RESPONSE OF MASONRY WALLS TO BLAST LOADING:

A DISCRETE ELEMENT ANALYSIS

By

TAHSEEN MICHAEL AL-ASWAD

Bachelor of Science in Engineering
Walla Walla College
College Place, Washington
1970

Master of Science
Tennessee Technological University
Cookeville, Tennessee
1973

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
DOCTOR OF PHILOSOPHY
May, 1979

RESPONSE OF MASONRY WALLS TO BLAST LOADING:

A DISCRETE ELEMENT ANALYSIS

Thesis Approved:

Thesis Adviser

Dean of the Graduate College

ACKNOWLEDGMENTS

The author wishes to express his sincere appreciation to the members of his advisory committee: Dr. William P. Dawkins, major adviser and chairman of the committee, for his excellent instruction, invaluable guidance, and suggestions throughout this study; Dr. Allen E. Kelly, Dr. John P. Lloyd, and Dr. Donald E. Boyd, for their helpful assistance and advisement.

A note of appreciation is also due to Ms. Charlene Fries for her excellent typing of the final manuscript.

The author is greatly indebted to his parents, Michael and Mary, brother Salem, and sister Ann, for their sacrifice, encouragement, and support.

TABLE OF CONTENTS

Chapter | Page

LIST OF TABLES

LIST OF FIGURES

Figure		Page

NOMENCLATURE

A	one-fourth the area of contact between a masonry unit and a mortar joint
a,b,c	half the length, height, and width of a masonry unit, respectively
E_i	modulus of elasticity of linear segment i of the idealized stress-strain curve for mortar
$\{F_c^A\}$	vector of centroidal forces for masonry unit A (6 x 1)
$\{F_c^A(t)\}$	vector of external centroidal forces for masonry unit A (6 x 1)
$\{F_n^A\}$	vector of nodal forces acting at the locations of the linkage elements for unit A (48 x 1)
$\{F_s(t)\}$	vector of external centroidal forces acting on the wall system (number of elements is six times the number of masonry units in the wall)
$F_u^{i1},F_v^{i1},F_w^{i1}$	forces acting in the u,v, and w directions at node i1 located near corner i on side 1 (the right side) of a masonry unit
G	shear modulus of mortar
$[G]$	geometry matrix relating nodal and centroidal displacements of a masonry unit (48 x 6)
$[G_m]$	geometry matrix for the masonry units surrounding a given unit (48 x 24 for the horizontal stack pattern, 48 x 36 for the running bond pattern)
H	clear height of the wall
h	center-to-center distance between masonry units in the y-direction (perpendicular to the bed joint)
i,j,k,ℓ,m,n,p,q	corner identification indices for a masonry unit
I_u,I_v,I_w	mass moment of inertia of a masonry unit around the u, v, and w axes, respectively

$[k]$	diagonal matrix of spring stiffnesses for a given masonry unit (48 x 48)
$k_1, k_2, k_3, \ldots, k_{48}$	spring stiffnesses in matrix $[k]$
$[K^A]$ $[K^B]$ $[K^D]$	stiffness matrices for unit under consideration (A) and unit to the right (B) and to the left (D) for horizontal stack and running bond patterns, 6 x 6 each
$[K^C]$ $[K^E]$	stiffness matrices for the units above (C) and below (E) unit A being considered (for the horizontal stack pattern, 6 x 6 each)
$[K^{CR}]$ $[K^{ER}]$ $[K^F]$ $[K^H]$	stiffness matrices for the units above (CR, ER) and below (F, H) unit A being considered (for the running bond pattern, 6 x 6 each)
$[K_s]$	system stiffness matrix (square, number of rows is six times the number of units in the system)
L	clear length of the wall
M	mass of a masonry unit
$[M]$	mass matrix of a masonry unit (diagonal, 6 x 6)
$[M_s]$	system mass matrix (diagonal, number of elements is six times the number of masonry units in the system)
$p_s(t)$	overpressure at time t for general loading distribution
$p_{sin}(t)$	overpressure at time t for sinusoidal loading distribution
P_{sin}	peak overpressure for sinusoidal loading distribution
P_{so}	peak overpressure for general loading distribution
$p_u(t)$	overpressure at time t for uniform loading distribution
P_u	peak overpressure for uniform loading distribution
$p_z(x,y,t)$	overpressure at time t for the combined sinusoidal and uniform loading distributions
t	elapsed time
t_d	duration of the positive phase of the blast wave

t_r	rise time to peak overpressure
t_x	thickness of a vertical interior mortar joint
t_y	thickness of a horizontal interior mortar joint
Δt	time increment for numerical integration
$\{U_c\},\{\ddot{U}_c\}$	vectors of centroidal displacements and accelerations, respectively, for a masonry unit (6 x 1 each)
U_m, $\dot{U}_m$, $\ddot{U}_m$	vectors of nodal displacements, velocities, and accelerations, respectively, for a masonry unit at station m in the numerical integration process
$\{U_n\}$	vector of nodal displacements for a masonry unit (48 x 1)
$\{U_s\},\{\ddot{U}_s\}$	vectors of centroidal displacements and accelerations, respectively, for the wall system (number of elements in each is six times the number of masonry units in the wall)
u,v,w	coordinate system for a masonry unit
u_{i1},v_{i1}, etc.	displacement in the direction of the given axes (u, v, etc.) at the node indicated (i_1, k_2, q_4, etc.)
x,y,z	global coordinate system
β	angle for rotation around the v-axis of a masonry unit
θ	angle for rotation around the u-axis of a masonry unit
λ_{ur},$\lambda_{u\ell}$,λ_v,λ_w	non-dimensional parameters for specifying the location of linkage elements between adjacent masonry units
ϕ	angle for rotation around the w-axis of a masonry unit

CHAPTER I

INTRODUCTION

1.1 Background

During the last three decades, increasing attention has been given to the effects of blast loading on the behavior of structures. The objectives were, first, to study the nature of the blast wave and the factors affecting its behavior both in free air and as it encounters a structure; and second, to examine and develop means of predicting the response of a structure to blast overpressures.

In spite of the fact that the most common blast loading used by investigators is that resulting from an atomic explosion, the general term "blast" refers to both fluctuations of air pressure due to man-made explosions and to vibrations induced in soil. The former, obviously, includes conventional (non-atomic) explosions, which yield blast waves comparable to the atomic blasts in nature but not in magnitude. A sonic boom may also be considered as a type of blast load.

The major effects in investigating the behavior of structures subjected to blast overpressure in the past have been focused on the response of the frame of a building. A number of assumptions were made by some investigators to incorporate the effects of the response of such structural elements as exterior walls, shear walls, and partitions on the overall behavior of the structure.

The resistance of masonry walls to blast forces can be significant. The dynamic behavior of the structure can, therefore, be different from that when only the frame is considered. Wall resistance is, of course, dependent on numerous factors such as the strength of the materials used, support conditions, and workmanship, to mention a few. Quite naturally then, response of this type of wall to blast overpressures must be treated more thoroughly in order to have a more realistic understanding of the behavior of walls and, therefore, the entire structure.

1.2 Purpose and Scope

The purpose of this investigation was to develop a mathematical model to simulate the behavior of masonry walls subjected to blast overpressure originating from a non-atomic source. The developments leading to this end may be outlined as follows:

1. Two patterns of block arrangements are treated in this dissertation, namely, the horizontal stack and the running bond. The mathematical model and the equations of motion for both patterns are developed in Chapter II.

2. Chapter III is devoted to the description of the characteristics of blast loads.

3. In Chapter IV, the physical properties of masonry walls and their constituent elements are presented.

4. Computer program "WALBLAST," for solving the equations of motion, is described in Chapter V.

5. To demonstrate the performance of the computer program, sample problems are presented in Chapter VI.

1.3 Literature Review

Research aimed at predicting the response of structures to horizontal blast loads has been underway for a significant period of time. As expected when a new field is investigated, the early efforts were limited to certain aspects of the problem. In addition, the complicated nature of both blast waves and the dynamic response of the structure made investigating the problem a difficult task to undertake. This led many investigators (25) (30) (33) (37) to develop methods aimed at establishing rapid, though for the most part approximate, techniques for design purposes. The common procedure was to excite a mass-spring system having effective vibration characteristics similar to those of the frame of a building.

The complex nature of the frame-wall interaction and its dependence on many unpredictable factors led many investigators to ignore wall resistance to applied loads. Several attempts, however, were made to study the behavior of infilled frames subjected to lateral loads. Benjamin and Williams (6) performed experiments on masonry walls to determine their effectiveness in resisting shear forces. A diagonal strut was used by Smith (31) (32) to represent the infill. Blume (8) incorporated wall stiffness into a joint rotation index in order to determine the contribution of joint rotation deformation to the overall building response. A "discrete physical model" representing the filler by a lumped mass-spring system was developed by Fedorkiw and Sozen (12) for the analysis of reinforced concrete frames with masonry filler.

Investigation of the behavior of walls subjected to transverse loading has been virtually limited to their response to static loads. A number of static tests on masonry walls were reported by Hedstrom (19) and Fishburn (13). Recently, however, there has been increasing interest in the dynamic behavior of masonry walls. Wiehle and Bockholt (38) developed a method for evaluating the strength of existing structures subjected to blast loading. Resistance functions were developed for exterior masonry and reinforced concrete walls using established analytical procedures which were then used in a dynamic analysis of a single-degree-of-freedom system. A series of tests on full-scale masonry wall panels were conducted by the URS Research Corporation (14) (15) (39) (40) using a shock tunnel facility to study various aspects of the behavior of masonry walls subjected to blast loading. Willoughby (40) reported test results on panels having both simple beam and simple plate mounting conditions. They were found to develop similar cracking patterns of those of beams and plates, respectively. Gabrielsen and Wilton studied the arching effect. Walls with "rigid" (very snugly fitted) arching were found to have failure overpressure four to five times those of non-arched walls, whereas those with "gapped" (having a gap at the top) arching were only slightly stronger than non-arched walls whose supports permit a certain degree of in-plane movement. Further tests (15) confirmed these results.

Although relatively extensive work has been done on the experimental side, an elaborate mathematical treatment was lacking. In an attempt to satisfy this need, Summers (34) developed a "grillage finite element model" whereby the wall panel is replaced by discrete plate elements joined at the nodes. Each element is, in turn, replaced by an equivalent beam network. The principle involved was that a plate bending grillage representation can replace the plate continuum with an equivalent system of beams. The model produced favorable results.

The model proposed in this dissertation treats the masonry units as rigid bodies interconnected by three-dimensional "linkage" elements at the nodes as discussed in Chapter II. The idea of the linkage elements was introduced by Ngo and Scordelis (27) in a two-dimensional form to represent the bond forces between steel bars and concrete in a finite element analysis of singly-reinforced concrete beams.

CHAPTER II

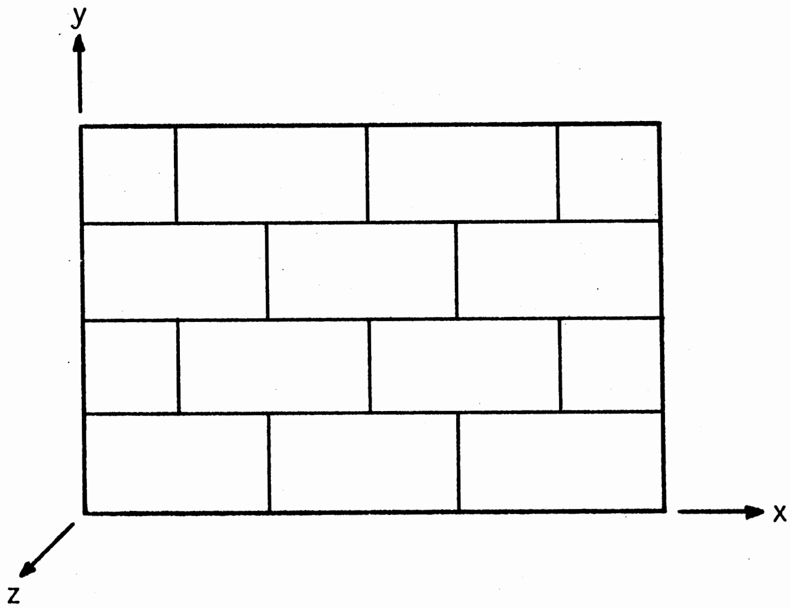
METHOD OF ANALYSIS

2.1 Wall Patterns

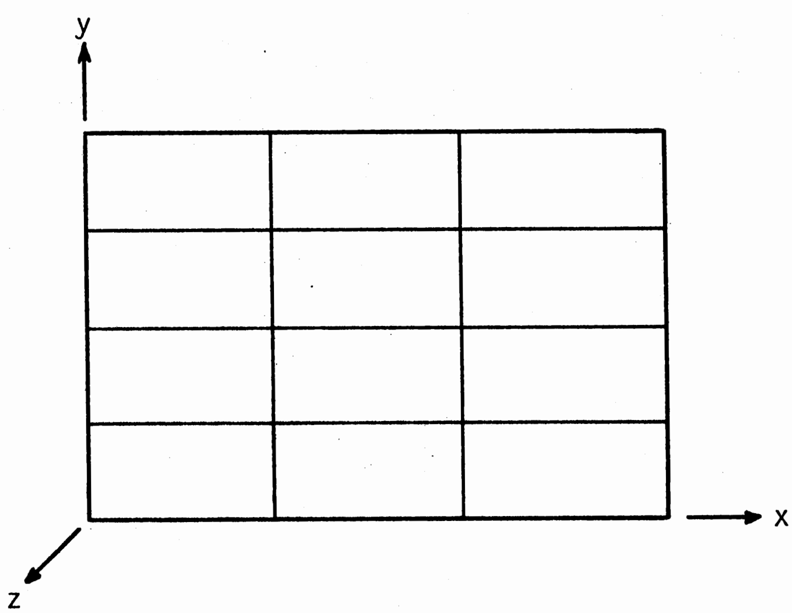
Masonry walls have traditionally been used in buildings in two major capacities: structural (load-bearing) and nonstructural, in which case walls are merely used for the protection of the interior of the building. The pattern commonly used for structural walls is the standard running bond (Figure 1(a)). Several patterns are in use for nonstructural walls. They include such patterns as the horizontal stack (Figure 1(b)), vertical stack, running bond, diagonal bond, basket weave, etc. The wall patterns treated in this analysis are the horizontal stack and the running bond.

2.2 Mathematical Model

Mortar is known to be the major contributor in developing the strength of masonry walls (10) (18) (19). It is also the critical factor in wall failures, particularly bond failures. Consequently, the importance of the role of mortar is emphasized in the chosen model. The model of a given wall panel consists of rigid masonry blocks "bound" together by a group of "linkage" elements which replace the mortar. Each linkage element is in the form of a cube for the horizontal stack pattern (a parallelepiped for the bed (horizontal) joints of the running bond pattern) containing three orthogonal springs. A typical interior portion of a modeled wall panel for each pattern is shown in Figure 2.

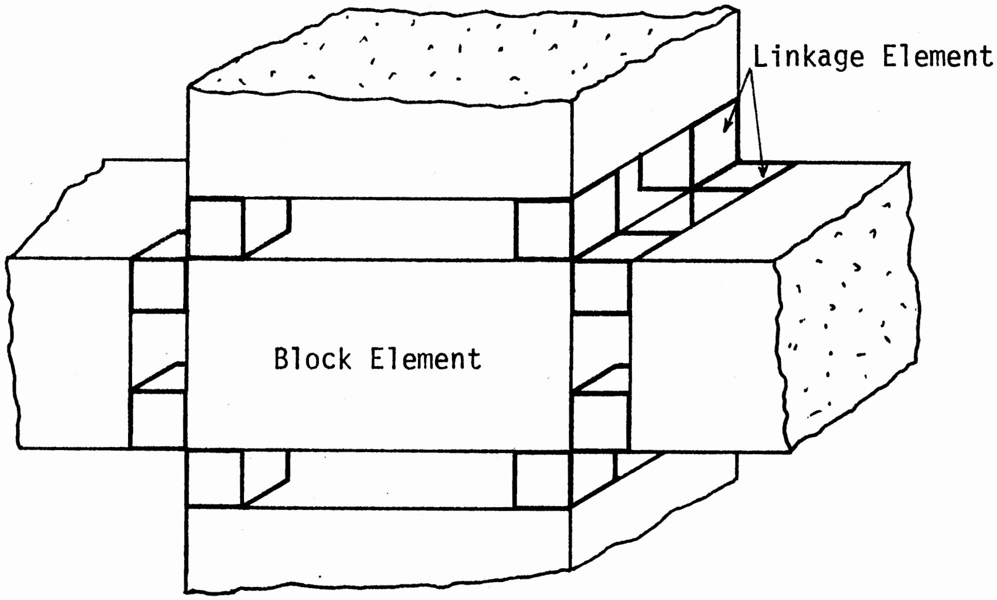


(a) Running Bond

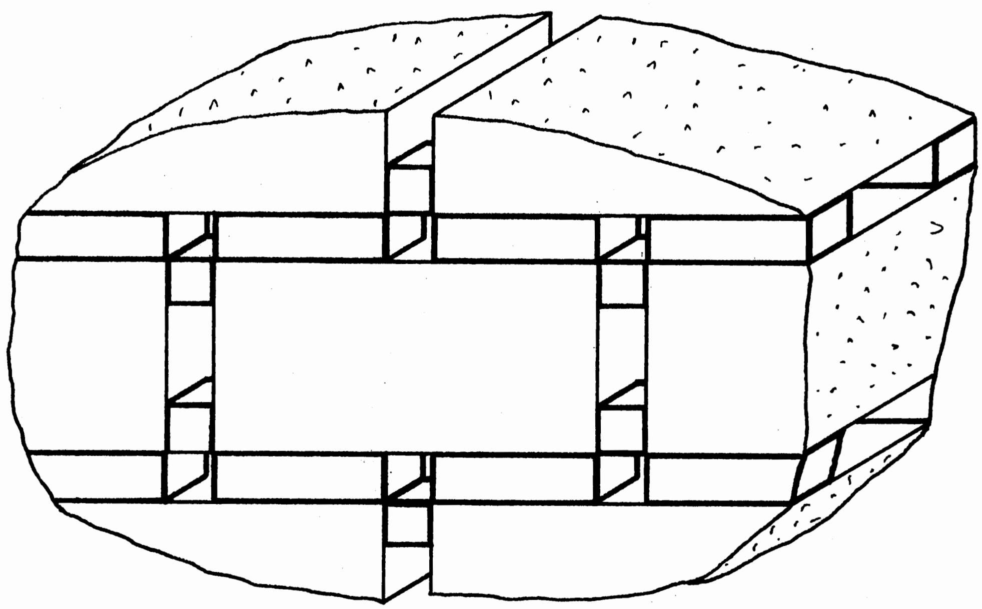


(b) Horizontal Stack

Figure 1. Wall Patterns Analyzed



(a) Horizontal Stack



(b) Running Bond

Figure 2. Model Segments

2.2.1 Block Elements

The masonry blocks are assumed to have no deformations. Thus, each block retains its original dimensions throughout the loading process. A local coordinate system is assigned to each block. Each block has six degrees of freedom (Figure 3(a)): a translation in the direction of, and a rotation about, each of the block's orthogonal axes.

