)
C
C: MINIMUM, MAXIMUM, AND STARTING VALUES ARE DETERMINED FOR CASES
C: WHERE THE LOAD CAUSING INCIPIENT COLLAPSE IS TO BE FOUND
13    IF(KINC.EQ.0)GOTO 23
       PF=QU
       PFMAX=0
       PFMIN=PF/2.0
      GOTO 20
16     PF=(PFMIN+PFMAX)/2.0
20    CALL FORCE(2)
23    IF(KRF.EQ.0)GOTO 24
      CALL FILL(PINT,2)
C
C:  INITIALIZE VALUES FOR BETA METHOD (BETA=1/6) AND COMPUTE VALUES
C:  FOR FIRST TIME INTERVAL ASSUMING ELEMENT INITIALLY AT REST
24     I=1
       TIME=0
      V(1)=0 $ Y(1)=0
       DELTA=0.001
JSHEAR=0
      IF(KRF.NE.1)GOTO 30
27    IF(TIME.GE.(DELAY-.00001))GOTO 30
      TIME=TIME+DELTA
      CALL FILL(PINT,3)
      GOTO 27
30    CALL RESIST(2)
      A(1)=0.0 $ VS(1)=0.0
      T(1)=TIME
C
C   PROCEDURE FOR ALL SUBSEQUENT TIME INTERVALS
       I=I+1
      IF(I.LT.81)GOTO 11
      PRINT 98,TIME
98 FORMAT(/*I=81:  TIME =*,F6.3,*,  FAILURE ASSUMED TO NOT OCCUR*)
      GOTO 6
11    TIME=TIME+DELTA
       T(I)=TIME
       A(I)=A(I-1)
      IF(KRF.NE.0)GOTO 10
      CALL FORCE(3)
      PN(I)=PEXT
      GOTO 2
10    CALL FILL(PINT,3)
      PN(I)=PINT
2     CONTINUE
      DO 8 JJ=1,10
       Y(I)=Y(I-1)+DELTA*V(I-1)+DELTA*DELTA*(A(I-1)/3.+A(I)/6.)
       CALL RESIST(2)
```

```
01960 4    ANEW=AREA*(PN(I)-Q)/(ZMASS*ZKLM)
01970       ADELTA=ANEW-A(I)
01980       A(I)=ANEW
01985 IF(ANEW.EQ.0)PRINT,*1985*,TIME,PN(I),Q,ZMASS,ZKLM,Y(I),A(I-1)
01990       IF(ABS(ADELTA/(ANEW+1E-10)).LT.0.01)GØTØ 9
02000 8     CØNTINUE
02010       A(I)=ANEW-ADELTA/2.0
02020 PRINT 80,TIME,PF,A(I),Y(I)
02030 9     CØNTINUE
02040       Y(I)=Y(I-1)+DELTA*V(I-1)+DELTA*DELTA*(A(I-1)/3.+A(I)/6.)
02050       V(I)=V(I-1)+DELTA*(A(I)+A(I-1))/2.0
02060      VS(I)=AREA*(VS1*PN(I)+VS2*Q)
02070      QQ(I)=Q
02080 IF(ISHEAR.EQ.1.AND.VS(I).GT.VSHEAR)JSHEAR=1
02085 IF(JSHEAR.EQ.1.AND.MEMB.EQ.0)GØTØ 7
02090C
02100C: CHECK FØR MAXIMUM DEFLECTIØN ØR FAILURE
02110C:    IF MAXIMUM DEFLECTIØN REACHED, WALL DID NØT FAIL
02120       IF(Y(I).LE.Y(I-1).AND.PN(I).LE.PN(I-1))GØTØ 6
02130       IF(Y(I).LT.0)GØTØ 6
02135       IF(Y(I).GE.YFAIL)GØTØ 7
02140       IF(TIME-DELAY.GE.0.010)DELTA=0.002
02160       IF(TIME-DELAY.GE.0.020)DELTA=0.005
02170       IF(TIME-DELAY.GE.0.100)DELTA=0.010
02180       IF(TIME-DELAY.GE.0.500)DELTA=0.050
02190C:    IF FAILURE DEFLECTIØN REACHED, ELEMENT FAILED
02210       GØTØ 1
02220C
02230C: INTERVAL HALVING PRØCEDURE TØ DETERMINE LØAD CAUSING INCIPIENT
02240C: CØLLAPSE FØR CASES WHERE DESIRED
02250C:    ELEMENT DID NØT FAIL -- SET PFMIN TØ PF
02260 6     CØNTINUE
02280       IF(KINC.EQ.0)GØTØ 18
02290 36    PFMIN=PF
02300 IF(PFMAX.GT.0)GØTØ 16
02310       PF=2.0*PF
02320       GØTØ 20
02330C:    ELEMENT FAILED -- SET PFMAX TØ PF
02340 7     CØNTINUE
02350       TIMEC=TIME
02370       IF(KINC.EQ.0)GØTØ 18
02380 37    PFMAX=PF
02390C:    CHECK TØ SEE IF LØAD RANGE IS WITHIN DESIRED ACCURACY
02400 17     IF((PFMAX-PFMIN)/PFMIN.GT.0.01)GØTØ 16
02410       IF(KRAND.NE.1)GØTØ 15
02420       CALL RANDØM(3)
02430       GØTØ 34
02440C
02450C: ØUTPUT DATA INCLUDES THE MAXIMUM DEFLECTIØN AND TIME ØF
02460C: ØCCURANCE FØR A NØN-FAILING ELEMENT ØR THE TIME AND VELØCITY
02470C: AT CØLLAPSE FØR A FAILING ELEMENT.  ØPTIØNAL ØUTPUT IS THE
02480C: ENTIRE BEHAVIØR TIME-HISTØRY.