2.2.2 Linkage Elements

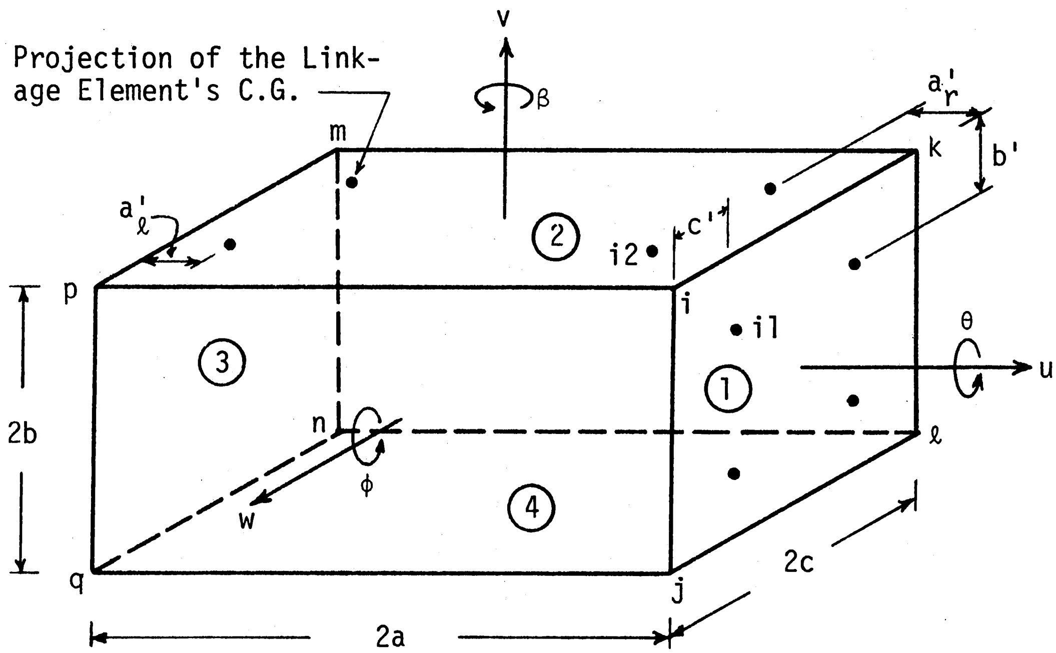
In order to simulate the mortar in the model, a three-dimensional linkage element was chosen. It consists of three orthogonal springs attached to two brackets in the form of diagonally split halves of a cube (or a parallelepiped) as shown in Figure 3(b). The springs represent the axial tensile or compressive force and the two planar shear forces. Linkage element surfaces normal to the axial spring are considered mounted to the blocks on the opposite sides.

A mortar joint between two adjacent blocks is represented by four linkage elements. The location of each element on a given face must be chosen in such a way as to give the best representation of the mortar. Therefore, the location of the projection of the element center of gravity on the side of a block (Figure 3(a)) was defined in terms of the adjustable nondimensional factors λ_{ur}, $\lambda_{u\ell}$, λ_v, and λ_w for locations in the u, v, and w directions as follows:

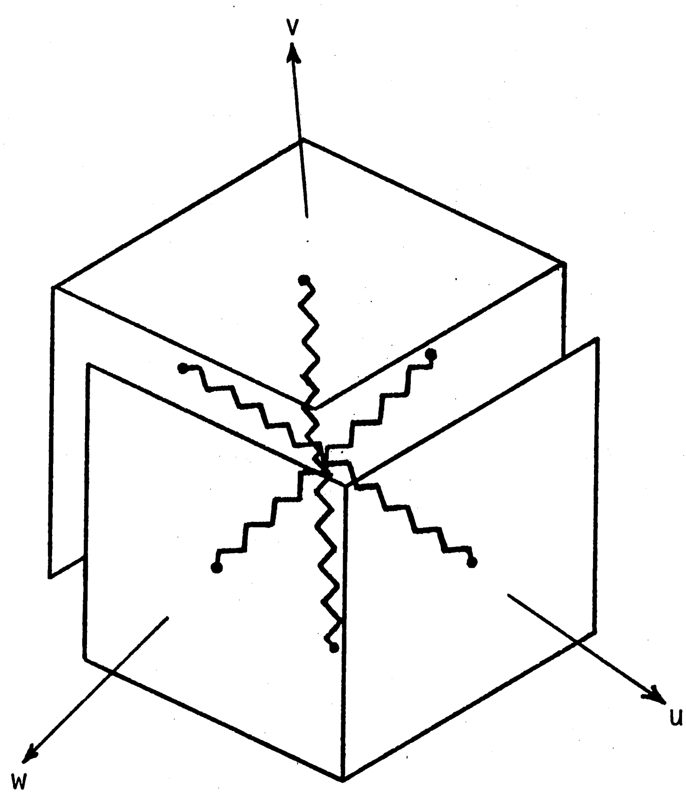
$$a_r' = a\lambda_{ur}; \quad a_\ell' = a\lambda_{u\ell}; \quad b' = b\lambda_v; \quad c' = c\lambda_w. \tag{2.1}$$

2.3 Equations of Motion

In order to formulate the equations of motion for a typical block,



(a) Block Element and Positive Degrees of Freedom



(b) Linkage Element

Figure 3. Model Elements

it is necessary to develop some geometrical relationships. As mentioned earlier, the blocks are assumed to be rigid. Consequently, all nodal displacements of a block can be conveniently related to those at its center of gravity.

2.3.1 Notation

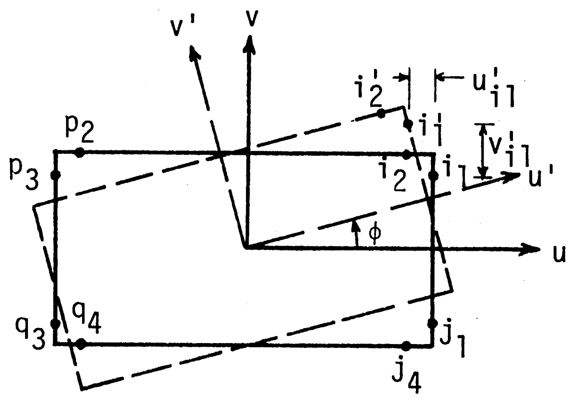
Prior to developing the geometrical relationships, it is appropriate to describe the notation used. The sides of a block are numbered from 1 to 4 in the counterclockwise direction about the w-axis, starting with the right side as shown in Figure 3(a). Block corners are assigned letter indices. Points on the block surface where the projection of the center of gravity of the linkage elements are located (hereafter referred to as "nodes") are assigned the corresponding corner letter index and side number. Thus node i1, for instance, is located near corner i on side 1.

2.3.2 Geometrical Considerations

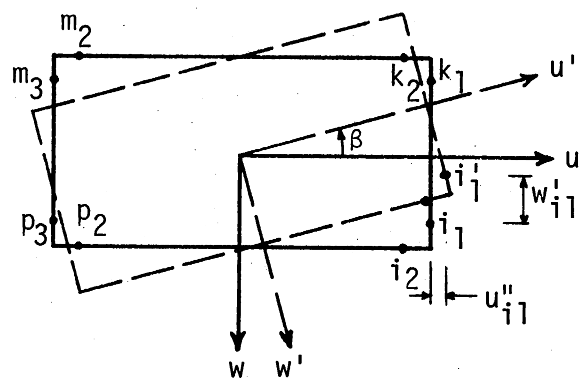
The small displacement theory will be employed to develop the relationships between the nodal and centroidal displacements. Its application, however, will be restricted to displacements occurring during a time increment Δt.

The displacements of a typical corner (e.g., i in Figure 3(a)) in the u, v, and w directions are equivalent to the corresponding centroidal displacements; they will be added to the geometrical relationships later. Figure 4(a) shows a positive rotation by a block through a small angle ϕ. The resulting horizontal and vertical displacements for corner i are

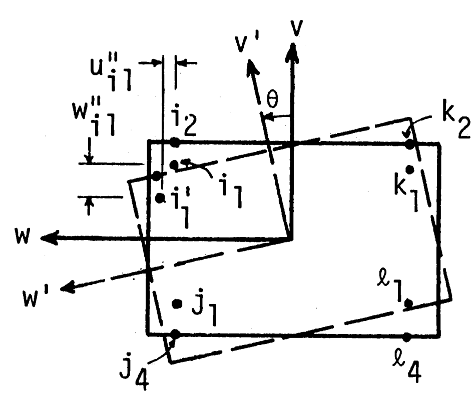
$$\begin{aligned} u_i' &= a\cos\phi - a - b\sin\phi \\ v_i' &= b\cos\phi - b + a\sin\phi \end{aligned} \qquad (2.2)$$



(a) Rotation About the w-Axis



(b) Rotation About the v-Axis



(c) Rotation About the u-Axis

Figure 4. Nodal Displacements Due to Rotation

Since ϕ is small, Equation (2.2) reduces to

$$u'_i = -b\phi \quad \text{and} \quad v'_i = a\phi \tag{2.3}$$

The displacements at node i1 can be determined using Equation (2.3):

$$\begin{aligned} u'_{i1} &= u'_i + b\sin\phi \\ v'_{i1} &= v'_i + b' - b'\cos\phi \end{aligned} \tag{2.4a}$$

Substitution of the expressions for u'_i and v'_i from Equation (2.3), for b' from Equation (2.1), and considering small displacements, allows Equation (2.4a) to be written as

$$u'_{i1} = -(1 - \lambda_v)b\phi \tag{2.4b}$$

and

$$v'_{i1} = a\phi \tag{2.4c}$$

The vertical and horizontal springs measuring shear forces are, however, located at a distance of $t_x/2$ from side 1, where t_x is the thickness of the vertical mortar joint. It follows that v'_{i1} must be measured at that location. Thus Equation (2.4c) must be modified to become

$$v'_{i1} = (a + \frac{t_x}{2})\phi \tag{2.4d}$$

In Figure 4(b), a rotation about the v-axis is considered. Following the above procedure, the displacements of node i1 due to this rotation are

$$\begin{aligned} u''_{i1} &= (1 - \lambda_w)c\beta \\ w'_{i1} &= (a + \frac{t_x}{2})\beta \end{aligned} \tag{2.5}$$

A rotation about the u-axis is illustrated in Figure 4(c). This in turn yields the following displacements for node i1:

$$w''_{i1} = (1 - \lambda_v)b\theta$$
$$v''_{i1} = -(1 - \lambda_w)c\theta \qquad (2.6)$$

The translational displacements of node i1 in the u, v, and w directions are equivalent to those at the center of gravity as indicated earlier. The total translational displacements are, therefore,

$$u_{i1} = u + u'_{i1} + u''_{i1}$$
$$v_{i1} = v + v'_{i1} + v''_{i1} \qquad (2.7a)$$
$$w_{i1} = w + w'_{i1} + w''_{i1}$$

Equations (2.4b), (2.4d), (2.5), and (2.6) are substituted in Equation (2.7a) to obtain

$$u_{i1} = u - (1 - \lambda_v)b\phi + (1 - \lambda_w)c\beta$$
$$v_{i1} = v + (a + \frac{t_x}{2})\phi - (1 - \lambda_w)c\theta \qquad (2.7b)$$
$$w_{i1} = w - (a + \frac{t_x}{2})\beta + (1 - \lambda_v)b\theta$$

Equation (2.7b) expresses the displacements of node i1 in terms of the six centroidal displacements. Similar expressions can be obtained for the remaining fifteen nodes. These relationships are conveniently presented in matrix form in Equation (2.8a). In symbolic form, Equation (2.8a) becomes

$$\{U_n\} = [G]\{U_c\} \qquad (2.8b)$$

where

$$
\begin{Bmatrix}
u_{i1}\\ v_{i1}\\ w_{i1}\\ u_{i2}\\ v_{i2}\\ w_{i2}\\ u_{j1}\\ v_{j1}\\ w_{j1}\\ u_{j4}\\ v_{j4}\\ w_{j4}\\ u_{k1}\\ v_{k1}\\ w_{k1}\\ u_{k2}\\ v_{k2}\\ w_{k2}\\ u_{\ell 1}\\ v_{\ell 1}\\ w_{\ell 1}\\ u_{\ell 4}\\ v_{\ell 4}\\ w_{\ell 4}\\ u_{m2}\\ v_{m2}\\ w_{m2}\\ u_{m3}\\ v_{m3}\\ w_{m3}\\ u_{n3}\\ v_{n3}\\ w_{n3}\\ u_{n4}\\ v_{n4}\\ w_{n4}\\ u_{p2}\\ v_{p2}\\ w_{p2}\\ u_{p3}\\ v_{p3}\\ w_{p3}\\ u_{q3}\\ v_{q3}\\ w_{q3}\\ u_{q4}\\ v_{q4}\\ w_{q4}
\end{Bmatrix}
=
\left[\begin{array}{ccc|c|c|c}
1 & 0 & 0 & 0 & (1-\lambda_w)c & -(1-\lambda_v)b \\
0 & 1 & 0 & -(1-\lambda_w)c & 0 & (a+t_x/2) \\
0 & 0 & 1 & (1-\lambda_v)b & -(a+t_x/2) & 0 \\ \hline
1 & 0 & 0 & 0 & (1-\lambda_w)c & -(b+t_y/2) \\
0 & 1 & 0 & -(1-\lambda_w)c & 0 & (1-\lambda_{ur})a \\
0 & 0 & 1 & (b+t_y/2) & -(1-\lambda_{ur})a & 0 \\ \hline
1 & 0 & 0 & 0 & (1-\lambda_w)c & (1-\lambda_v)b \\
0 & 1 & 0 & -(1-\lambda_w)c & 0 & (a+t_x/2) \\
0 & 0 & 1 & -(1-\lambda_v)b & -(a+t_x/2) & 0 \\ \hline
1 & 0 & 0 & 0 & (1-\lambda_w)c & (b+t_y/2) \\
0 & 1 & 0 & -(1-\lambda_w)c & 0 & (1-\lambda_{ur})a \\
0 & 0 & 1 & -(b+t_y/2) & -(1-\lambda_{ur})a & 0 \\ \hline
1 & 0 & 0 & 0 & -(1-\lambda_w)c & -(1-\lambda_v)b \\
0 & 1 & 0 & (1-\lambda_w)c & 0 & (a+t_x/2) \\
0 & 0 & 1 & (1-\lambda_v)b & -(a+t_x/2) & 0 \\ \hline
1 & 0 & 0 & 0 & -(1-\lambda_w)c & -(b+t_y/2) \\
0 & 1 & 0 & (1-\lambda_w)c & 0 & (1-\lambda_{ur})a \\
0 & 0 & 1 & (b+t_y/2) & -(1-\lambda_{ur})a & 0 \\ \hline
1 & 0 & 0 & 0 & -(1-\lambda_w)c & (1-\lambda_v)b \\
0 & 1 & 0 & (1-\lambda_w)c & 0 & (a+t_x/2) \\
0 & 0 & 1 & -(1-\lambda_v)b & -(a+t_x/2) & 0 \\ \hline
1 & 0 & 0 & 0 & -(1-\lambda_w)c & (b+t_y/2) \\
0 & 1 & 0 & (1-\lambda_w)c & 0 & (1-\lambda_{ur})a \\
0 & 0 & 1 & -(b+t_y/2) & -(1-\lambda_{ur})a & 0 \\ \hline
1 & 0 & 0 & 0 & -(1-\lambda_w)c & -(b+t_y/2) \\
0 & 1 & 0 & (1-\lambda_w)c & 0 & -(1-\lambda_{u\ell})a \\
0 & 0 & 1 & (b+t_y/2) & (1-\lambda_{u\ell})a & 0 \\ \hline
1 & 0 & 0 & 0 & -(1-\lambda_w)c & -(1-\lambda_v)b \\
0 & 1 & 0 & (1-\lambda_w)c & 0 & -(a+t_x/2) \\
0 & 0 & 1 & (1-\lambda_v)b & (a+t_x/2) & 0 \\ \hline
1 & 0 & 0 & 0 & -(1-\lambda_w)c & (1-\lambda_v)b \\
0 & 1 & 0 & (1-\lambda_w)c & 0 & -(a+t_x/2) \\
0 & 0 & 1 & -(1-\lambda_v)b & (a+t_x/2) & 0 \\ \hline
1 & 0 & 0 & 0 & -(1-\lambda_w)c & (b+t_y/2) \\
0 & 1 & 0 & (1-\lambda_w)c & 0 & -(1-\lambda_{u\ell})a \\
0 & 0 & 1 & -(b+t_y/2) & (1-\lambda_{u\ell})a & 0 \\ \hline
1 & 0 & 0 & 0 & (1-\lambda_w)c & -(b+t_y/2) \\
0 & 1 & 0 & -(1-\lambda_w)c & 0 & -(1-\lambda_{u\ell})a \\
0 & 0 & 1 & (b+t_y/2) & (1-\lambda_{u\ell})a & 0 \\ \hline
1 & 0 & 0 & 0 & (1-\lambda_w)c & -(1-\lambda_v)b \\
0 & 1 & 0 & -(1-\lambda_w)c & 0 & -(a+t_x/2) \\
0 & 0 & 1 & (1-\lambda_v)b & (a+t_x/2) & 0 \\ \hline
1 & 0 & 0 & 0 & (1-\lambda_w)c & (1-\lambda_v)b \\
0 & 1 & 0 & -(1-\lambda_w)c & 0 & -(a+t_x/2) \\
0 & 0 & 1 & -(1-\lambda_v)b & (a+t_x/2) & 0 \\ \hline
1 & 0 & 0 & 0 & (1-\lambda_w)c & (b+t_y/2) \\
0 & 1 & 0 & -(1-\lambda_w)c & 0 & -(1-\lambda_{u\ell})a \\
0 & 0 & 1 & -(b+t_y/2) & (1-\lambda_{u\ell})a & 0
\end{array}\right]
\begin{Bmatrix} u \\ v \\ w \\ \theta \\ \beta \\ \phi \end{Bmatrix}
\qquad (2.8a)
$$

$\{U_n\}$ = nodal displacements vector (48 x 1);

$\{U_c\}$ = centroidal displacements vector (6 x 1); and

$[G]$ = geometry matrix (48 x 6).