02490C
02500C: ØUTPUT LØAD DATA
02510 18    CALL FØRCE(4)
02520C
02530C: ØUTPUT FINAL RESULTS
02535 IF(JSHEAR.EQ.1.AND.MEMB.EQ.0)GØTØ 40
02540 IF(Y(I).LT.YFAIL)PRINT 70,Y(I),T(I)
02550 IF(Y(I).GE.YFAIL)PRINT 71,T(I),V(I)
02560 IF(JSHEAR.EQ.1)PRINT 96
02570 GØTØ 42
02577C
02578C CHECK TØ SEE IF ENTIRE TIME-HISTØRY IS DESIRED
02579 40 PRINT 97,T(I),V(I)
02580 42 PRINT 72
02590 READ,M
02600        IF(M.EQ.0)GØTØ 25
02680 PRINT 76,(T(J),PN(J),A(J),V(J),Y(J),QQ(J),VS(J),J=1,I)
02690 25 PRINT 77
02700 GØTØ 5
02710C
02720 67 FØRMAT(/*INPUT TITLE*,+)
02730 68 FØRMAT(A59)
```

PROGRAM FLAT (CONTINUED)

```
70 FORMAT(/*NO FAILURE - MAX DEFLECTION OF*,F6.2,
+    ' IN. REACHED AT*F7.3,* SEC*)
71 F  .TAI(/*FAILURE OCCURRED AT*,F7.3,* SEC (FINAL VELOCITY =*,
+    - ,.* IN./SEC)*)
72 FO   7 (* S TIME HISTORY DESIRED (YES=1, NO=0)*,*)
76 F           TIME  PRESSURE  ACCELERATION  VELOCITY  *
+    *.    . F RNT    QU         VS*,/,
+    (F .3, .3,F12.1,F12.2,F12.4,F10.2,F9.0))
77     FORMAT(///,7(*-----------*))
80 FORMAT(/*ACCELERATION NOT CONVERGING AT TIME =*,F6.3,
+          * SEC (PF =*,F7.3,* PSI)*/*      A(I) SET EQUAL TO*,
+          F8.1,* (AVG OF LAST 2 ITERATIONS)*/*      Y(I) =*,
+          F8.4,* IN.*)
85 FORMAT(/*INPUT KINC,LDTYPE,KRF,KRAND(1=RANDOM)*,*)
90 FORMAT(/*ARE REACTIONS TO BE OUTPUT TO FILE (0=NO,1=YES)*,*)
95 FORMAT(/*INPUT NAME OF SLAB REACTION DATA FILE*,*)
96 FORMAT(/*SHEAR FAILURE -- TENSILE MEMBRANE RESISTANCE*,
+    * CONTINUED*)
97 FORMAT(/*SHEAR FAILURE AT*,F7.3,* SEC (FINAL VELOCITY =*,
+    F7.2* IN./SEC)*)
C
999   STOP
      END
SUBROUTINE FORCE(IENTRY)
C  THIS SUBROUTINE INPUTS THE LOAD PARAMETERS AND DETERMINES
C  THE LOAD AT A GIVEN TIME FOR THE FOLLOWING LOAD TYPES:
C      1. IDEALIZED BLAST LOAD (FRONT OR SIDE FACE)
COMMON KINC,LDTYPE,KRF,KRAND,TIME,I,Y(100),Q,QU,YU,YFAIL,
+  ZLS,HS,FDY,AREA,ZMASS,ZKLM,VS1,VS2,PSO,PDO,PR,P,PC,TC,TO,
+  PO,DELAY,S
DIMENSION TT(20),PP(20)
C
IF(LDTYPE.EQ.5)GOTO 500
C
GOTO(100,200,300,4),IENTRY
C
C * INPUT LOAD PARAMETERS *
C: LOCATION 1. FRONT FACE LOADING (USED IN ROOM-FILLING PROCEDURE)
100 W=1000.0 $ PO=14.7 $ CO=1120.0
IF(KRF.NE.1)GOTO 102
LOC=1
IF(KRAND.EQ.1)RETURN
PRINT 600
READ,S
GOTO 105
C: LOCATION 2. TOP FACE LOADING
102 CD=0 $ LOC=2
ZLEN=ZLS/12.0
105 IF(KINC.EQ.1)RETURN
PRINT 610
READ,PSO
PR=2.0*PSO*(7.0*PO+4.0*PSO)/(7.0*PO+PSO)
GOTO 215
C
C  CALCULATE LOAD PROPERTIES FOR GIVEN PEAK PRESSURE
200 GOTO(205,210),LOC
205 PSO=(PR-14.0*PO+SQRT(196.0*PO*PO+196.0*PO*PR+PR*PR))/16.0
GOTO 215
210 PSO=PR
215 PDO=2.5*PSO*PSO/(7.0*PO+PSO)
U=CO*SQRT(1.0+(6.0*PSO)/(7.0*PO))
TO=W**0.3333/(2.2399+0.1886*PSO)
GOTO(220,225),LOC
220 TC=3.0*S/U
PC=PSO*(1-TC/TO)*EXP(-TC/TO)+PDO*(1-TC/TO)**2*EXP(-2*TC/TO)
CD=1.0
RETURN
225 TA=ZLEN/U
TA2=TA/2.0
TA2TO=TA2/TO
PA=PSO*(1-TA2TO)*EXP(-TA2TO)+CD*PDO*(1-TA2TO)**2*EXP(-2*TA2TO)
RETURN
C
C  CALCULATE LOAD
```

PROGRAM FLAT (CONTINUED)

```
300 GØTØ(305,310),LØC
305 TTO=TIME/TO
IF(TIME.GT.TC)GØTØ 320
P=PC+(TC-TIME)*(PR-PC)/TC
RETURN
310 TTO=(TIME-TA2)/TO
IF(TIME.GT.TA)GØTØ 320
P=PA*TIME/TA
RETURN
320 IF(TTO.GE.1.0)GØTØ 330