2.3.3 Horizontal Stack Pattern

In the following equations the superscripts refer to the appropriate block in Figure 5. Forces at the center of gravity of block A can be related to those at the nodes using the [G] matrix as follows:

$$\{F_c^A\} = [G]^T \{F_n^A\} \tag{2.9}$$

Forces acting at the center of gravity are the applied time-dependent forces and the inertia forces which are, of course, in opposite directions. Thus,

$$\{F_c^A\} = \{F_c^A(t)\} - [M] \{\ddot{U}_c^A\} \tag{2.10}$$

where

$\{F_c^A(t)\}$ = vector of external centroidal forces (6 x 1);

$\{\ddot{U}_c^A\}$ = vector of centroidal accelerations (6 x 1); and

$[M]$ = diagonal mass matrix for a block (6 x 6).

The full mass matrix for a masonry unit is given below:

$$[M] = \begin{bmatrix} M & & & & & \\ & M & & & 0 & \\ & & M & & & \\ & & & I_u & & \\ & 0 & & & I_v & \\ & & & & & I_w \end{bmatrix} \tag{2.11}$$

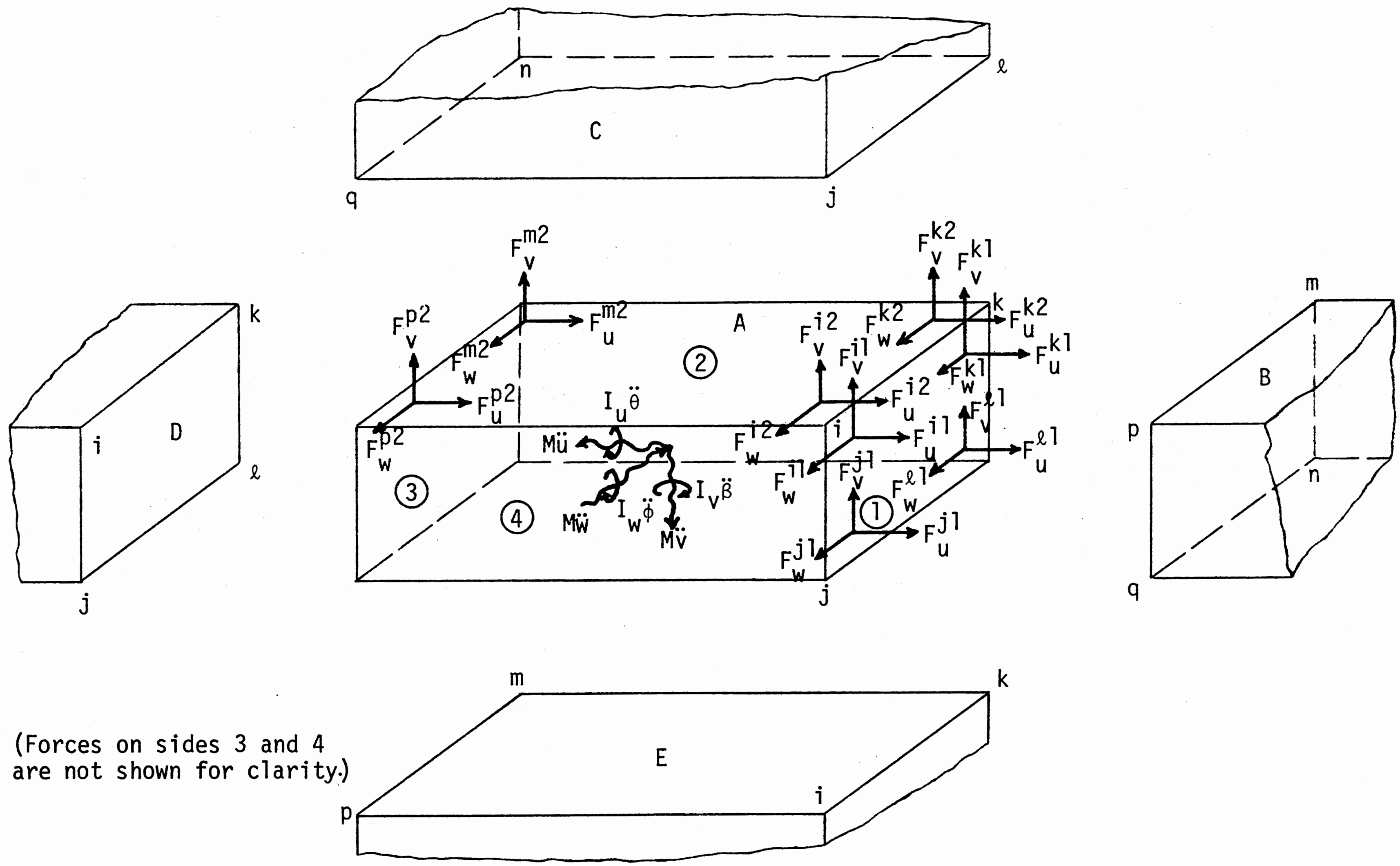


Figure 5. Nodal and Inertia Forces for the Horizontal Stack Pattern

For a solid brick:

$$M = \text{mass of the brick}$$

$$I_u = \frac{M}{3}(b^2 + c^2)$$

$$I_v = \frac{M}{3}(c^2 + a^2) \qquad (2.12)$$

$$I_w = \frac{M}{3}(a^2 + b^2)$$

For a concrete block, different expressions for I_u, I_v, and I_w are presented in Appendix A.

Nodal forces are dependent on the spring stiffnesses of the linkage elements, the relative nodal displacements of block A, and those of the surrounding blocks. These relationships are given by Equation (2.13a) on the following page. The same equation can be expressed as

$$\{F_n^A\} = [k] (\{U_n^A\} - \{U_n^{BCDE}\}) \qquad (2.13b)$$

where $\{U_n^{BCDE}\}$ is the vector of nodal displacements (24 x 1) of the surrounding blocks. Substitution of Equation (2.10) into (2.9) results in

$$\{F_c^A(t)\} - [M]\{\ddot{U}_c^A\} = [G]^T[k](\{U_n^A\} - \{U_n^{BCDE}\}) \qquad (2.14)$$

Combination of Equations (2.8b) and (2.14) leads to Equation (2.15a):

$$F_c^A(t) - [M]\{\ddot{U}_c^A\} = [G]^T[k]([G]\{U_c^A\} - [G_m]\{U_n^{BCDE}\}) \qquad (2.15a)$$

in which $[G_m]$ is the 48 x 24 geometry matrix of elements B, C, D, and E given on page 21. Symbolic expansion of $[G_m]$ and $\{U_n^{BCDE}\}$ in Equation (2.15a) yields

$$
\begin{bmatrix}
F_u^{11} \\ F_v^{11} \\ F_w^{11} \\ F_u^{12} \\ F_v^{12} \\ F_w^{12} \\ F_u^{j1} \\ F_v^{j1} \\ F_w^{j1} \\ F_u^{j4} \\ F_v^{j4} \\ F_w^{j4} \\ F_u^{k1} \\ F_v^{k1} \\ F_w^{k1} \\ F_u^{k2} \\ F_v^{k2} \\ F_w^{k2} \\ F_u^{l1} \\ F_v^{l1} \\ F_w^{l1} \\ F_u^{l4} \\ F_v^{l4} \\ F_w^{l4} \\ F_u^{m2} \\ F_v^{m2} \\ F_w^{m2} \\ F_u^{m3} \\ F_v^{m3} \\ F_w^{m3} \\ F_u^{n3} \\ F_v^{n3} \\ F_w^{n3} \\ F_u^{n4} \\ F_v^{n4} \\ F_w^{n4} \\ F_u^{p2} \\ F_v^{p2} \\ F_w^{p2} \\ F_u^{p3} \\ F_v^{p3} \\ F_w^{p3} \\ F_u^{q3} \\ F_v^{q3} \\ F_w^{q3} \\ F_u^{q4} \\ F_v^{q4} \\ F_w^{q4}
\end{bmatrix}
=
\begin{bmatrix}
k_1 & & & & \text{ZERO ELEMENTS} \\
 & k_2 & & & \\
 & & k_3 & & \\
 & & & \ddots & \\
\text{ZERO ELEMENTS} & & & & k_{48}
\end{bmatrix}
\begin{bmatrix}
u_{11}^A - u_{p3}^B \\ v_{11}^A - v_{p3}^B \\ w_{11}^A - w_{p3}^B \\
u_{12}^A - u_{j4}^C \\ v_{12}^A - v_{j4}^C \\ w_{12}^A - w_{j4}^C \\
u_{j1}^A - u_{q3}^B \\ v_{j1}^A - v_{q3}^B \\ w_{j1}^A - w_{q3}^B \\
u_{j4}^A - u_{12}^E \\ v_{j4}^A - v_{12}^E \\ w_{j4}^A - w_{12}^E \\
u_{k1}^A - u_{m3}^B \\ v_{k1}^A - v_{m3}^B \\ w_{k1}^A - w_{m3}^B \\
u_{k2}^A - u_{l4}^C \\ v_{k2}^A - v_{l4}^C \\ w_{k2}^A - w_{l4}^C \\
u_{l1}^A - u_{n3}^B \\ v_{l1}^A - v_{n3}^B \\ w_{l1}^A - w_{n3}^B \\
u_{l4}^A - u_{k2}^E \\ v_{l4}^A - v_{k2}^E \\ w_{l4}^A - w_{k2}^E \\
u_{m2}^A - u_{n4}^C \\ v_{m2}^A - v_{n4}^C \\ w_{m2}^A - w_{n4}^C \\
u_{m3}^A - u_{k1}^D \\ v_{m3}^A - v_{k1}^D \\ w_{m3}^A - w_{k1}^D \\
u_{n3}^A - u_{l1}^D \\ v_{n3}^A - v_{l1}^D \\ w_{n3}^A - w_{l1}^D \\
u_{n4}^A - u_{m2}^E \\ v_{n4}^A - v_{m2}^E \\ w_{n4}^A - w_{m2}^E \\
u_{p2}^A - u_{q4}^C \\ v_{p2}^A - v_{q4}^C \\ w_{p2}^A - w_{q4}^C \\
u_{p3}^A - u_{11}^D \\ v_{p3}^A - v_{11}^D \\ w_{p3}^A - w_{11}^D \\
u_{q3}^A - u_{j1}^D \\ v_{q3}^A - v_{j1}^D \\ w_{q3}^A - w_{j1}^D \\
u_{q4}^A - u_{p2}^E \\ v_{q4}^A - v_{p2}^E \\ w_{q4}^A - w_{p2}^E
\end{bmatrix}
\tag{2.13a}
$$

$$\{F_c^A(t)\} - [M]\{\ddot{U}_c^A\} = [G]^T[k]([G]\{U_c^A\} - [G^B:G^C:G^D:G^E] \begin{Bmatrix} U_c^B \\ U_c^C \\ U_c^D \\ U_c^E \end{Bmatrix}) \tag{2.15b}$$

The multiplications on the right side of Equation (2.15b) are performed to obtain

$$\begin{aligned} \{F_c^A(t)\} - [M]\{\ddot{U}_c^A\} = {} & [G]^T[k][G]\{U_c^A\} - [G]^T[k][G^B]\{U_c^B\} \\ & - [G]^T[k][G^C]\{U_c^C\} - [G]^T[k][G^D]\{U_c^D\} \\ & - [G]^T[k][G^E]\{U_c^E\} \end{aligned} \tag{2.15c}$$

or

$$\begin{aligned} \{F_c^A(t)\} - [M]\{\ddot{U}_c^A\} = {} & [K^A]\{U_c^A\} - [K^B]\{U_c^B\} - [K^C]\{U_c^C\} \\ & - [K^D]\{U_c^D\} - [K^E]\{U_c^E\} \end{aligned} \tag{2.15d}$$

where $[K] = [G]^T[k][G]$ and $[K^A]$, $[K^B]$, $[K^D]$, and $[K^E]$ are the stiffness matrices for block A and the surrounding blocks. Equation (2.15d) may be rearranged as

$$\begin{aligned} & [M]\{\ddot{U}_c^A\} + [K^A]\{U_c^A\} - [K^B]\{U_c^B\} - [K^C]\{U_c^C\} - [K^D]\{U_c^D\} - [K^E]\{U_c^E\} \\ & = \{F_c^A(t)\} \end{aligned} \tag{2.15e}$$

Equation (2.15e) is the equation of motion for a typical block in the horizontal stack pattern. The stiffness matrices are given in Appendix B.

$$
[G_m] = \left[\begin{array}{cccccccccccccccc}
\begin{matrix}1&0&0\\0&1&0\\0&0&1\end{matrix} & \begin{matrix}0\\-(1-\lambda_w)c\\(1-\lambda_v)b\end{matrix} & \begin{matrix}(1-\lambda_w)c\\0\\(a+t_x/2)\end{matrix} & \begin{matrix}-(1-\lambda_v)b\\-(a+t_x/2)\\0\end{matrix} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \begin{matrix}1&0&0\\0&1&0\\0&0&1\end{matrix} & \begin{matrix}0\\-(1-\lambda_w)c\\-(b+t_y/2)\end{matrix} & \begin{matrix}(1-\lambda_w)c\\0\\-(1-\lambda_{wr})a\end{matrix} & \begin{matrix}(b+t_y/2)\\(1-\lambda_{wr})a\\0\end{matrix} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\begin{matrix}1&0&0\\0&1&0\\0&0&1\end{matrix} & \begin{matrix}0\\-(1-\lambda_w)c\\-(1-\lambda_v)b\end{matrix} & \begin{matrix}(1-\lambda_w)c\\0\\(a+t_x/2)\end{matrix} & \begin{matrix}(1-\lambda_v)b\\-(a+t_x/2)\\0\end{matrix} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \begin{matrix}1&0&0\\0&1&0\\0&0&1\end{matrix} & \begin{matrix}0\\-(1-\lambda_w)c\\(b+t_y/2)\end{matrix} & \begin{matrix}(1-\lambda_w)c\\0\\-(1-\lambda_{wr})a\end{matrix} & \begin{matrix}-(b+t_y/2)\\(1-\lambda_{wr})a\\0\end{matrix} \\
\begin{matrix}1&0&0\\0&1&0\\0&0&1\end{matrix} & \begin{matrix}0\\(1-\lambda_w)c\\(1-\lambda_v)b\end{matrix} & \begin{matrix}-(1-\lambda_w)c\\0\\(a+t_x/2)\end{matrix} & \begin{matrix}-(1-\lambda_v)b\\-(a+t_x/2)\\0\end{matrix} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \begin{matrix}1&0&0\\0&1&0\\0&0&1\end{matrix} & \begin{matrix}0\\(1-\lambda_w)c\\-(b+t_y/2)\end{matrix} & \begin{matrix}-(1-\lambda_w)c\\-(1-\lambda_{wr})a\\0\end{matrix} & \begin{matrix}(b+t_y/2)\\(1-\lambda_{wr})a\\0\end{matrix} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\begin{matrix}1&0&0\\0&1&0\\0&0&1\end{matrix} & \begin{matrix}0\\(1-\lambda_w)c\\-(1-\lambda_v)b\end{matrix} & \begin{matrix}-(1-\lambda_w)c\\0\\(a+t_x/2)\end{matrix} & \begin{matrix}(1-\lambda_v)b\\-(a+t_x/2)\\0\end{matrix} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \begin{matrix}1&0&0\\0&1&0\\0&0&1\end{matrix} & \begin{matrix}0\\(1-\lambda_w)c\\(b+t_y/2)\end{matrix} & \begin{matrix}-(1-\lambda_w)c\\0\\-(1-\lambda_{wr})a\end{matrix} & \begin{matrix}-(b+t_y/2)\\(1-\lambda_{wr})a\\0\end{matrix} \\
0 & 0 & 0 & 0 & \begin{matrix}1&0&0\\0&1&0\\0&0&1\end{matrix} & \begin{matrix}0\\(1-\lambda_w)c\\-(b+t_y/2)\end{matrix} & \begin{matrix}-(1-\lambda_w)c\\0\\(1-\lambda_{wl})a\end{matrix} & \begin{matrix}(b+t_y/2)\\-(1-\lambda_{wl})a\\0\end{matrix} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \begin{matrix}1&0&0\\0&1&0\\0&0&1\end{matrix} & \begin{matrix}0\\(1-\lambda_w)c\\(1-\lambda_v)b\end{matrix} & \begin{matrix}-(1-\lambda_w)c\\0\\-(a+t_x/2)\end{matrix} & \begin{matrix}-(1-\lambda_v)b\\(a+t_x/2)\\0\end{matrix} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \begin{matrix}1&0&0\\0&1&0\\0&0&1\end{matrix} & \begin{matrix}0\\(1-\lambda_w)c\\-(1-\lambda_v)b\end{matrix} & \begin{matrix}-(1-\lambda_w)c\\0\\-(a+t_x/2)\end{matrix} & \begin{matrix}(1-\lambda_v)b\\(a+t_x/2)\\0\end{matrix} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \begin{matrix}1&0&0\\0&1&0\\0&0&1\end{matrix} & \begin{matrix}0\\(1-\lambda_w)c\\(b+t_y/2)\end{matrix} & \begin{matrix}-(1-\lambda_w)c\\0\\(1-\lambda_{wl})a\end{matrix} & \begin{matrix}-(b+t_y/2)\\-(1-\lambda_{wl})a\\0\end{matrix} \\
0 & 0 & 0 & 0 & \begin{matrix}1&0&0\\0&1&0\\0&0&1\end{matrix} & \begin{matrix}0\\-(1-\lambda_w)c\\-(b+t_y/2)\end{matrix} & \begin{matrix}(1-\lambda_w)c\\0\\(1-\lambda_{wl})a\end{matrix} & \begin{matrix}(b+t_y/2)\\-(1-\lambda_{wl})a\\0\end{matrix} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \begin{matrix}1&0&0\\0&1&0\\0&0&1\end{matrix} & \begin{matrix}0\\-(1-\lambda_w)c\\(1-\lambda_v)b\end{matrix} & \begin{matrix}(1-\lambda_w)c\\0\\-(a+t_x/2)\end{matrix} & \begin{matrix}-(1-\lambda_v)b\\(a+t_x/2)\\0\end{matrix} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \begin{matrix}1&0&0\\0&1&0\\0&0&1\end{matrix} & \begin{matrix}0\\-(1-\lambda_w)c\\-(1-\lambda_v)b\end{matrix} & \begin{matrix}(1-\lambda_w)c\\0\\-(a+t_x/2)\end{matrix} & \begin{matrix}(1-\lambda_v)b\\(a+t_x/2)\\0\end{matrix} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \begin{matrix}1&0&0\\0&1&0\\0&0&1\end{matrix} & \begin{matrix}0\\-(1-\lambda_w)c\\(b+t_y/2)\end{matrix} & \begin{matrix}(1-\lambda_w)c\\0\\(1-\lambda_{wl})a\end{matrix} & \begin{matrix}-(b+t_y/2)\\-(1-\lambda_{wl})a\\0\end{matrix}
\end{array}\right] \qquad (2.16)
$$