P=PSØ*(1-TTO)*EXP(-TTO)+CD*PDØ*(1-TTO)**2*EXP(-2*TTO)
RETURN
330 P=0
RETURN
C
C  PRINT LØAD DATA
4 IF(KINC.EQ.0)GØTØ 400
PRINT 640,LDTYPE
GØTØ 410
400 PRINT 645,LDTYPE
410 CØNTINUE
415 GØTØ(420,425),LØC
420 PRINT 650
GØTØ 430
425 PRINT 655
430 PRINT 660,W,PØ,CØ
IF(KRAND.NE.0)RETURN
GØTØ(435,440),LØC
435 PRINT 665,S,TC,PR
GØTØ 445
440 PRINT 670,ZLEN,TA,PA
445 PRINT 675,U,TO,CD,PSØ,PDØ
RETURN
C
C  LØAD TYPE 5 -- ARBITRARY LØAD SHAPE
500 GØTØ(510,520,530,540),IENTRY
C
C  INPUT LØAD DATA
510 PRINT 680
READ,NPØINT,(TT(J),PP(J),J=1,NPØINT)
FACTØR=1.0
IF(KINC.EQ.0)GØTØ 518
PMAX=PP(1)
DØ 515 J=2,NPØINT
515 IF(PP(J).GT.PMAX)PMAX=PP(J)
518 PX=PP(2)-PP(1)
TX=TT(2)-TT(1)
JJ=1
RETURN
C
C  CALCULATE MAXIMUM LØAD
520 FACTØR=PR/PMAX
GØTØ 518
RETURN
C
C  CALCULATE LØAD
530 IF(TIME.LE.TT(JJ+1))GØTØ 535
JJ=JJ+1
PX=PP(JJ+1)-PP(JJ)
TX=TT(JJ+1)-TT(JJ)
IF(TX.EQ.0)TX=1E-10
GØTØ 530
535 P=FACTØR*(PP(JJ)+(TIME-TT(JJ))*PX/TX)
RETURN
C
C  PRINT LØAD DATA
540 IF(KINC.EQ.1)PRINT 640,LDTYPE
IF(KINC.EQ.0)PRINT 645,LDTYPE
PRINT 690
DØ 545 J=1,NPØINT
P=FACTØR*PP(J)
545 PRINT 695,TT(J),P
RETURN
C
```

PROGRAM FLAT (CONTINUED)

```
14010 600 FORMAT(/*INPUT S*,+)
14020 610 FORMAT(/*INPUT PS0*,+)
14070 640 FORMAT(/*LOAD CAUSING INCIPIENT FAILURE IS AS FOLLOWS:*,
14071+     /,5X,*LOAD TYPE NUMBER*,I2)
14080 645 FORMAT(/*PROPERTIES OF LOAD ACTING ON WALL ARE AS FOLLOWS:*,
14081+     /,5X,*LOAD TYPE NUMBER*,I2)
14090 650 FORMAT(8X,*(FRONT FACE)*)
14100 655 FORMAT(5X,*(SIDE OR TOP FACE)*)
14110 660 FORMAT(10X,*W =*,F8.1,* KT    PO =*,F6.2,* PSI     CO =*,
14111+     F7.1,* FPS*)
14120 665 FORMAT(10X,*S =*,F6.1,* FT      TC =*,F6.3,* SEC     PR =*,
14121+     F7.3,* PSI*)
14130 670 FORMAT(10X,*L =*,F6.1,* FT      TA =*,F6.3,* SEC     PA =*,
14131+     F7.3,* PSI*)
14140 675 FORMAT(10X,*U =*,F7.1,* FPS    TO =*,F6.3,* SEC     CD =*,
14141+     F5.1,/,8X,*PS0 =*,F7.3,* PSI  PD0 =*,F7.3,* PSI*)
14150 680 FORMAT(/*INPUT NUMBER OF LOAD POINTS AND THE TIME AND *
14151+     *PRESSURE AT EACH POINT*)
14160 690 FORMAT(/10X,*TIME      PRESSURE*)
14170 695 FORMAT(F15.3,F12.2)
15000 END
20000 SUBROUTINE FILL(P3,IENTRY)
20010C: COMPUTES AVERAGE AIR PRESSURE IN ROOM DUE TO BLAST WAVE
20020C: INCIDENT HEAD-ON UPON FRONT WALL.
20030C
20050 COMMON KINC,LDTYPE,KRF,KRAND,TIME,II,Y(100),Q,QU,YU,YFAIL,
20052+  ZLS,HS,FDY,AREA,ZMASS,ZKLM,VS1,VS2,PS0,PD0,PR,PEXT,PC,TC,TO,
20054+  PO,DELAY,S
20080 DIMENSION AA(8,2),NN(8)
20090 LOGICAL L1,L2,L3
20095C
20100 GOTO(10,13,11),IENTRY
20105C
20110 10 PRINT 700
20120 READ,NWIN,V3
20122 RHO0=0.076 $ L1=.FALSE.
20123 DELAY=1E10
20125 AT=0 $ AFRONT=0 $ ASIDE=0
20130 DO 18 I=1,NWIN
20140 PRINT 710,I
20150 READ,AA(I,1),NN(I),AA(I,2)
20160 AA(I,2)=AA(I,2)/1000.0
20161 AT=AT+AA(I,1)
20162 M=NN(I) $ GOTO(12,14,14),M
20163 12 AFRONT=AFRONT+AA(I,1)
20164 GOTO 18
20165 14 ASIDE=ASIDE+AA(I,1)
20170 18 IF(AA(I,2).LT.DELAY)DELAY=AA(I,2)
20175 AFRONT=AFRONT/AT $ ASIDE=ASIDE/AT
20180 700 FORMAT(/*INPUT NUMBER OF OPENINGS AND ROOM VOLUME (CF)*,+)
20200 710 FORMAT(/*INPUT AREA (SQ FT),LOCATION CODE & DELAY(MSEC) *,
20210+     *FOR WINDOW*,I2,+)