2.3.4 Running Bond Pattern

Equations (2.8), (2.9), and (2.10) are applicable to the running bond pattern also. Due to the pattern difference, however, the vector of relative nodal displacements is different. The nodal displacements can be expressed as (see Figure 6)

$$\{F_n^A\} = [k] \begin{Bmatrix} \{U_{i1}^A\} - \{U_{p3}^B\} \\ \{U_{i2}^A\} - \{U_{q4}^C\} \\ \{U_{j1}^A\} - \{U_{q3}^B\} \\ \{U_{j4}^A\} - \{U_{p2}^H\} \\ \{U_{k1}^A\} - \{U_{m3}^B\} \\ \{U_{k2}^A\} - \{U_{n4}^C\} \\ \{U_{\ell 1}^A\} - \{U_{n3}^B\} \\ \{U_{\ell 4}^A\} - \{U_{m2}^H\} \\ \{U_{m2}^A\} - \{U_{\ell 4}^E\} \\ \{U_{m3}^A\} - \{U_{k1}^D\} \\ \{U_{n3}^A\} - \{U_{\ell 1}^D\} \\ \{U_{n4}^A\} - \{U_{k2}^F\} \\ \{U_{p2}^A\} - \{U_{j4}^E\} \\ \{U_{p3}^A\} - \{U_{i1}^D\} \\ \{U_{q3}^A\} - \{U_{j1}^D\} \\ \{U_{q4}^A\} - \{U_{i2}^F\} \end{Bmatrix} \tag{2.17a}$$

which in symbolic form becomes

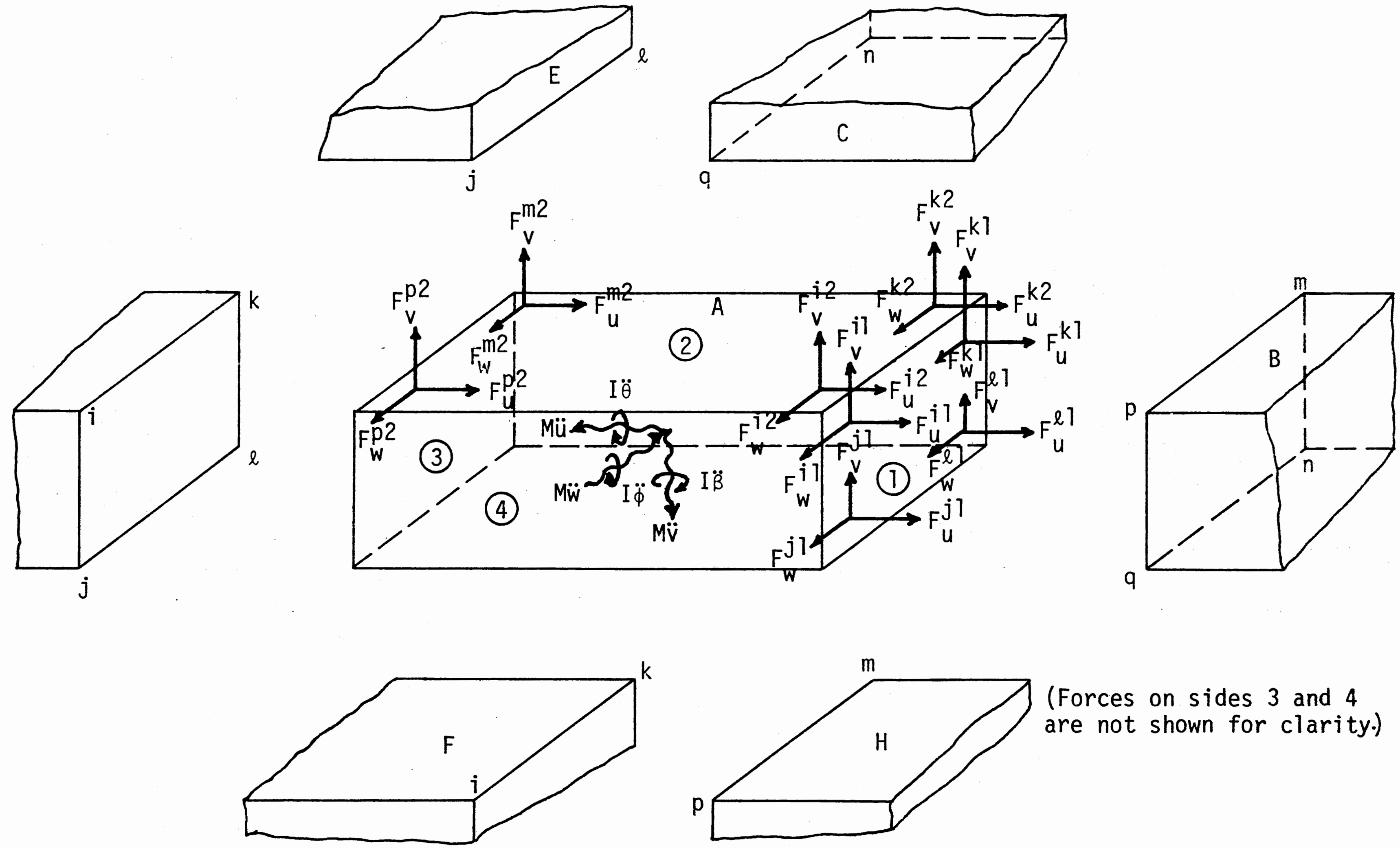


Figure 6. Nodal and Inertia Forces for the Running Bond Pattern

$$\{F_n^A\} = [k](\{U_n^A\} - \{U_n^{BCEDFH}\}) \tag{2.17b}$$

From this point on the same procedure used for the horizontal stack pattern can be applied, resulting in the following equation of motion for a typical block in the running bond pattern:

$$[M]\{\ddot{U}_c^A\} + [K^A]\{U_c^A\} - [K^B]\{U_c^B\} - [K^{CR}]\{U_c^{CR}\} - [K^{ER}]\{U_c^{ER}\}$$

$$- [K^D]\{U_c^D\} - [K^F]\{U_c^F\} - [K^H]\{U_c^H\} = \{F_c^A(t)\} \tag{2.18}$$

The stiffness matrices $[K^A]$, $[K^B]$, and $[K^D]$ are the same matrices developed for the horizontal stack pattern and are given in Appendix B. Matrices $[K^{CR}]$, $[K^{ER}]$, $[K^F]$, and $[K^H]$ are also presented in Appendix B.

2.3.5 Equations of Motion for the Wall System

The equations of motion for a complete wall can be assembled using the equation of motion for a typical block (Equation (2.15e) or (2.18)). Consider, for example, the horizontal stack wall portion shown in Figure 7. The mass matrix and vectors of accelerations, displacements, and loads are arranged as shown in Equation (2.19a). To assemble the system stiffness matrix, each block in Figure 7 is compared to the block arrangement of Figure 4. Blocks 1, 2, and 5, for instance, correspond to blocks A, B, and C, respectively, in Figure 4. Therefore, $[K^A]$, $[K^B]$, and $[K^C]$ must be located in the first row in the proper positions to be compatible with the associated displacement vectors. This process is then repeated for the other blocks. The system equations of motion for a wall with n blocks are given in Equation (2.19a). In a compact form, Equation (2.19a) becomes

$$[M_s]\{\ddot{U}_s\} + [K_s]\{U_s\} = \{F_s(t)\} \tag{2.19b}$$

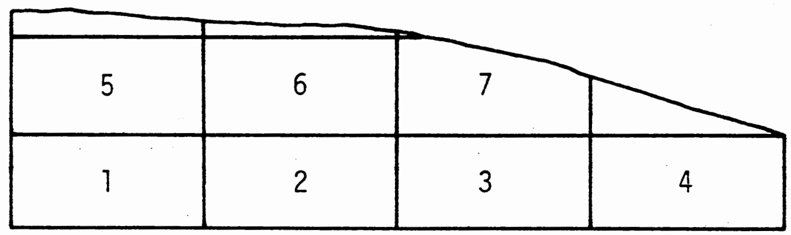


Figure 7. Portion of a Horizontal Stack Wall

$$
\underset{(6n\times 6n)}{\begin{bmatrix} [M] & & & & & & \\ & [M] & & & & & \\ & & [M] & & & & \\ & & & [M] & & & \\ & & & & [M] & & \\ & & & & & [M] & \\ & & & & & & [M] \end{bmatrix}}
\underset{(6n\times 1)}{\begin{Bmatrix} \{\ddot{U}_1\} \\ \{\ddot{U}_2\} \\ \{\ddot{U}_3\} \\ \cdot\cdot \\ \cdot\cdot \\ \cdot\cdot \\ \{\ddot{U}_n\} \end{Bmatrix}}
+
\underset{(6n\times 6n)}{\begin{bmatrix} [K_1^A] & [K_1^B] & 0 & 0 & [K_1^C] & 0 & 0 \\ [K_2^D] & [K_2^A] & [K_2^B] & 0 & 0 & [K_2^C] & 0 \\ 0 & [K_3^D] & [K_3^A] & [K_3^B] & 0 & 0 & [K_3^C] \\ \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot \\ \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot \\ \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot \\ 0 & 0 & [K_n^E] & 0 & 0 & [K_n^D] & [K_n^A] \end{bmatrix}}
\underset{(6n\times 1)}{\begin{Bmatrix} \{U_1\} \\ \{U_2\} \\ \{U_3\} \\ \cdot\cdot \\ \cdot\cdot \\ \cdot\cdot \\ \{U_n\} \end{Bmatrix}}
=
\underset{(6n\times 1)}{\begin{Bmatrix} \{F_1\} \\ \{F_2\} \\ \{F_3\} \\ \cdot\cdot \\ \cdot\cdot \\ \cdot\cdot \\ \{F_n\} \end{Bmatrix}}
\qquad (2.19a)
$$

(n is the number of elements in the wall.)

2.4 Solution of the Equations of Motion

Equation (2.19b) represents the general form of the equation of motion for both the horizontal stack and the running bond patterns. Newmark's β method (26) was used to solve these equations as outlined here.

2.4.1 Application of Newmark's Method

Equation (2.19b) can be reorganized to solve for the accelerations as follows:

$$[M_s]\{\ddot{U}_s\} = \{F_s(t)\} - [K_s]\{U_s\} \tag{2.19c}$$

Premultiplying both sides by $[M_s]^{-1}$ yields

$$\{\ddot{U}_s\} = [M_s]^{-1}(\{F_s(t)\} - [K_s]\{U_s\}) \tag{2.19d}$$

The general form of the expression for the displacements in the β method is given by

$$U_{m+1} = U_m + \dot{U}_m(\Delta t) + (\frac{1}{2} - \bar{\beta})(\Delta t)^2 \ddot{U}_m + \bar{\beta}(\Delta t)^2 \ddot{U}_{m+1} \tag{2.20a}$$

in which Δt is the time increment. For $\bar{\beta} = 1/6$

$$U_{m+1} = U_m + \dot{U}_m(\Delta t) + \frac{1}{3}\ddot{U}_m(\Delta t)^2 + \frac{1}{6}\ddot{U}_{m+1}(\Delta t)^2 \tag{2.20b}$$

The velocities are calculated from

$$\dot{U}_{m+1} = \dot{U}_m + \frac{1}{2}(\ddot{U}_m + \ddot{U}_{m+1})\Delta t \tag{2.21}$$

This procedure was implemented in the computer program as detailed in Chapter V.

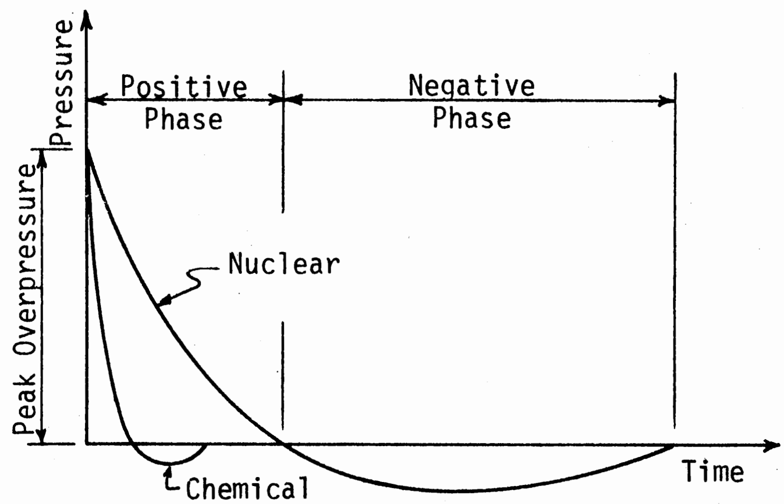
CHAPTER III

LOADING CHARACTERISTICS

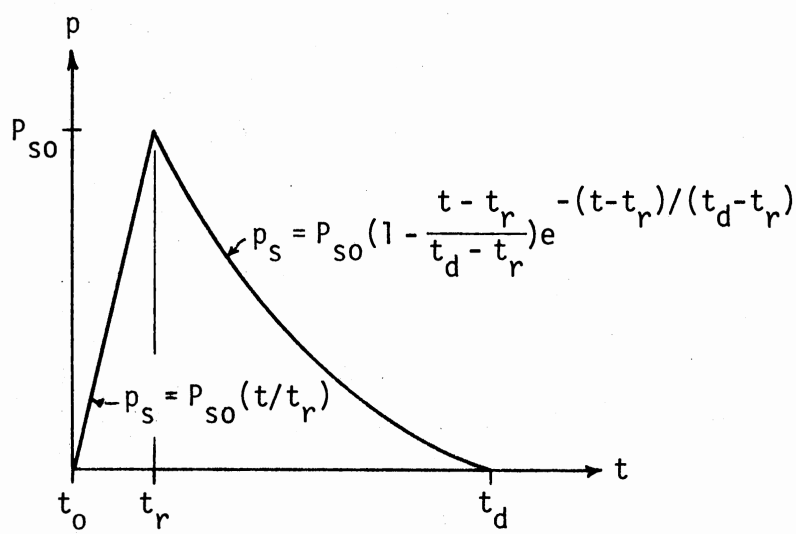
3.1 Blast Wave

A blast wave is defined by Merritt (23, p. 74) as "an atmospheric disturbance characterized by an almost discontinuous increase in pressure accompanied by simultaneous changes in temperatures, density, and particle velocity." In order to study the blast effects on a given structure, the characteristics of the blast wave must be known. Such characteristics include the peak overpressure and the pressure-time history. The variation in these factors depends on the type of explosion and the atmospheric conditions.

Although the physical properties of the energy source or explosive affect the observed characteristics of air blast waves in one or more respects, evidence indicates that at reasonable distance from the center of the explosion all blast waves, regardless of the source, share a common general configuration. Figure 8(a) shows typical pressure-time curves resulting from nuclear and chemical explosions, measured at a given location from the source of explosion. Both waves have positive as well as negative phases. The latter, however, is of minor importance and is often neglected, especially when the failure of a structural element is being investigated.



(a) Nuclear and Chemical Blast Curves



(b) Positive Phase With Rise Time

Figure 8. Pressure-Time History for Blast Waves

3.2 Non-Atomic Blast Data

A blast wave may be generated by the sudden release of stored energy which occurs in the form of an explosion. The failure of a high-pressure gas storage vessel or steam boiler are examples of non-atomic sources of explosion. Another source is that of a chemical explosion. Experimental data on explosions of this type are not readily available on a wide scale. One good source for data is a Ballistic Research Laboratories (BRL) report compiled by Goodman (16) on bare spherical pentolite charges. Information such as overpressure, duration, impulse, and distance from the explosion are tabulated and plotted in the report. Another reference is a comprehensive text by Baker (7) dealing with blast phenomena.

Blast data in the BRL report or in a similar source can be used in two ways. For a wall located at a given distance from the center of explosion of a known charge, the pressures and durations are determined from the graphs in the report. The computer program described in Chapter V can then be used to determine the wall behavior under the given circumstances. Conversely, the impulse causing failure of a given wall can be determined after running the program. The distance at which failure occurs can then be obtained from the impulse graph in the report, followed by the side-on overpressure and duration from other graphs.

Even though reference above was made to the BRL report which was prepared for pentolite, the procedure is by no means restricted to pentolite. Pentolite has been widely used in blast experiments because it gave reproducible data when detonated in small quantities. Blast properties of other chemical explosives can be obtained using conversion factors as reported by Baker (7).

3.3 Blast Loading

Blast loading on a given structure is a direct result of the presence of that structure in the path of the blast wave. The interaction between the blast wave and the structure is a complex phenomenon affected by numerous factors such as type of both the explosion and the structure, distance from the center of the explosion, and orientation with respect to the direction of the blast wave. When this information is specified, the load further depends on the face of the building being considered, since a certain pressure-time relationship exists for each face of the building.