20230 G=1.4 $ G2=1./G $ G3=1.-G2 $ G4=2./G3
20240 PP2=.1912
20250 C=SQRT(G*PO*32.*144./RHO0)
20260 TAU=2.*(V3**(1./3.))/C
20270 DT=TAU/4.0
20280 RETURN
20310C
20320 13 P30=PO
20330 TT=0.$ T0=0.
20340 RHO30=RHO0
20350 L2=.FALSE. $ L3=.FALSE.
20360 RETURN
20370C
20380 11 IF(L1)GOTO 52
20385 IF(L2.A.L3)GOTO 9
20390 52 DDT=(TIME-T0)*0.5
20395 ISTOP=2
20400 53 IF(DDT.LT.DT)GOTO 51
20410 50 DDT=0.5*DDT
20415 ISTOP=2*ISTOP
20420 GO TO 53
20430 51 CONTINUE
```

PROGRAM FLAT (CONTINUED)

```
DO 99 I=1,ISTOP
TT=T0+I*DDT
IF(TT.GT.TO)GO TO 99
DM=0. $ WW=0. $ NW=0
DO 500 K=1,NWIN
M=NN(K) $ DLY=AA(K,2)+0.000001
IF(DLY.GE.TT)GO TO 500
GOTO(15,16,16),M
15 CDF=1.0
IF(TT-TC)20,20,21
20  P11=(TC-TT)*(PR-PC)/TC+PC
P11=P11+P0
GO TO 30
16 CDF=-0.4
21  R=TT/TO $ RR=1.-R
PD=PD0*RR*RR*EXP(-2.*R)
PS=PS0*RR*EXP(-R)
P11=PS+CDF*PD
P11=P11+P0
30  RH01=RH00*((P11/P0)**G2)
IF(P11-P30)36,36,37
36  JSIGN=-1
L2=.TRUE.
303  P2=P11
RH02=((P2/P30)**G2)*RH030
X=P30/RH030
GO TO 38
37  JSIGN=+1
306 P2=PP2*P11
RH02=((P2/P11)**G2)*RH01
X=P11/RH01
38  U22=G4*(X-P2/RH02)*32.*144.
IF(U22)40,39,39
40  PRINT,*U22 NEGATIVE*,U22
STOP
39  U2=SQRT(U22)*JSIGN
DDM=U2*RH02*AA(K,1)*DDT
DM=DM+DDM
WW=WW+P11*DDM/(G3*RH01)
C
500 CONTINUE
P30=P30+(G-1.)*WW/V3
RH030=RH030+DM/V3
99  CONTINUE
T0=TT
P3=P30-P0
IF(TIME.GE.TC)L3=.TRUE.
RETURN
9 R=TIME/TO $ RR=1.0-R
PD=PD0*RR*RR*EXP(-2.0*R)
PS=PS0*RR*EXP(-R)
P3=PS+PD*(AFRONT-0.4*ASIDE)
999 RETURN
END
SUBROUTINE RESIST(IENTRY)
C * THIS SUBROUTINE DETERMINES THE RESISTANCE FUNCTION,
C:  TRANSFORMATION FACTORS, AND REACTION COEFFICIENTS FOR
C:  A REINFORCED CONCRETE FLAT SLAB
C
COMMON KINC,LDTYPE,KRF,KRAND,TIME,I,Y(100),Q,QU,YU,YFAIL,
+  ZLS,HS,FDY,AREA,ZMASS,ZKLM,VS1,VS2,PS0,PD0,PR,PEXT,PC,TC,TO,
+  P0,DELAY,S
COMMON /SHEAR/ ISHEAR,JSHEAR,VSHEAR,MEMB
REAL N,IG,MU(4),ICR(4),IC,IAVG,KE
DIMENSION AS(4),APS(4),D(4),DP(4)
C
GOTO(4,500,45),IENTRY
C
C * ENTRY 1. INPUT AND ECHO SLAB AND REINFORCEMENT PROPERTIES *
4 PRINT 600
READ,ZLS,C2,HS,FPC,FDY,ZLD,HD
FDC=1.25*FPC
EC=57619.0*SQRT(FPC)
ES=29E6
```

PROGRAM FLAT (CONTINUED)

```
AREA=ZLS*ZLS-C2*C2
ECKIP=EC/1000.0 $ ESKIP=ES/1000.0
PRINT 670
DO 8 I=1,4
PRINT 610,I
8 READ,AS(I),D(I),APS(I),DP(I)
PRINT 720
READ,ISHEAR
PRINT 711
READ,MEMB
IF(MEMB.EQ.0)GOTO 15
PRINT 706
READ,ASCS
ASCS=ASCS/12.0
C
C * DETERMINE PROPERTIES INDEPENDENT OF FDY *
15 N=ES/EC
ZMASS=150.0*AREA*HS/(150.0*1728.0)
PRINT 620,ZLS,C2,HS,FPC,FDC,ECKIP,FDY,ESKIP,ZLD,HD
PRINT 630
DO 110 I=1,4
P=AS(I)/(12.0*D(I)) $ PP=APS(I)/(12.0*D(I))
PRINT 640,I,AS(I),P,D(I),APS(I),PP,DP(I)
C * CHANGE UNITS OF REINFORCEMENT FROM SQ IN./FT TO SQ IN./IN.