3.3.1 Pressure-Time History

Attempting to define the form of the blast wave as a function of time in mathematical terms has not been an easy task. Expressions of varying complexity have been suggested to describe the positive phase as reported by Baker (7). They were based on theoretical and/or experimental data. The following expressions, modified from expressions in References (7) and (29) to describe the positive phase of Figure 8(b), are probably the best compromise:

$$p_s(t) = P_{so}(t/t_r) \qquad t_o < t < t_r \quad (3.1a)$$

and

$$p_s(t) = P_{so}\left(1 - \frac{t - t_r}{t_d - t_r}\right) e^{-(t-t_r)/(t_d-t_r)} \qquad t_r < t < t_d \quad (3.1b)$$

where

$p_s(t)$ = overpressure at time t;

P_{so} = peak side-on overpressure;

t_r = rise time after shock arrival; and

t_d = duration of the positive phase.

The corresponding positive impulse representing the area under the curve in Figure 8(b) is obtained by integrating Equation (3.1a, b), resulting in

$$\text{Impulse} = P_{so} \left[\frac{1}{2} t_r + (2 - e^{-1})/(t_d - t_r)\right]. \qquad (3.2)$$

3.3.2 Loading Distribution

The distribution of the blast pressure over the wall area in diffraction type (closed or almost completely closed) structures depends on a number of factors. In addition to the type and orientation of the building mentioned earlier, the loading distribution further depends on the location of the wall in the building and the total area of openings in the wall, if any. Given the many uncertainties involved in the interaction process between the blast wave and the structure, precise loading information is hard to obtain and may not be justified.

For walls with no openings, the blast wave generally yields a loading distribution which can be considered uniform, sinusoidal (e.g., on an interior wall), or a combination of both. For sinusoidal distribution Equation (3.1a, b) are expressed as

$$p_{sin}(t) = P_{sin}(t/t_r) \qquad t_o < t < t_r \qquad (3.3a)$$

and

$$p_{sin}(t) = P_{sin}\left(1 - \frac{t - t_r}{t_d - t_r}\right) e^{-(t-t_r)/(t_d-t_r)} \qquad t_r < t < t_d \qquad (3.3b)$$

Similarly, the following expressions apply for uniformly distributed load:

$$p_u(t) = P_u(t/t_r) \qquad t_o < t < t_r \qquad (3.4a)$$

and

$$p_u(t) = P_u\left(1 - \frac{t - t_r}{t_d - t_r}\right) e^{-(t-t_r)/(t_d-t_r)} \qquad t_r < t < t_d \qquad (3.4b)$$

For the general case when a combination of uniform and sinusoidal distribution is present, the pressure at any point at time t is given by

$$p_z(x,y,t) = p_u(t) + p_{sin}(t) \left(\sin \frac{\pi x}{L} \ \sin \frac{\pi y}{H}\right) \qquad (3.5)$$

in which L and H are the clear length and height of the wall, respectively (refer to Figure 1 for the coordinate system used).

In order to evaluate the external load acting on a given block, Equation (3.5) is used to determine the pressure on the block. The equivalent concentrated load, acting at the centroid of the block in the w-direction, is then calculated. This represents the third element in the vector of external forces $\{F_c^A(t)\}$ in Equations (2.15e) and (2.18).

CHAPTER IV

MATERIAL PROPERTIES AND SPRING STIFFNESSES

4.1 General

The basic elements used in masonry construction are the masonry units and their bonding agent, mortar. As a result, the strength properties of a masonry structure depend on, excluding other factors such as the method of construction and curing conditions, the properties of its elements. The properties of these elements, in turn, depend entirely on those of their constituents and on the methods of production.

The first portion of this chapter will be devoted to the description of the basic properties of masonry. The remainder will deal with the evaluation of spring stiffnesses and related properties for linkage elements.

4.2 Masonry Units

The masonry units commonly used in masonry construction are clay bricks and concrete blocks. Several versions of each type are available. Selection of the type of units to be used in a given structure depends on the specifications of the structure.

4.2.1 Clay Bricks

Common (building) brick is, as its name implies, a clay product fired at high temperatures to near-vitrification. It is available in a wide variety of shapes and qualities. The basic unit, however, is a rectangular block, solid or cored not in excess of 25 percent.

In selecting a brick for use in masonry construction, the general requirements are that it should be durable and possess sufficient strength to enable the masonry to carry the design loads. The durability of the unit would largely depend on the weather conditions at the construction site, particularly with respect to the degree of exposure to moisture and freezing conditions. Compressive strength of brick is a function of raw material, manufacturing process, degree of burning, and unit size and shape. Some of the physical properties of building bricks are tabulated in Table I. It is to be mentioned that size variations do exist, especially the length and thickness.

4.2.2 Concrete Blocks

Concrete blocks are masonry units made of water, portland cement, and various types of aggregates such as sand, gravel, crushed stone, air-cooled slag, coal cinders, expanded shale, clay, etc. Depending on the type of aggregate used, blocks of various weights are produced.

A concrete block may be solid or hollow with two or three cores. It may be load-bearing or non-load-bearing. Some of the physical properties of concrete masonry units are given in Table II.

4.3 Mortar

Mortar for masonry construction is a mixture of cementitious materials, sand (natural or manufactured), and water. The cementitious materials may consist of portland cement, masonry cement, or a

TABLE I

SELECTED PHYSICAL PROPERTIES OF CLAY BRICKS

Dimensions[a] (in.)				Compressive Strength (psi)	Unit Weight (lb/ft^3)
Type	Length	Width	Thickness		
Modular	12	4	2	2,500[b] - 20,000[c]	104 - 142
Modular	8, 12	4	$2\frac{3}{4}$, 4, $5\frac{1}{3}$		
Standard	8	$3\frac{3}{4}$	$2\frac{1}{4}$		

[a]From Reference (21).

[b]Minimum specified by ASTM C62-75a (3) for Grade MW.

[c]Maximum normally available (18).

TABLE II

SELECTED PHYSICAL PROPERTIES OF CONCRETE MASONRY UNITS

Dimensions (in.)		Compressive Strength (psi)		Unit Weight (lb/ft^3)		
		Load-Bearing[a] (Average Gross Area)	Non-Load-Bearing[b] (Average Net Area)			
Nominal	Actual			Light	Medium	Normal
8x8x16	$7\frac{5}{8}$x$7\frac{5}{8}$x$15\frac{5}{8}$	1000[c] (800)[d]	600[c] (500)[d]	up to 104	105 to 124	125 or more

[a]Minimum specified by ASTM C90-75 (3) for Grade N-I.

[b]Minimum specified by ASTM C129-75 (3).

[c]Average of three units.

[d]Individual unit.

combination thereof, and may include in addition quicklime or hydrated lime.

The American Concrete Institute (ACI) specifications (1) for concrete masonry structures recommend ASTM mortar Type M, S, or N for load-bearing non-reinforced masonry. ASTM specifications (3) for proportioning and compressive strength of mortar are listed in Table III, along with corresponding tensile strength from Reference (20).

4.4 Mortar-Unit Interaction

As indicated earlier, the strength of masonry construction depends primarily on the properties of both the masonry units and the mortar. Furthermore, the integrity of the masonry depends on how well the masonry units are joined together by the mortar. The stresses needed to separate the mortar from a masonry unit by axial or shear forces are known as the tensile bond strength and shear bond strength, respectively.

The structural bond between the mortar and the units is an important factor in the structural behavior of masonry walls. In fact, bond strength was found to have a distinct limiting effect on the flexural strength of masonry walls. In flexural strength tests on various masonry walls subjected to static transverse loads, Fishburn (13) and Hedstrom (19) reported bond failure in all walls tested in flexure. Thus, bond strength is considered the weak link in masonry walls.

Various factors are known to have an effect on the bond strength. Of major importance is having the joints between the individual units completely filled with mortar. Furthermore, mortar should have an adequate water retentivity. This can be achieved by having the maximum water content of mortar consistent with workability, along with an

TABLE III

MIX DATA AND PHYSICAL PROPERTIES OF MORTAR

Proportions (Parts by Volume)[a]					Compressive Strength[a] (psi)	Tensile Strength[b] (psi)
Mortar Type	Portland Cement	Masonry Cement	Hydr. Lime or Lime Putty	Aggregate		
M	1	1	--	The sum of the volumes of the cements and limes must be greater than 2¼ and less than 3	2500	360
	1	--	1/4			400
S	1/2	1	--		1800	280
	1	--	over 1/4 to 1/2			340
N	--	1	--		750	175
	1	--	over 1/2 to 1¼			145
O	--	1	--		350	70
	1	--	over 1¼ to 2½			120
K	1	--	over 2½ to 4		75	--

[a]ASTM specifications (3).

[b]From Reference (20).

acceptable initial rate of absorption of the masonry units. In addition, Copeland and Saxer (10) found that tensile and shear bond increased with the ratio of portland cement to the weaker cementitious constituents (lime or masonry cement) and with the compressive strength of mortar. They further noted that higher initial flow had a favorable effect on tensile bond, while increasing the air content beyond 7 to 8 percent had an adverse effect.

Noteworthy in this context is an expression proposed by Grimm (18) for calculating the bond strength of conventional mortar to brick. Based on extensive research, the expression was given in terms of the initial flow, air content, and time of exposure of mortar.

4.5 Spring Stiffnesses for Linkage Elements

As indicated in section 2.2.2, mortar between the masonry units is represented in the wall model by four linkage elements. Each linkage element encloses three springs arranged in the x, y, and z directions to measure the displacements and therefore the corresponding axial, in-plane shear, and transverse shear forces.

Spring stiffnesses k_i used in Equation (2.13) were determined for beams by applying the basic theories of strength of materials. In order to have the same shear and moment capacity in a masonry beam as those in an equivalent elastic beam, the stiffness expression for springs in the axial direction of head joints was found to be

$$k_i^A = \frac{AE}{\ell_2} \tag{4.1}$$

where

A = one-fourth of the area of contact between a masonry unit and a mortar joint;

E = modulus of elasticity of mortar; and

ℓ_2 = center-to-center distance between blocks in the x-direction.

Accordingly, each of the four axial springs must be located such that

$$c - c' = \frac{c}{\sqrt{3}} \tag{4.2}$$

Using the expression for c' from Equation (2.1), Equation (4.2) yields

$$\lambda_w = 1 - \frac{1}{\sqrt{3}} \tag{4.3}$$

The stiffness expression for transverse shear was found to be

$$k_i^{sw} = \frac{2AG}{3\ell_2} \tag{4.4}$$

in which G is the shear modulus of mortar.

For simply supported walls with two-way action, the stiffnesses obtained from the above expressions proved to be too high when the wall behavior was compared with an equivalent elastic plate. Thus, further investigation was necessary. The solution of several examples was attempted. This led to a modified version of Equation (4.4) to be used for plates. The modified expression is

$$k_i^{sw} = \frac{2AG\ (2c)^2}{3\ell_2\ h^2} \tag{4.5}$$

in which 2c is the thickness of the wall, and h is the center-to-center distance between blocks parallel to the y-axis. In addition, the stiffness of the springs representing in-plane shear, which was not important for beams, had a significant effect on the behavior of walls. The stiffness expression for these springs is

$$k_i^{sv} = \frac{AG}{\ell_2} \tag{4.6}$$

Favorable results were obtained when Equations (4.1), (4.5), and (4.6) were used for simply supported walls, with the linkage elements located such that

$$\lambda_u = \frac{1}{3}$$

for the horizontal stack pattern and 0.488 for the running bond pattern (due to geometry),

$$\lambda_v = \frac{1}{3}$$

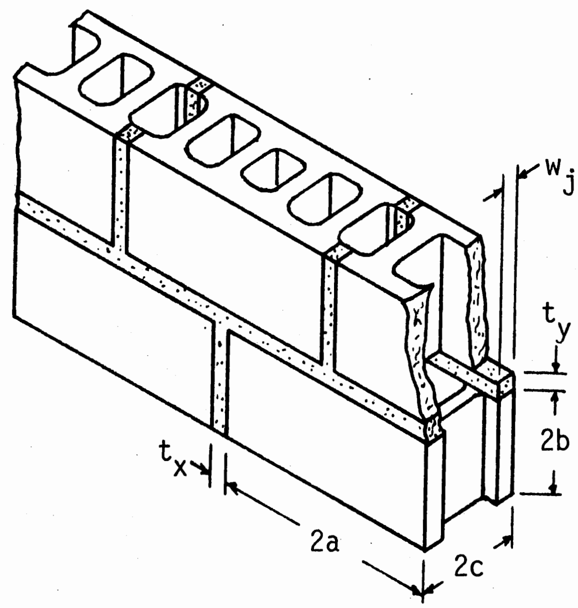
and

$$\lambda_w = 1 - \frac{1}{\sqrt{3}}$$

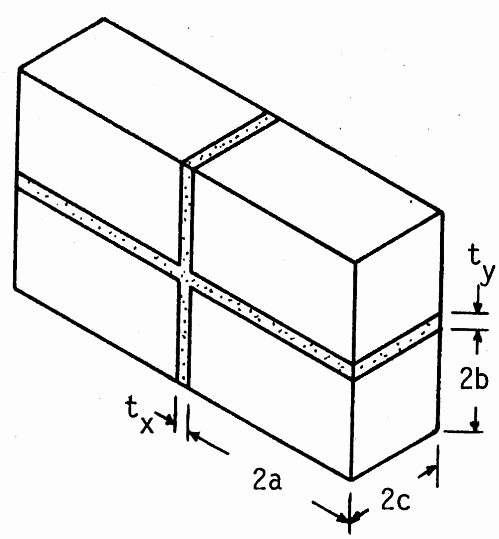
Using Equations (4.1), (4.4), (4.5), and (4.6) and referring to Figure 9, the axial and shear stiffness formulas for head as well as bed joints in terms of the dimensions of the mortar and the masonry units were developed. These expressions are given in Table IV. It should be noticed that when hollow concrete units are used, mortar for bed joints is commonly spread over the face shells only. Thus, the area of contact was considered as such for this case.

4.5.1 Modulus of Elasticity of Mortar

Illustrated in Figure 10 is the nonlinear nature of the stress-strain curve for mortar. It can be seen, however, that up to a stress of 60 to 70 percent of the ultimate strength, the relationship is



(a) Concrete Blocks in Running Bond Pattern



(b) Solid Bricks in Horizontal Stack Pattern

Figure 9. Typical Masonry Unit Arrangements

TABLE IV

STIFFNESS EXPRESSIONS FOR LINKAGE ELEMENTS

Type of Masonry Unit	Head Joints			Bed Joints		
	Axial	Shear (v)	Shear (w)	Axial	Shear (u)	Shear (w)
Hollow Concrete Blocks	$\frac{b\,w_j\,E_i}{\ell_2}$ [a][b]	$\frac{b\,w_j\,G}{\ell_2}$ [c]	$\frac{2b\,w_j\,G}{3\ell_2}\,(r)$ [d]	$\frac{a\,w_j\,E_i}{h}$ [e]	$\frac{a\,w_j\,G}{h}$	$\frac{2a\,w_j\,G}{3h}\,(s)$ [f]
Solid Masonry Units	$\frac{bc\,E_i}{\ell_2}$	$\frac{bc\,G}{\ell_2}$	$\frac{2bc\,G}{3\ell_2}\,(r)$	$\frac{ac\,E_i}{h}$	$\frac{ac\,G}{h}$	$\frac{2ac\,G}{3h}\,(s)$

[a] For compression $i = 1, 2, \ldots,$ number of segments in the stress-strain curve, for tension $i = 1$.

[b] ℓ_2 = the center-to-center distance between blocks in the x-direction.

[c] $G = E/[2(1+\nu)]$, where ν is Poisson's ratio for mortar.

[d] $r = 1$ for beams and $(2c/h)^2$ for walls.

[e] h = the center-to-center distance between blocks in the y-direction.

[f] $s = 1$ for beams and $(2c/\ell_2)^2$ for walls.

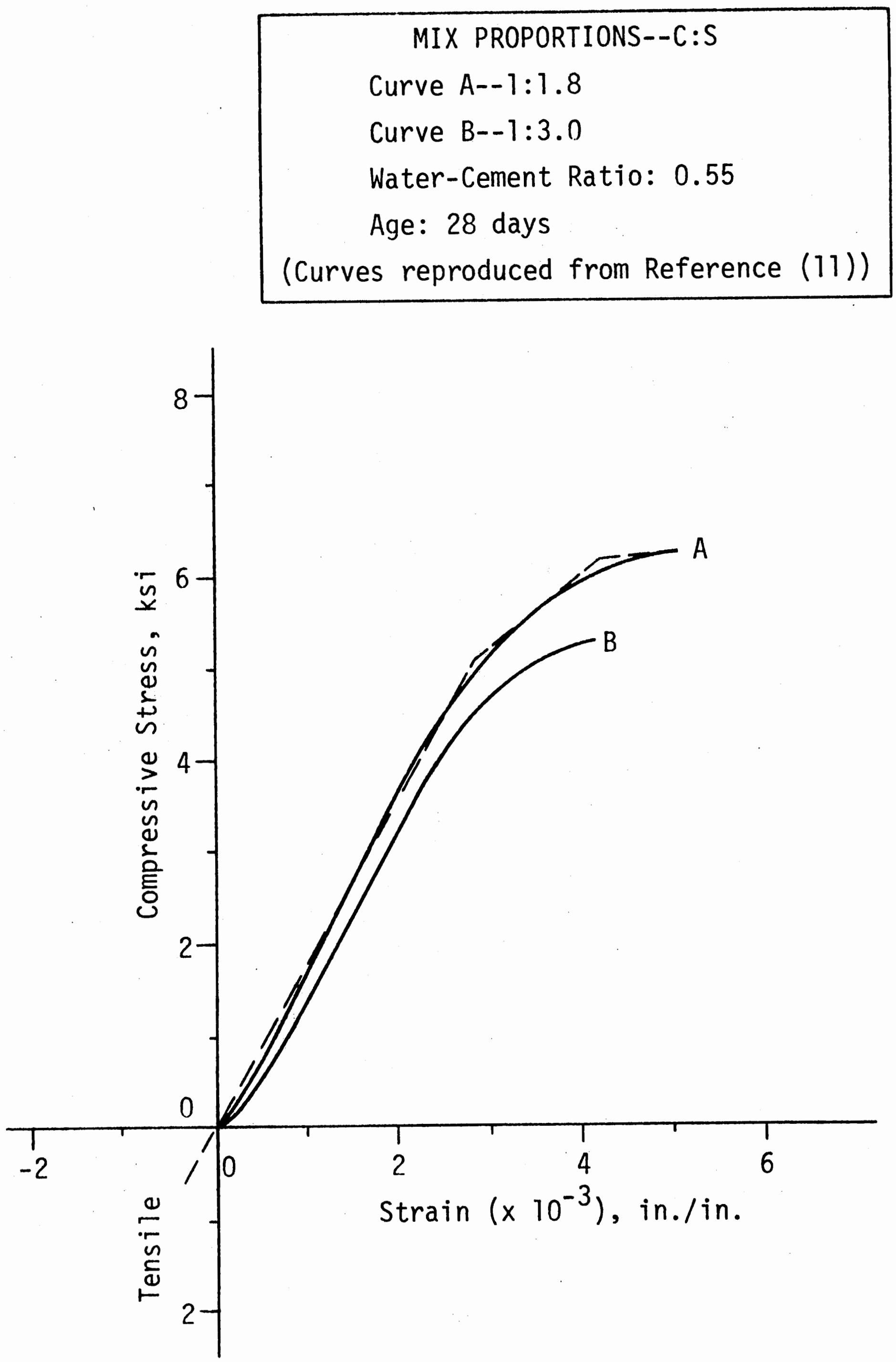


Figure 10. Typical Stress-Strain Curves of Mortar

essentially linear. Obviously, the slope and therefore the modulus of elasticity are variable in the nonlinear region.