AS(I)=AS(I)/12.0 $ APS(I)=APS(I)/12.0
110 CONTINUE
IG=HS**3/12+(N-1)*(AS(1)*(D(1)-HS/2)**2+APS(1)*(HS/2-DP(1))**2)
RETURN
C
C * ENTRY 2. DETERMINE WALL PROPERTIES DEPENDENT ON FDY *
45 CALL MOMENT(FDC,FDY,ES,N,0,1.0,AS,APS,D,DP,MU,ICR,IC)
SUMMP=MU(1)+MU(2)+MU(3)+MU(4)
IAVG=0.5*(IG+ICR(1))
C
C * DETERMINE RESISTANCE CURVE FOR SLAB *
QU=4.0*(ZLS+C2)*SUMMP/(ZLS*AREA)
KE=189.0*EC*IAVG/(ZLS-0.5*C2)**4
YU=QU/KE
YT=999.9
YFAIL=YU*0.1/(AS(1)/D(1))
IF(YFAIL.GT.30.0*YU)YFAIL=30.0*YU
IF(MEMB.NE.1)GOTO 25
C * TENSILE MEMBRANE BEHAVIOR *
20 TS=ASCS*FDY
YT=QU*ZLS*ZLS/(20.25*TS)
QT=QU
IF(YT.LE.YFAIL)GOTO 22
YT=YFAIL
QT=20.25*YT*TS/(ZLS*ZLS)
22 IF(YFAIL.LT.0.15*ZLS)YFAIL=0.   *ZLS
QFAIL=0.15*20.25*TS/ZLS
C
C * ADJUST LOAD-DEFLECTION CURVE FOR SLAB DEAD LOAD *
25 QDL=150.0*HS/1728.0
YDL=QDL/KE
QU=QU-QDL$ QT=QT-QDL$ QFAIL=QFAIL-QDL
YU=YU-YDL$ YT=YT-YDL$ YFAIL=YFAIL-YDL
IF(KRAND.NE.1)PRINT 633,QDL,YDL
C
C OUTPUT LOAD-DEFLECTION CURVE
IF(KRAND.EQ.1)GOTO 335
PRINT 650
IF(MEMB.EQ.1)GOTO 332
PRINT 660,QU,YU,QU,YFAIL
GOTO 335
332 IF(QT.NE.QU)GOTO 333
PRINT 660,QU,YU,QT,YT,QFAIL,YFAIL
GOTO 335
333 PRINT 660,QU,YU,QU,YT,QT,YT,QFAIL,YFAIL
335 CONTINUE
C
C * CALCULATE MINIMUM SHEAR RESISTANCE *
VPS=3.0*SQRT(FPC)
BO=C2+D(4)
```

```
QSHR1=4.0*B0*(HS+HD)*VPS/(ZLS*ZLS-B0*B0)
B0=ZLD+(D(4)-HD)
QSHR2=4.0*B0*HS*VPS/(ZLS*ZLS-B0*B0)
VBS=2.0*SQRT(F'C)
QSHR3=HS*VBS/(0.5*(ZLS-C2)-D(3))
QSHEAR=QSHR1
IF(KRAND.NE.1)PRINT 957,QSHR1,QSHR2,QSHR3
IF(QSHR2.LT.QSHEAR)QSHEAR=QSHR2
IF(QSHR3.LT.QSHEAR)QSHEAR=QSHR3
QSHEAR=QSHEAR-QDL
IF(KRAND.NE.1)PRINT 695,QSHEAR
VSHEAR=QSHEAR*AREA*0.25
RETURN
C
C * ENTRY 3. DETERMINE THE RESISTANCE AS A FUNCTIØN ØF Y(I) *
500 IF(JSHEAR.EQ.1)GØTØ 520
IF(Y(I).GT.YU)GØTØ 510
C
C ELASTIC RANGE
Q=Y(I)*KE
ZKLM=0.64
VS1=0.04 $ VS2=0.21
RETURN
C
C PLASTIC RANGE
510 IF(Y(I).GT.YFAIL)GØTØ 530
ZKLM=7.0/12.0
VS1=1.0/28.0 $ VS2=6.0/28.0
IF(Y(I).GT.YT)GØTØ 520
Q=QU
RETURN
C
C TENSILE MEMBRANE RANGE
520 Q=QT+(Y(I)-YT)*(QFAIL-QT)/(YFAIL-YT)
RETURN
C
C FAILURE (SET RESISTANCE TØ VERY SMALL VALUE)
530 Q=1E-11
RETURN
C
600 FØRMAT(/*INPUT LS,C2,  'PC,FDY,LD,HD*,+)
610 FØRMAT(*INPUT AS,D,A'S,..  FØR SECTIØN*,I2,+)
620 FØRMAT(/*PRØPERTIES ØF R/C FLAT SLAB ØR PLATE*,/
+     *  LS =*,F6.1,* IN.      C2 =*,F6.1,* IN.*,6X,*HS =*,
+     F6.1,* IN.*,/,* F'C =*,F7.1,* PSI    FDC =*,F7.1,
+     * PSI*,5X,*EC =*,F7.1,* KSI*,/,* FDY =*,F8.1,* PSI*,
+     3X,*ES =*,F8.1,* KSI*,/,*  LD =*,F6.1,* IN.      HD =*,F6.1,
+     * IN.*)
630 FØRMAT(/*REINFØRCEMENT VALUES*/* SECTIØN     AS      (P)*,
+     9X,*D*,8X,*A'S     (P')*,8X,*D'*,/,8X,*(SQ IN./FT)*,10X,
+     *(IN.)    (SQ IN./FT)*,10X,*(IN.)*)
633 FØRMAT(/*QDL =*,F6.2,* PSI      YDL =*,F8.2,* IN.*)
640 FØRMAT(I5,F11.4,* (*,F6.4,*)*,F9.3,F10.4,* (*,F6.4,*)*,F9.3)
650 FØRMAT(/*LØAD-DEFLECTIØN CURVE*,/,4X,*Q (PSI)    Y (IN.)*)
660 FØRMAT(F9.2,F12.4)
670 FØRMAT(1H )
695 FØRMAT(/*QSHEAR =*,F9.2,* PSI*)
706 FØRMAT(/*INPUT CØNTINUØUS REINFØRCEMENT (SQ IN./FT)*,+)
711 FØRMAT(/*IS TENSILE MEMBRANE TØ BE INCLUDED *,
+     *(0=NØ;1=YES)*,+)
720 FØRMAT(/*IS SHEAR FAILURE TØ BE CØNSIDERED*,
+    * (0=NØ;1=YES)*,+)
957 FØRMAT(/*QSHR1 =*,F10.3,/,*QSHR2 =*,F10.3,/,*QSHR3 =*,F10.3)
END
SUBRØUTINE MØMENT(FDC,FDY,ES,N,PV,B,AS,APS,D,DP,MU,ICR,IC)
C THIS SUBRØUTINE DETERMINES THE ULTIMATE MØMENT CAPACITY AND