In order to evaluate the modulus of elasticity for the entire stress-strain range, the original curve is replaced by a number of linear segments representing the original curve as closely as possible. Thus, the moduli of elasticity are reduced to a finite number. This is illustrated in Figure 10, in which curve A is replaced by the three-segment dashed line. Consequently, three moduli of elasticity are obtainable for the three linear segments.

4.5.2 Poisson's Ratio for Mortar

In evaluating the shear modulus G for mortar to be used in Equations (4.4), (4.5), and (4.6), it is necessary to know Poisson's ratio for mortar. Various mixes were tested by Anson and Newman (4). They reported an increase in Poisson's ratio with the water-cement ratio of a given mix. On the other hand, they observed a decrease in Poisson's ratio with the increase of aggregate ratio in the mix. The tests produced Poisson's ratios ranging from 0.168 to 0.227. Based on these results, an average value of 0.19 may be suggested. In general, Poisson's ratio for mortars and concretes lies between 0.1 and 0.2 (4). An average value of 0.15, therefore, seems more reasonable for general use.

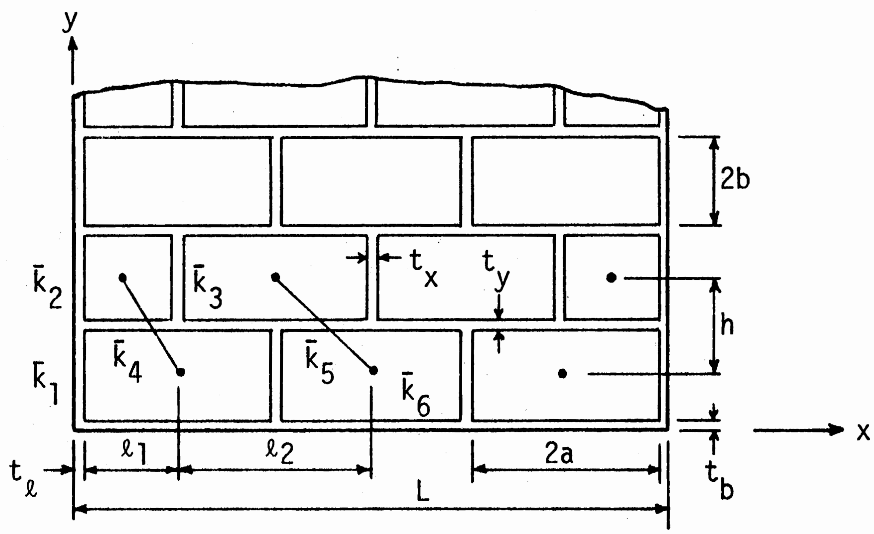
4.5.3 Stiffness Modifications for Simple Supports and Running Bond

The spring stiffness for shear in the transverse direction calculated from Equation (4.4) is the basic stiffness, applicable wherever the distance between the centroids of adjacent blocks is a block length plus a joint thickness in the x-direction or a block thickness and a joint thickness in the y-direction. In Figure 11 the basic center-to-center distance in the x-direction is ℓ_2. The distance between the center of the first block and the left support is ℓ_1. To be consistent, the stiffness of the side boundaries must be increased by the ratio ℓ_2/ℓ_1. This is true for all boundaries. It is also applicable at the head joints of the running bond pattern when one of two adjacent blocks is not a complete block. The same principle applies to the bed joints of the running bond pattern. Since the centroids of blocks in consecutive courses are not aligned along a line parallel to the y-axis, the bed joint stiffness must be reduced by the ratio of h to the actual inclined distance between the centroids of the blocks. Expressions for the modified stiffnesses are given in Figure 11.

Boundary stiffnesses obtained from the expressions of Figure 11 are the maximum values. In order to simulate a simple support, all blocks along the boundary must have the same transverse displacement. This was done by having boundary stiffnesses vary according to a sine curve along each boundary, with the expressions of Figure 11 representing the peak values. For example, the expression for the upper and lower boundaries is

$$k_{TB} = \bar{k}_6 \sin \left(\frac{\pi x}{L}\right)/[1 - \frac{b\pi}{H} \sin \left(\frac{\pi x_c}{L}\right)]$$

where $\bar{k}_6$ is defined in Figure 11, x is the distance from the left support to the spring, and x_c is the distance from the left support to the centroid of the block.



Head Joints

$$\ell_2 = 2a + t_x$$

$$\bar{k}_1 = k_i^{SW} \left(\frac{\ell_2}{a + t_\ell}\right)$$

$$\bar{k}_2 = k_i^{SW} \left(\frac{\ell_2}{\frac{1}{2}(a - t_x/2) + t_\ell}\right)$$

$$\bar{k}_3 = k_i^{SW} \left(\frac{2\ell_2}{3(a + t_x/2)}\right)$$

Bed Joints

$$h = 2b + t_y$$

$$\bar{k}_4 = k_i^{B} \left(\frac{h}{\sqrt{h^2 + [\frac{1}{2}(a + t_x/2)]^2}}\right)$$

$$\bar{k}_5 = k_i^{B} \left(\frac{h}{\sqrt{h^2 + (a + t_x/2)^2}}\right)$$

$$\bar{k}_6 = k_i^{B} \left(\frac{h}{b + t_b}\right)$$

where k_i^{SW} is the basic shear stiffness for head joints (Equation (4.5)) and k_i^B is the corresponding bed joint stiffness (Table IV).

Figure 11. Modified Shear Stiffnesses for Simple Supports and Running Bond

CHAPTER V

DESCRIPTION OF THE COMPUTER PROGRAM

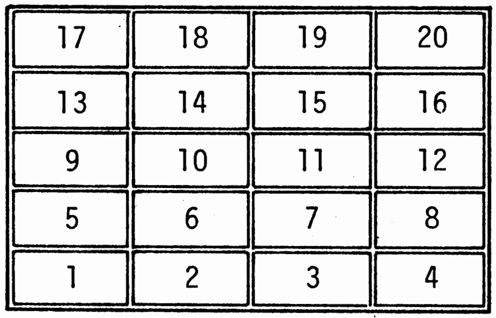
5.1 General

The computer program described herein was prepared to perform the calculations involved in solving the system equations of motion developed in Chapter II. The program is capable of analyzing the horizontal stack pattern of masonry walls as well as four versions of the running bond pattern (Figure 12). The loading provisions consist of a typical blast loading, in addition to a number of the common dynamic loads. Masonry units may be either solid or hollow-core blocks.

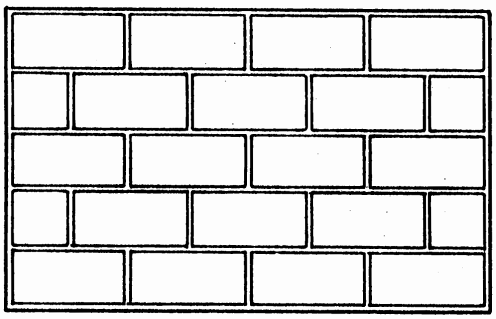
The program was written in FORTRAN IV language for solution using the IBM system 370/158 computer. Minor changes may be necessary for running the program on other types of computers having a FORTRAN compiler.

5.2 Program Organization

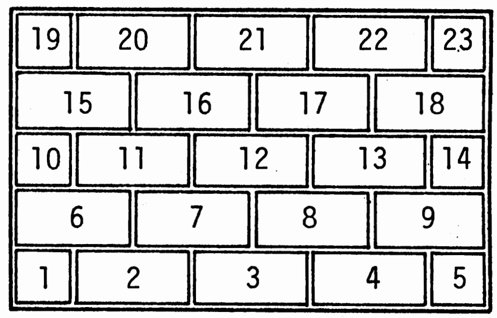
Program WALBLAST is composed of a main program, 11 subroutines, and a BLOCK DATA subprogram. The main program (Figure 13) is the organizer which basically consists of four CALL statements to execute the major subroutines in the required sequence. The CALL statements are executed as many times as the number of problems. Written in this modular form, the program offers the degree of flexibility that is necessary in long multi-function programs.



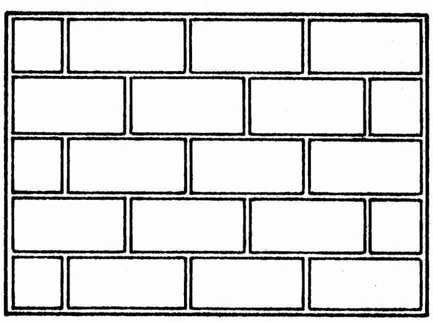
(a) Horizontal Stack
(KODE = 0)



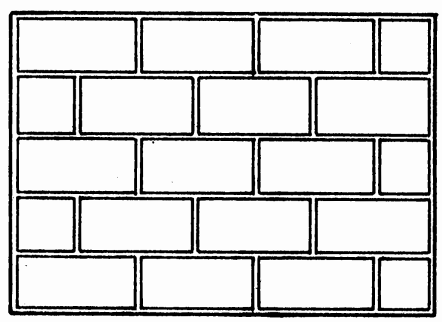
(b) Running Bond, Version I
(KODE = 1)



(c) Running Bond, Version II
(KODE = 2)



(d) Running Bond, Version III
(KODE = 3)



(e) Running Bond, Version IV
(KODE = 4)

Figure 12. Wall Patterns to be Used in the Program

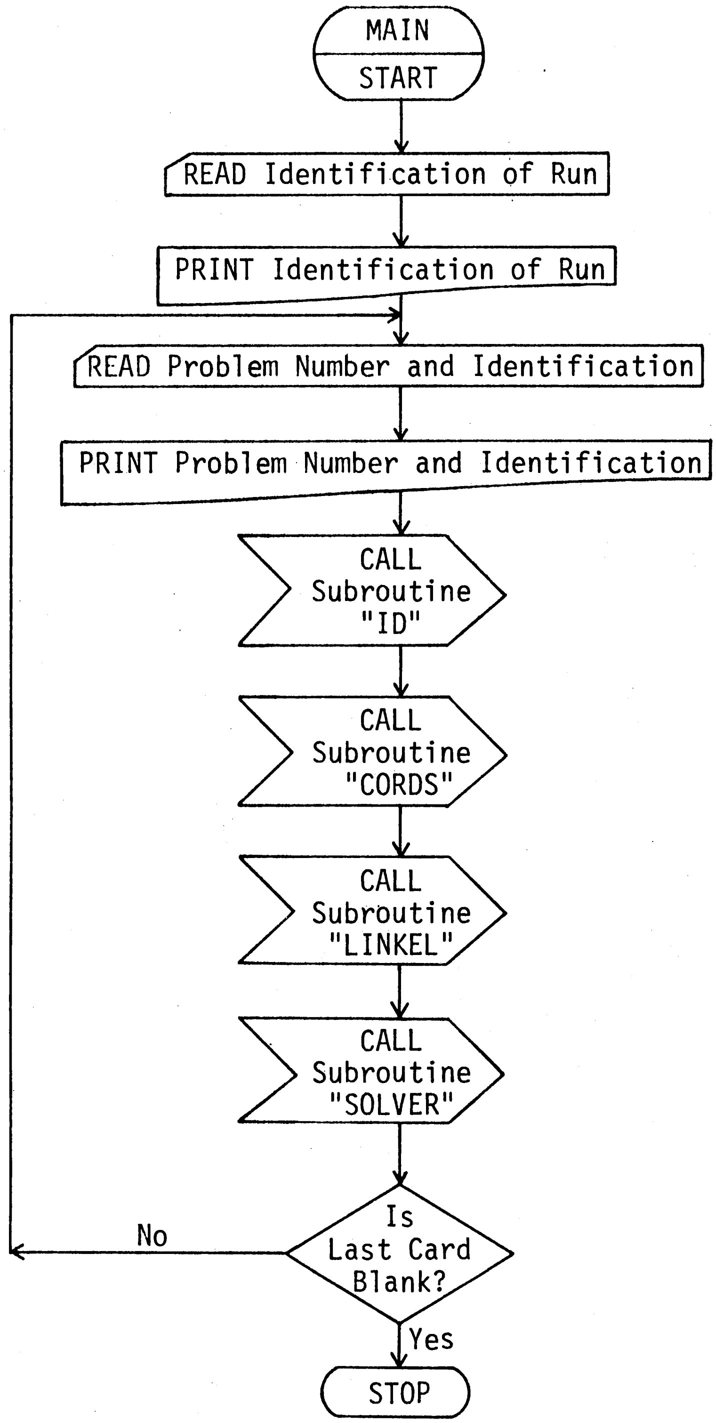


Figure 13. Flow Diagram for the Main Program

The function of each of the subroutines will be discussed in some detail in the following sections of this chapter. A listing of the program is provided in Appendix C.

5.2.1 Subroutine ID

The purpose of this subroutine was for reading and printing the data for the problem. A guide for input data is presented in Appendix D. The wall type is to be specified by the appropriate code number, as indicated in Figure 12. A summary flow diagram is given in Figure 14.

5.2.2 Subroutine CORDS

This subroutine performs two major functions. They are: (1) to determine the number of blocks in the wall, along with other information such as the number of blocks per row, the number of rows, etc.; and (2) to assign a number and determine the x, y, and z coordinates for each block with respect to the global coordinates system. A summary flow diagram is shown in Figure 15.

Examples of the numbering system used are given in Figure 12. Choosing this system was primarily due to the programming method used in association with the stiffness matrices (section 5.2.8).

5.2.3 Subroutine LINKEL

All the necessary spring stiffnesses are calculated in this subroutine using the material in section 4.5. As indicated in the flow diagram (Figure 16), the stiffnesses are calculated for interior as well as boundary springs. The expressions of Table IV were used for interior joints. At the boundary, however, the following conditions were treated:

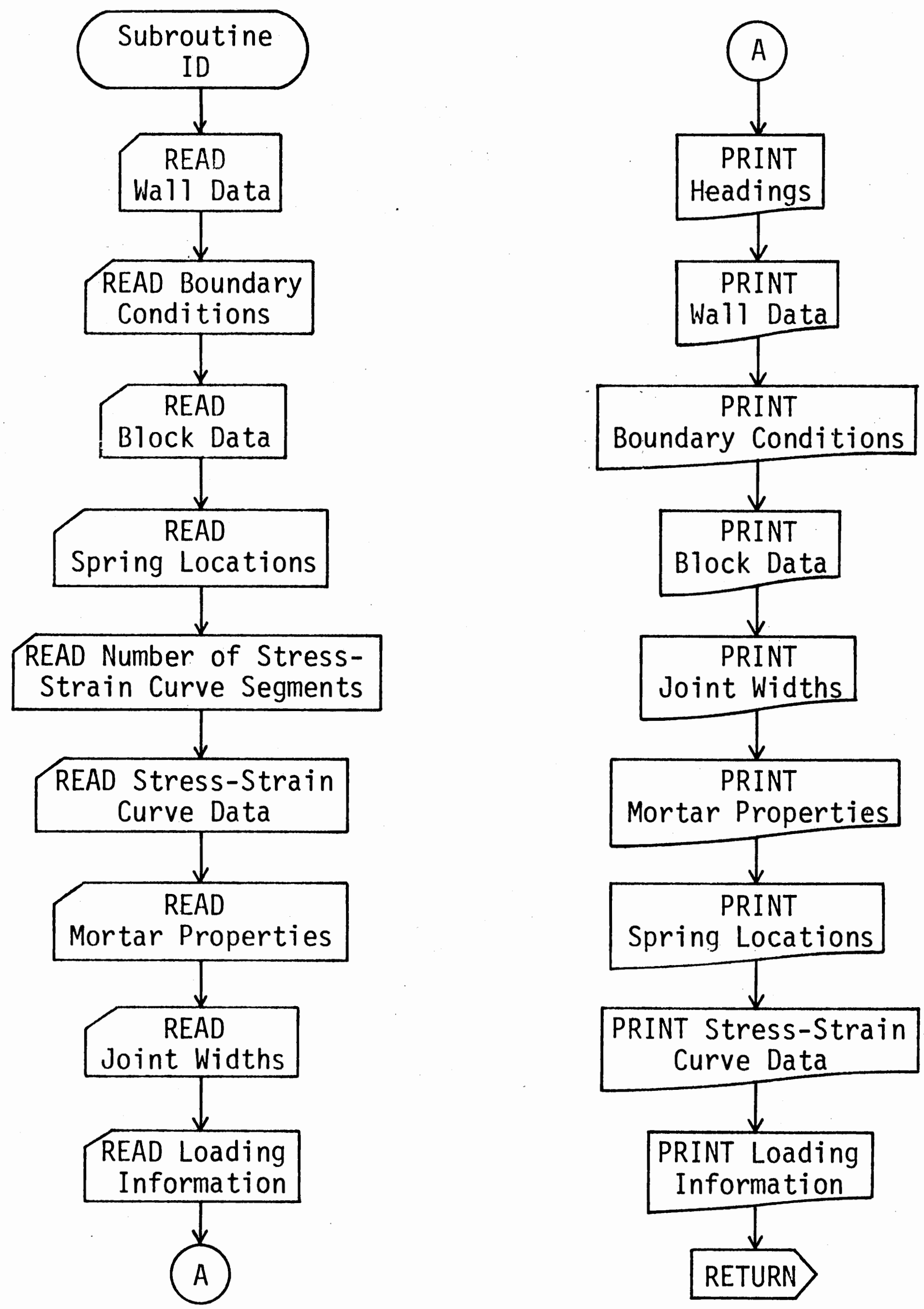


Figure 14. Summary Flow Diagram for Subroutine ID

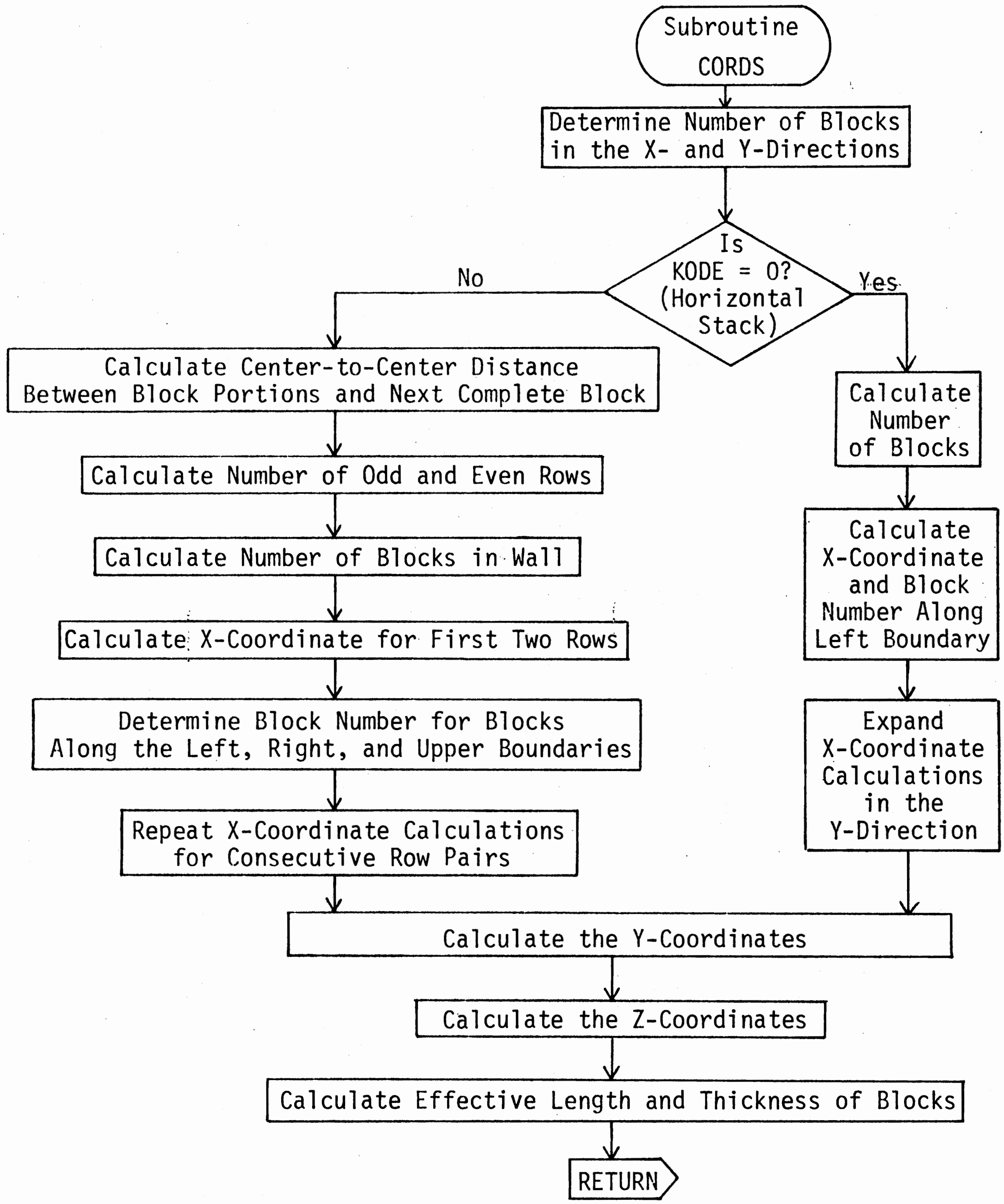


Figure 15. Summary Flow Diagram for Subroutine CORDS

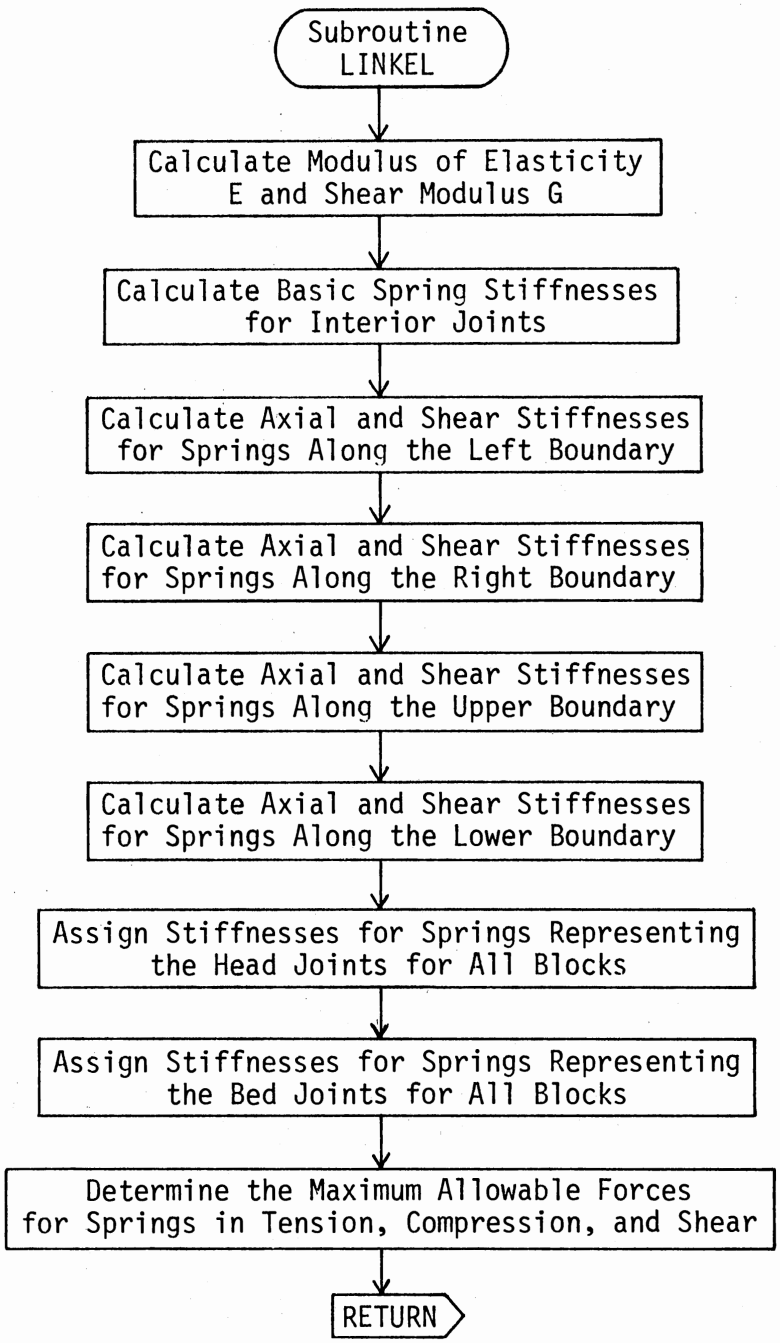


Figure 16. Summary Flow Diagram for Subroutine LINKEL

1. Free support. All spring stiffnesses were set to zero.

2. Simple support. For axial springs the stiffnesses were set to zero. The expressions given in section 4.5.3 were used for shear.

3. Symmetry condition. This condition is useful in a situation when only a portion of the wall is analyzed due to symmetry. Such symmetry may be with respect to the x-axis, y-axis, or both. Since walls with running bond patterns are not symmetric, this condition is restricted to the horizontal stack pattern. The spring stiffensses specified for this case are those of a similar interior joint.

Noteworthy is the fact that the modulus of elasticity was neglected in all calculations in this subroutine. The task of multiplying each stiffness value by the proper modulus of elasticity is performed in subroutine STIF (section 5.2.6).

5.2.4 Subroutines LOAD, STAHS, and STARB

The loads on a given wall are stored by the program in a two-dimensional array (P) having six rows and as many columns as the number of blocks in the wall. The six rows allow for a load in the direction of each of the degrees of freedom of a block. The loads applied in this study are the static in row 2 and the dynamic in row 3. All loads on a block are considered to be concentrated at its centroid.

The static load is due to the wall's own weight. For the horizontal stack pattern the static load is calculated in subroutine STAHS. For the running bond pattern this function is performed by subroutine STARB. Depending on the type of wall being analyzed, one of these subroutines is executed by a CALL statement in subroutine LOAD (Figure 17).

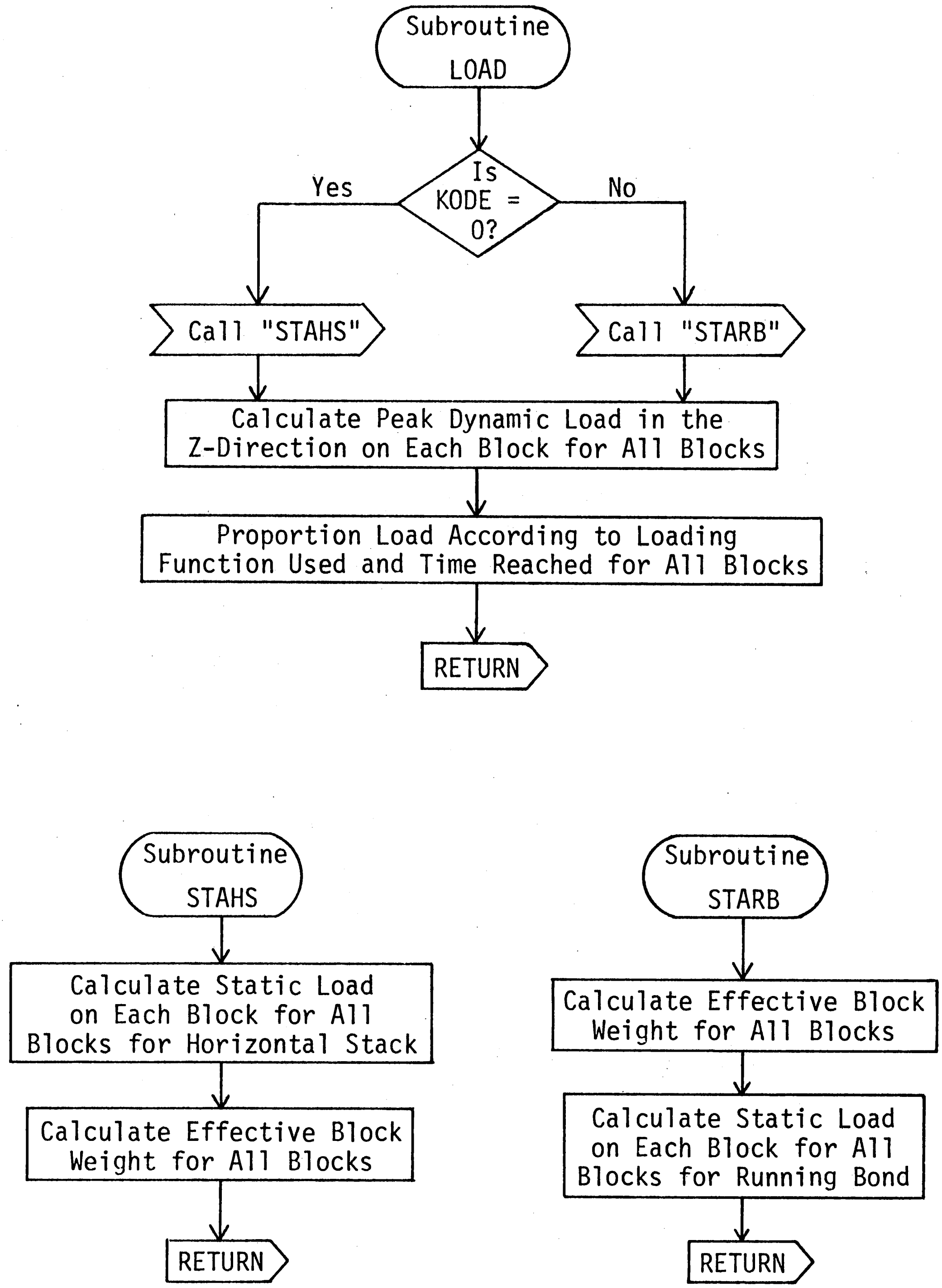


Figure 17. Summary Flow Diagrams for Subroutines
LOAD, STAHS, and STARB

Dynamic load calculations are performed by subroutine LOAD. The load on each block is calculated from a loading distribution over the entire wall that is sinusoidal, uniform, or a combination of both, as specified. The variation with time is calculated from one of the load-time functions shown in Figure 18. One of these functions needs to be specified in the problem data by specifying the appropriate ILOAD code. Subroutine LOAD is executed when called by subroutine SOLVER (Figure 22) at TIME = 0 and at every time interval thereafter.

5.2.5 Subroutine MASS

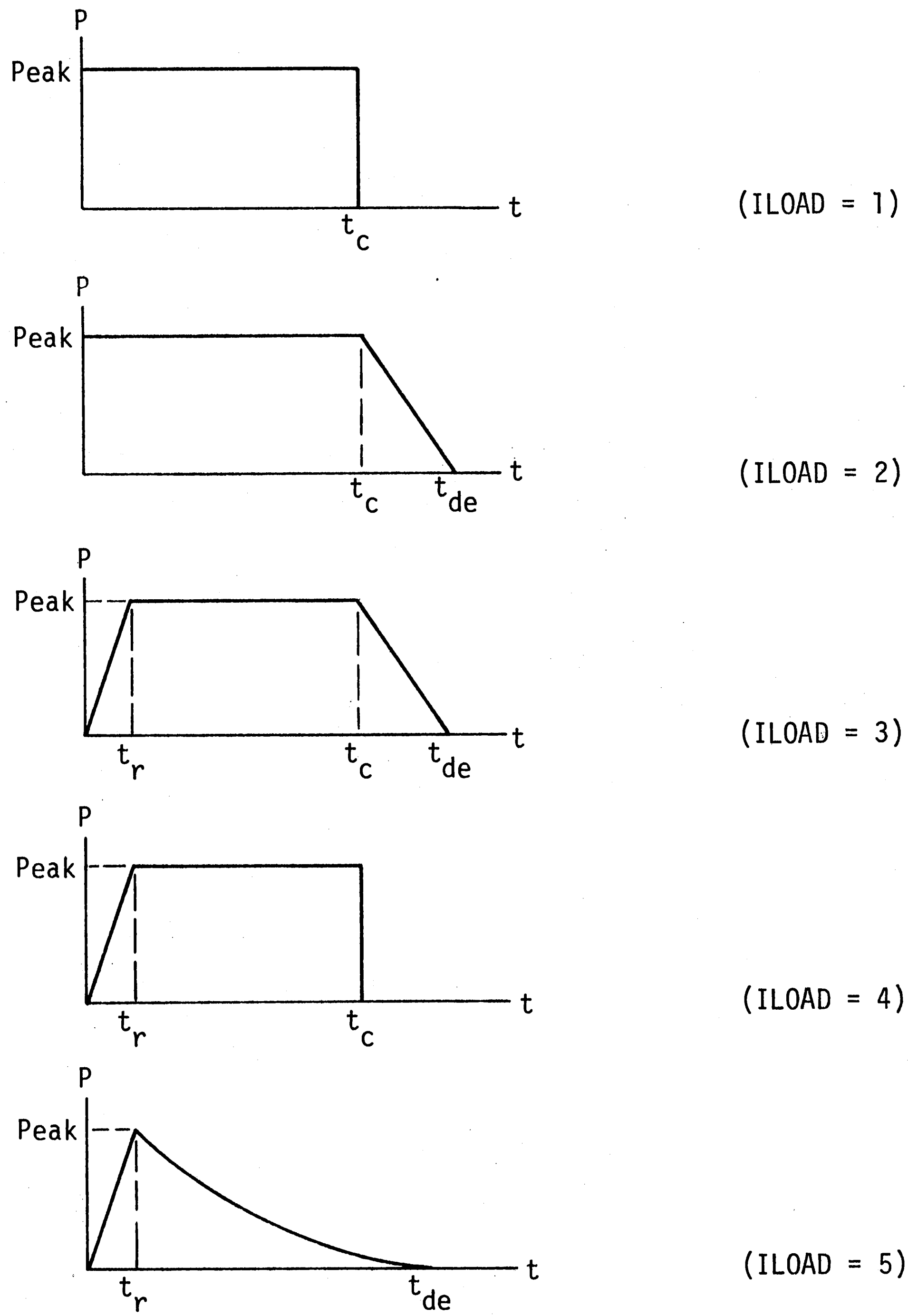
The mass and mass moment of inertia are calculated in this subroutine. Solid brick as well as two- and three-core concrete blocks are treated. Mass calculations were based on the effective block weight calculated in subroutines STAHS and STARB. This weight includes half the weight of each of the surrounding mortar joints in addition to the actual weight of the unit.

In calculating the mass moment of inertia for solid and hollow units, separate calculations were necessary, as shown in the flow diagram of Figure 19. Equation (2.2) was used for solid bricks; those used for the hollow concrete blocks are given in Appendix A.

This subroutine is executed once by a CALL statement in subroutine SOLVER (Figure 22).