C CRACKED MØMENT ØF INERTIA  FØR REQUIRED SECTIØNS
REAL K1,K2,K3,KUD,N,IC,ICTØT,MU(4),ICR(4),AS(4),APS(4),D(4),DP(4)
C
C: DETERMINE VALUES ØF CØNCRETE PARAMETERS
45 K1=0.94-FDC/26E3
K2=0.50-FDC/8E4
K3=(3900.0+0.35*FDC)/(3E3+0.82*FDC-FDC*FDC/26E3)
EPSU=0.004-FDC/65E5
```

PROGRAM FLAT (CONTINUED)

```
C: **************************************************
C: * DETERMINE ULTIMATE MOMENT CAPACITY AND CRACKED *
C: * MOMENT OF INERTIA FOR REQUIRED SECTIONS        *
C: **************************************************
C
 II=0; ICTOT=0
 DO 170 I=1,4
 MU(I)=0
 IF(AS(I).EQ.0)GOTO 170
 II=II+1
C: ALL PROPERTIES ARE COMPUTED FOR A WIDTH B
 TENS=AS(I)*FDY+PV
 IF(APS(I).LE.0)GOTO 150
C
C: WALL HAS COMPRESSION REINFORCEMENT
 C=K1*K3*FDC*B*DP(I)
 TERM1=0.5*(TENS/APS(I)+ES*EPSC)
 TERM2=ES*EPSC*(TENS-C)/APS(I)
C: DETERMINE LOCATION OF NEUTRAL AXIS
 IF(TENS.LE.C)GOTO 140
C
C: KUD > D'
 FPS=TERM1+K3*FDC/2.0-SQRT((TERM1-K3*FDC/2.0)**2
+  -(TERM2+ES*EPSC*K3*FDC))
C: F'S MUST BE <= FDY
 IF(FPS.LT.FDY)GOTO 130
 FPS=FDY
 130 TPS=APS(I)*(FPS-K3*FDC)
 KUD=(TEN-TPS)/(K1*K3*FDC*B)
 MU(I)=(TENS-TPS)*(D(I)-K2*KUD)+TPS*(D(I)-DP(I))
 ICR(I)=B*KUD**3/3.0+N*AS(I)*(D(I)-KUD)**2
+  +(N-1)*APS(I)*(KUD-DP(I))**2
 GOTO 152
C
C: KUD < D'
 140 FPS=-TERM1+SQRT(TERM1**2-TERM2)
C: F'S MUST BE <= FDY
 IF(FPS.LT.FDY)GOTO 145
 FPS=FDY
 145 TERM3=TENS+APS(I)*FPS
 KUD=TERM3/(K1*K3*FDC*B)
 MU(I)=TERM3*(D(I)-K2*KUD)-APS(I)*FPS*(D(I)-DP(I))
 ICR(I)=B*KUD**3/3+N*AS(I)*(D(I)-KUD)**2+N*APS(I)*(DP(I)-KUD)**2
 GOTO 152
C
C: WALL HAS NO COMPRESSION REINFORCEMENT
 150 KUD=TENS/(K1*K3*FDC*B)
 MU(I)=TENS*(D(I)-K2*KUD)
 ICR(I)=B*KUD**3/3.0+N*AS(I)*(D(I)-KUD)**2
C
 152 ICTOT=ICTOT+ICR(I)
 170 CONTINUE
C
C: DETERMINE AVERAGE CRACKED MOMENT OF INERTIA
 175 IC=ICTOT/II
 RETURN
 END
       SUBROUTINE RANDOM (IENTRY)
C THIS SUBROUTINE INPU   MEAN AND STANDARD DEVIATIONS FOR RANDOM
C VARIABLES;  GENERATI    NDOM VALUES;  AND CONTROLS REQUIRED
C NUMBER OF CASES TO E     N;  AND OUTPUTS FINAL RESULTS AND SUMMARY
C
 COMMON KINC,LDTYPE,KRF,KRAND,TIME,II,Y(100),Q,QU,YU,YFAIL,
+  ZLS,HS,FDY,AREA,ZMASS,ZKLM,VS1,VS2,PSO,PDO,PR,PEXT,PC,TC,TO,
+  PO,DELAY,S
 COMMON /SHEAR/ ISHEAR,JSHEAR,VSHEAR,MEMB
       COMMON /RAND/ TIMEC
 DIMENSION CHI25(7),CHI975(7),TDIST(7)
C
C VALUES FOR 97.5%  (F=19,24,29,34,39,44,49)
       DATA CHI25/.4688,.5167,.5533,.5825,.6065,.6267,.6440/
       DATA CHI975/1.7295,1.6402,1.5766,1.5284,1.4903,1.4591,1.4331/
       DATA TDIST/2.093,2.064,2.045,2.032,2.022,2.016,2.010/
C
```

```
      G0T0(5,50,70),IENTRY
    5 XDUMMY=XN0RM1(-1.0,0.0,1.0)
C INITIALIZE RANDOM NUMBER GENERAT0R
      PRINT,/,*INPUT NRAND*,
      READ,NRAND
      D0 47 I=1,NRAND
      XDUMMY=XN0RM1(0.0,0.0,1.0)
   47 C0NTINUE
      INDEX=0$  SPS0=0$  SSPS0=0
      ICHECK=20
C
C  INPUT MEAN AND STANDARD DEVIATI0N F0R RAND0M VARIABLES
      IF(KRF.EQ.0)G0T0 30
      PRINT 87
      READ,SMEAN,SSD
C  REINF0RCED C0NCRETE WALLS
   30 PRINT 86
      READ,FDYMEAN,FDYSD
      IF(KRF.EQ.1)PRINT 96
      IF(KRF.EQ.0)PRINT 97
      RETURN
C
C  GENERATE RAND0M VALUES
   50 FDY=XN0RM1(0.0,FDYMEAN,FDYSD)
      IF(FDY.LE.0)G0T0 50
      IF(KRF.EQ.0)G0T0 65
   60 S=XN0RM1(0.0,SMEAN,SSD)
      IF(S.LE.0)G0T0 60
   65 INDEX=INDEX+1
      RETURN
C SUM VALUES 0F PS0 AND PS0**2 F0R USE IN STATISTICAL ANALYSIS
   70 SPS0=SPS0+PS0
      SSPS0=SSPS0+PS0*PS0
C
C 0UTPUT FINAL RESULTS
   76 IF(KRF.EQ.1)PRINT 92,FDY,S,PS0,TIMEC
      IF(KRF.EQ.0)PRINT 90,FDY,PS0,TIMEC
      IF(JSHEAR.EQ.1)PRINT 110
      IF(JSHEAR.NE.1)PRINT 120
   80 IF(INDEX.LT.ICHECK)RETURN
C
C DETERMINE MEAN, STANDARD DEVIATI0N, AND STANDARD ERR0R F0R PS0
      ZN0=INDEX
      ZMEAN=SPS0/ZN0
      SD=SQRT((SSPS0-ZN0*ZMEAN*ZMEAN)/ZN0)
      STDERR=SD/(SQRT(ZN0-1))
C CHECK IF MAXIMUM 0F 50 PS0 SAMPLES 0BTAINED
      IF(INDEX.EQ.50)G0T0 62
C CHECK IF 95% C0NFIDENCE INTERVAL F0R MEAN PS0 VALUE IS
      IF(STDERR*TDIST((INDEX-15)/5)/ZMEAN.GT.0.10)G0T0 61
C
C C0NFIDENCE INTERVAL IS WITHIN 10%  - DETERMINE UPPER LIMIT 0F
C 95% C0NFIDENCE INTERVAL F0R STANDARD DEVIATI0N
C PR0BABILITY VALUE AND ITS 95% C0NFIDENCE INTERVAL UPPER LIMIT
   62 SDU=SD/(SQRT(CHI25((INDEX-15)/5)))
C CHECK IF MAXIMUM 0F 50 PS0 SAMPLES 0BTAINED
      IF(INDEX.EQ.50)G0T0 53
C CHECK IF UPPER VALUE 0F 95% C0NFIDENCE INTERVAL F0R STANDARD
C DEVIATI0N IS WITHIN 0.10*MEAN 0F THE STANDARD DEVIATI0N
      IF(((SDU-SD)/ZMEAN).GT.0.10)G0T0 61
C
C 95% C0NFIDENCE INTERVAL IS WITHIN 10% F0R B0TH MEAN AND 90%
C PR0BABILITY VALUE -- THEREF0RE SUFFICIENT SAMPLES 0BTAINED
C DETERMINE 95% C0NFIDENCE INTERVALS F0R MEAN, STANDARD DEVIATI0N
,
C AND 10% AND 90% PR0BABILITY VALUES
   53 ZMEANL=ZMEAN-STDERR*TDIST((INDEX-15)/5)
      ZMEANU=ZMEAN+STDERR*TDIST((INDEX-15)/5)
      SDL=SD/(SQRT(CHI975((INDEX-15)/5)))
      P10=ZMEAN-1.282*SD
      P10L=ZMEAN-1.282*SDU
      P10U=ZMEAN-1.282*SDL
      P90=ZMEAN+1.282*SD
      P90L=ZMEAN+1.282*SDL
```

PROGRAM FLAT (CONCLUDED)

```
      P90=ZMEAN+1.282*SD
      F9?U=ZMEAN+1.282*SDU
      P9OU=ZMEAN+1.282*SDU
C
C ØUTPUT STATISTICAL PARAMETERS ØF INCIPIENT CØLLAPSE PRESSURE
      PRINT 100,ZMEAN,ZMEAN' ZMEANU,SD,SDL,SDU,P10,P10L,P10U,
+       P90,P90L,P90U
      PRINT 105,INDEX,STDERR
      GØTC 999
C
C 95% CØNFIDENCE INTERVAL IS NØT WITHIN 10% FØR BØTH MEAN AND 90
B
C VALUES -- THEREFØRE ØBTAIN 5 ADDITIØNAL SAMPLES
  61 ICHECK=ICHECK+5
      RETURN
C
 86 FØRMAT(/*INPUT MEAN AND STANDARD DEVIATIØN FØR FDY*,*)
 87 FØRMAT(/*INPUT MEAN AND STANDARD DEVIATIØN FØR S*,*)
 90 FØRMAT(F9.1,F10.2,F14.3,*)
 92 FØRMAT(F9.1,F11.2,F10.2,F14.3,*)
 96 FØRMAT(///,5X,*FDY*,9X,*S*,8X,*PSØ*,6X,*CØLLAPSE TIME*)
 97 FØRMAT(///,4X,FDY,9X,*PSØ*,4X,*CØLLAPSE TIME*)
 100 FØRMAT(///,11X,*STATISTICAL PRØPERTIES ØF INCIPIENT PSØ*,
+       //,39X,*95% CØNFIDENCE LIMITS*,/,7X,*IIEM*,18X,
+       *VALUE        LØWER        UPPER*,//,* MEAN*,F29.2,
+       2F12.2,//,* STANDARD DEVIATIØN*,F15.2,2F12.2,//,
+       * 10% PRØBABILITY VALUE*,3F12.2,//,
+       * 90% PRØBABILITY VALUE*,3F12.2)
 105 FØRMAT(//,5X,*NUMBER ØF ØBSERVATIØNS  *,I3,/,5X,
+       *STANDARD ERRØR =*,F5.2)
 110 FØRMAT(5X,*(SHEAR FAILURE)*)
 120 FØRMAT(* *)
 99? STØPS END
FUNCTIØN XNØRM1(X,A,B)
IF(X)10,20,20
10 X0=RANF(-1.0)
20 X1=RANF(0.0)
X2=RANF(0.0)
Y=SQRT(-2.0*ALØG(X1))*(CØS(6.283184*X2))
XNØRM1=A+Y*B
RETURN
END
```

REFERENCES

1. Wiehle, C. K., and J. L. Bockholt, "Existing Structures Evaluation, Part I: Walls" (for Office of Civil Defense), Stanford Research Institute, Menlo Park, California (November 1968) AD-687 293.

2. Iverson, J. H., "Existing Structures Evaluation, Part II: Window Glass and Applications" (for Office of Civil Defense), Stanford Research Institute, Menlo Park, California (December 1968) AD-687 294.

3. Jensen, G. F., "Existing Structures Evaluation, Part III: Structural Steel Connections" (for Office of Civil Defense), Stanford Research Institute, Menlo Park, California (December 1969) AD-701 088.

4. Wiehle, C. K., and J. L. Bockholt, "Existing Structures Evaluation, Part IV: Two-Way Action Walls" (for Office of Civil Defense), Stanford Research Institute, Menlo Park, California (September 1970) AD-719 306.

5. Wiehle, C. K., and J. L. Bockholt, "Existing Structures Evaluation, Part V: Applications" (for Office of Civil Defense), Stanford Research Institute, Menlo Park, California (July 1971) AD-733 343.

6. Wiehle, C. K., and J. L. Bockholt, "Blast Response of Five NFSS Buildings" (for Office of Civil Defense), Stanford Research Institute, Menlo Park, California (October 1971) AD-738 547.

7. Wiehle, C. K. and J. L. Bockholt, "Dynamic Analysis of Reinforced Concrete Floor Systems" (for Defense Civil Preparedness Agency), Stanford Research Institute, Menlo Park, California (May 1973) AD-768 [illegible]06.

8. Wiehle, C. K., "Dynamic Analysis of a Building and Building Elements" (for Defense Civil Preparedness Agency), Stanford Research Institute, Menlo Park, California (April 1974), AD-A001 387.

9. Newmark, N. M., "A Method of Computation for Structural Dynamics," ASCE Transactions, Paper No. 3384, Vol. 127, Part I (1962).

10. "The Effects of Nuclear Weapons," Glasstone, S., ed., Department of Defense and Atomic Energy Commission, Washington, D.C. (February 1964).

11. "Weapons Effects Data," EM 1110-345-413, The Design of Structures to Resist the Effects of Atomic Weapons, Massachusetts Institute of Technology for the Office of Chief of Engineers, U S. Army, Washington, D.C. (July 1959).

12. Rempel, J. R. "Room Filling From Air Blast," Appendix E of Technical Report by H. L. Murphy and J. E. Beck, Slanting for Combined Nuclear Effects: Examples with Estimates, and Air Blast Room Filling, Stanford Research Institute (For Defense Civil Preparedness Agency), Menlo Park, California (June 1973) AD-783-061.

13. Tolman, D. F., R. O. Lyday, and E. L. Hill, "Statistical Classification Report, Estimated Characteristics of NFSS Inventory" (for Defense Civil Preparedness Agency), Research Triangle Institute, Research Triangle Park, North Carolina (December 1973).

14. T. Y. Lin & Associates, "A Computer Program to Analyze the Dynamic Response of High Rise Buildings to Nuclear Blast Loading," Vols. I and II, PG 80-18-1 & 2, Office of Civil Defense, Washington, D.C. (February 1964).

15. Building Exits Code, No. 101, National Fire Protection Association, Boston, Massachusetts (1963).

16. Building Code Requirements for Reinforced Concrete, ACI 318-63, American Concrete Institute, Detroit Michigan (June 1963).

NOMENCLATURE

	Area of concrete section, sq in.
	Area of tension steel in reinforced concrete slab per unit width, sq in./in.
	Area of window, sq ft
	Width of cross section, in.
	Width of wall between edge of door opening and areaway support wall, in.
	Distance from compressive face of reinforced concrete slab to centroid of tension steel, in.
	Compressive strength of 6- by 12-in. concrete cylinder, psi
	Dynamic compressive strength of concrete, psi
	Dynamic yield strength of reinforcing steel, psi
	Ultimate compressive strength of masonry unit wall, psi
	Modulus of rupture, psi
	Static yield strength of reinforcing steel, psi
	Soil depth, ft
	Lateral soil coefficient
	Span length, in.
	Horizontal length (width) of wall, in.
	Vertical length (height) of wall, in.
	Bending moment per unit width, in.-lb/in.
	Ultimate moment capacity per unit width, in.-lb/in.
	Steel ratio, tension steel
t)	Unit pressure exerted against any surface varying with time, psi
p_h	Lateral static soil pressure, psf
p_r	Reflected overpressure, psi
p_{so}	Peak incident overpressure, psi
P_T	Total lateral load acting on wall, lb
P_v	Total vertical force per unit width, lb/in.
q	Unit resistance for uniformly loaded member, psi
S	Clearing distance, ft
t_c	Clearing time, front face, sec
t_o	Duration of positive overpressure, sec
t_w	Thickness of wall, in.
V	Total shear per unit width, lb/in.
$(V_c)_s$	Total shear capacity per unit width at support, lb/in.
$(v_c)_s$	Unit shear capacity per unit width at support, psi
W	Weapon yield
γ	Unit weight, pcf
σ	Unit weight of soil, pcf
φ	Coefficient of flexure