5.2.6 Subroutine STIF

This subroutine had a dual purpose. The first was providing the proper modulus of elasticity for spring stiffnesses calculated in



Definition of terms for data input:

RTIME = Rise time = t_r

CTIME = Duration of constant pressure = t_c

DTIME = Decay time = t_{de}

Figure 18. Pressure-Time Functions Incorporated into the Program

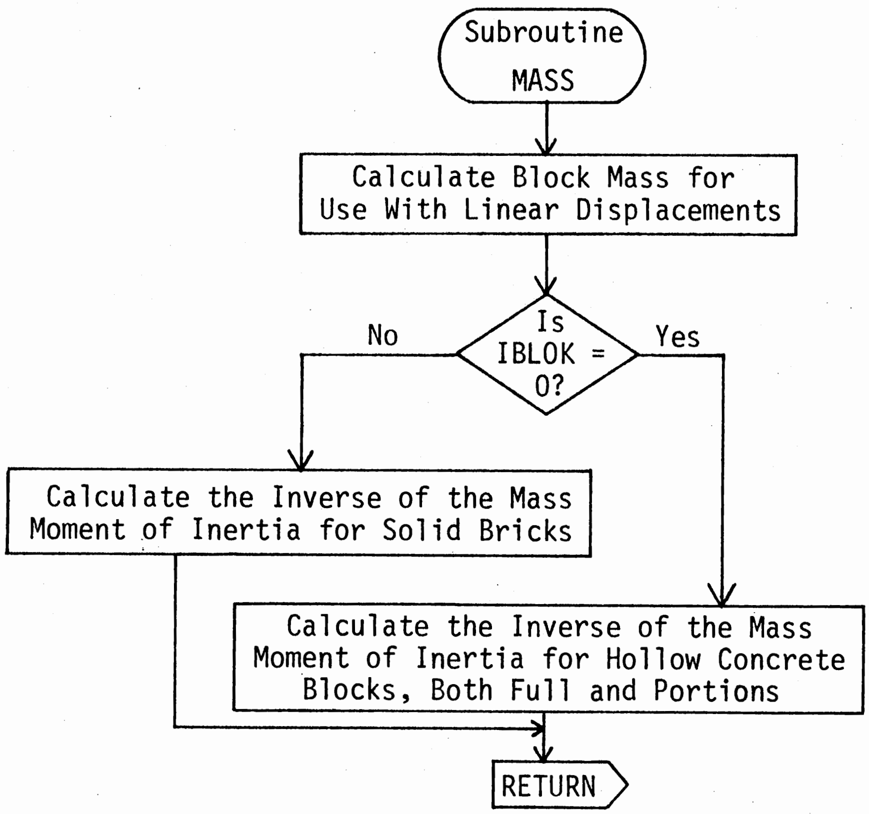


Figure 19. Summary Flow Diagram for Subroutine MASS

subroutine LINKEL. The second was constructing the various stiffness matrices of Chapter II. A summary flow diagram is shown in Figure 20.

Coding for the specification of the modulus of elasticity E occupies a very small area at the beginning of the subroutine, but performs an important function. Depending on a code stored in array NSTAGE for each spring in the wall, the proper E value is selected to multiply by the value obtained for spring stiffnesses in subroutine LINKEL. Originally, the code is set to one for all springs. It is then updated in subroutine SOLVER for the successive regions for springs in compression, or failure for those in shear or tension.

The overwhelming portion of the subroutine was reserved for constructing the stiffness matrices used in Equations (2.15e) and (2.18) and given in expanded form in Appendix B.

5.2.7 Subroutines FORCES and GEOMET

The force developed in each spring is calculated in subroutine FORCES, as shown in the flow diagram of Figure 21. First, the geometry matrix [G] used in Equation (2.8a,b) is assembled in subroutine GEOMET which is executed by a CALL statement. Knowing the centroidal displacements from subroutine SOLVER, the nodal displacements for each block are obtained using the geometry matrix. Change in the length of each spring is then calculated as the absolute difference between the nodal displacements of the two blocks at its ends. Finally, the force in each spring is calculated as the product of the stiffness and the change in length.

Subroutine FORCES is executed by CALL statements in subroutine SOLVER (Figure 22). The first CALL is for calculating the spring forces

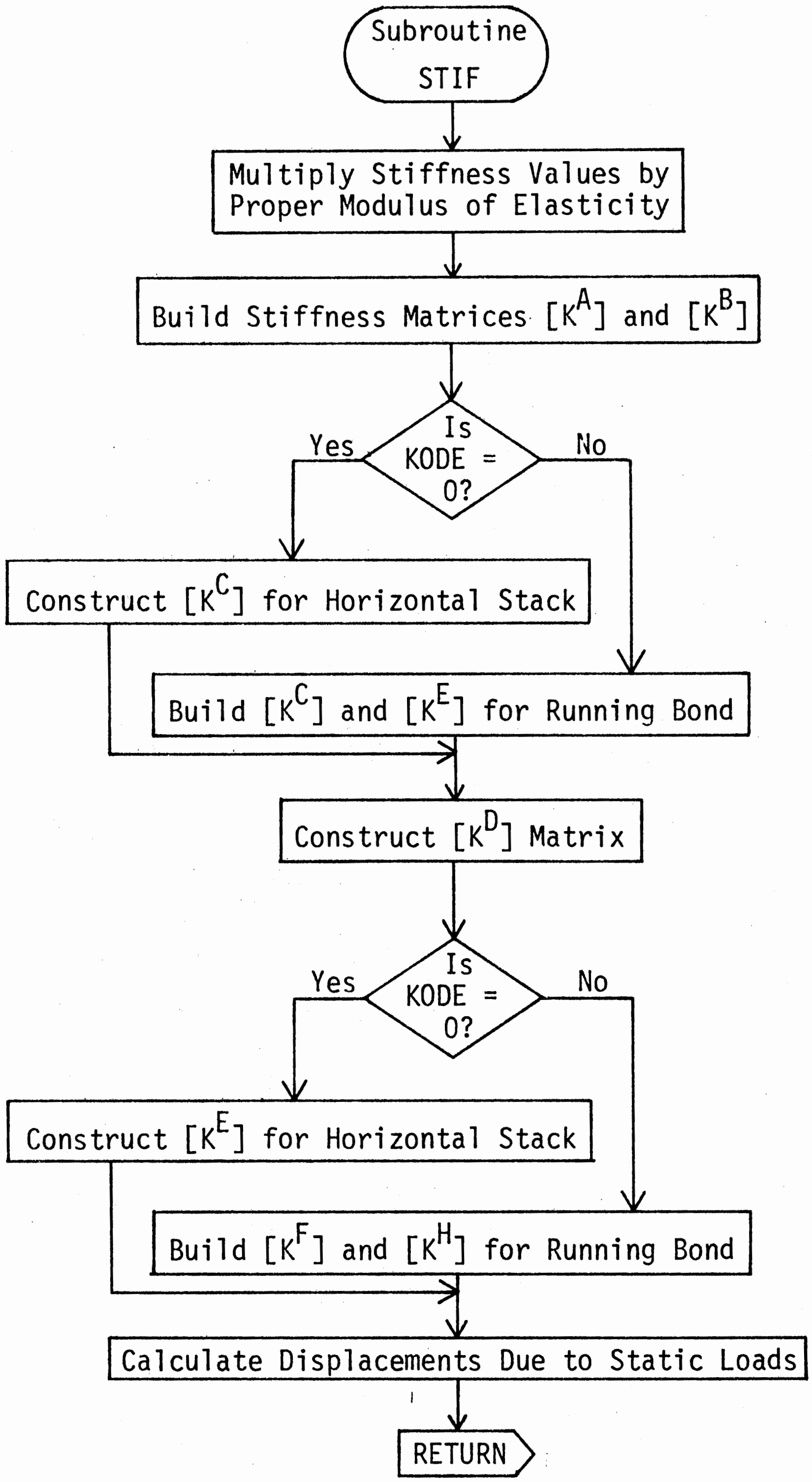


Figure 20. Summary Flow Diagram for Subroutine STIF

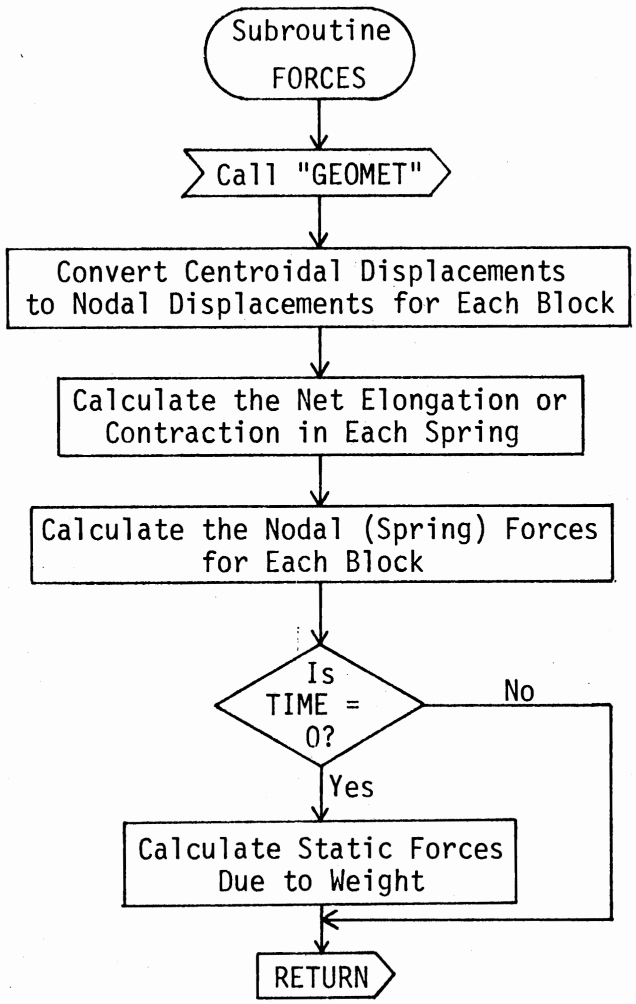


Figure 21. Summary Flow Diagram for Subroutine FORCES

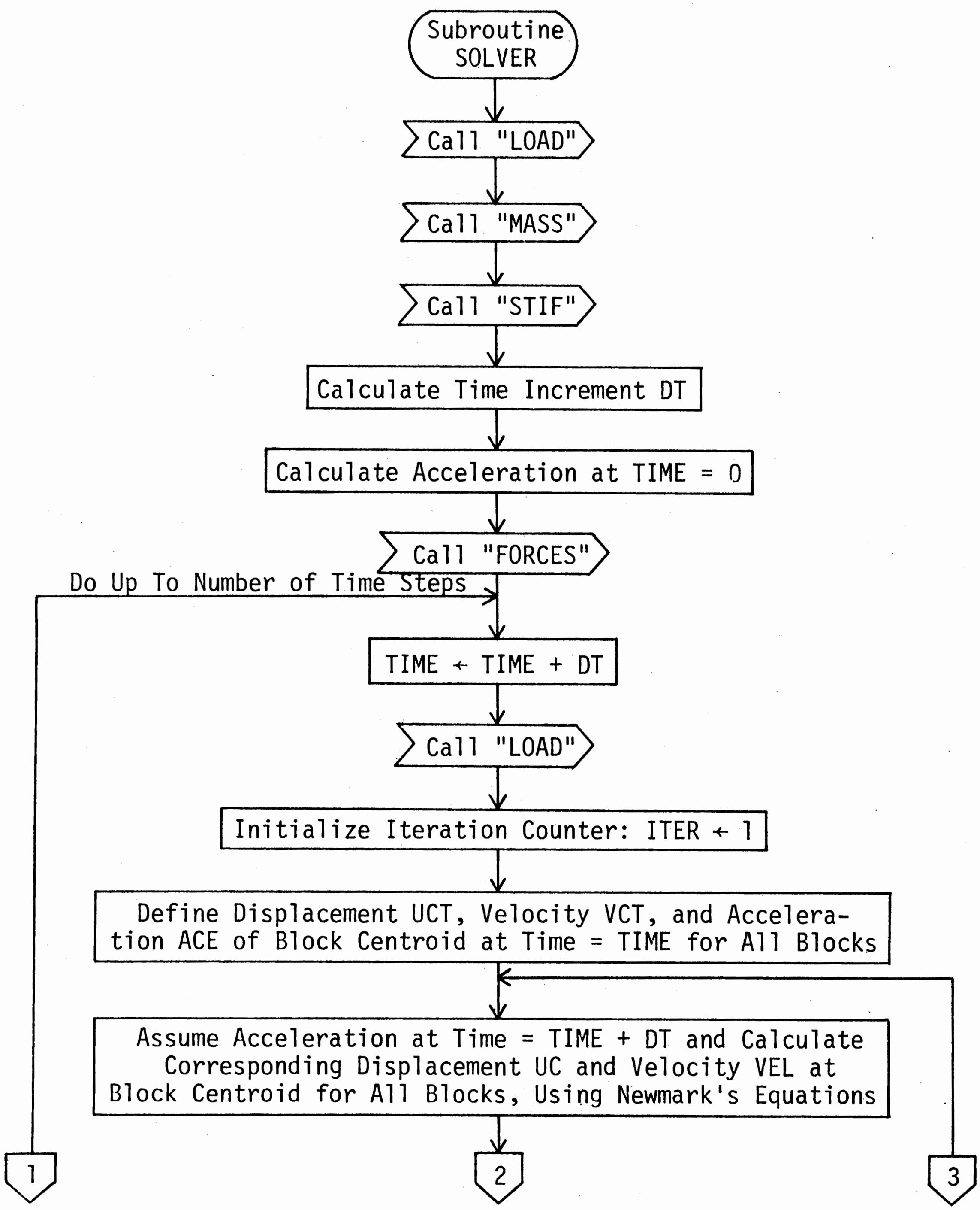


Figure 22. Summary Flow Diagram for Subroutine SOLVER

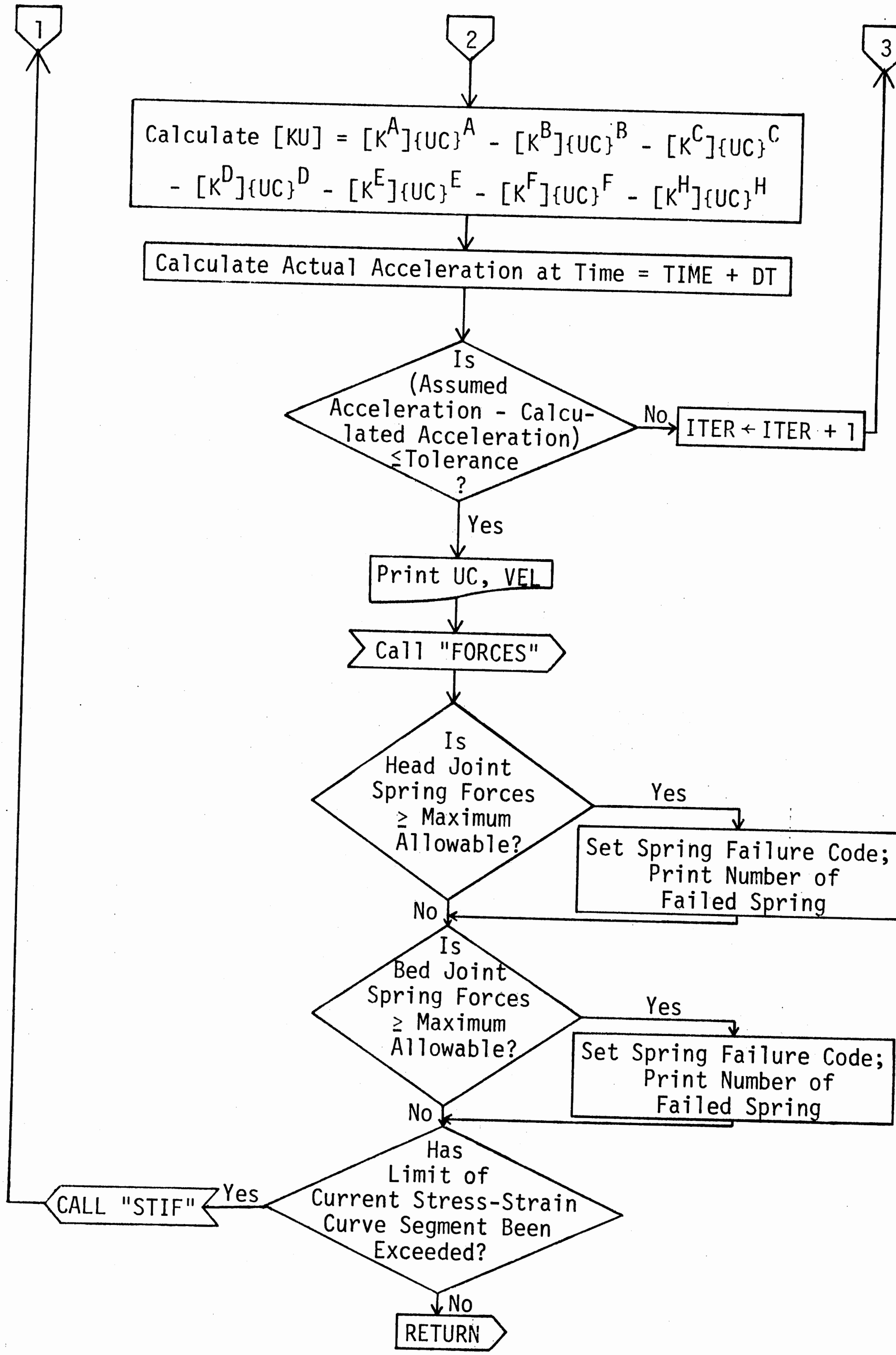


Figure 22. (Continued)

due to the static loads. In later CALLs, spring forces due to the dynamic loads are calculated.

5.2.8 Subroutine SOLVER

The main purpose of this subroutine was to calculate the centroidal displacements, velocities, and accelerations for each block. This is done by applying Newmark's equations given in section 2.4.1 and the equations of motion of the system (Equation (2.19b)). Much of the information needed in the process was calculated earlier. Thus, subroutines LOAD, MASS, and STIF are called by this subroutine.

In preparing the stiffness matrix for Equation (2.19b), the system stiffness matrix as given in Equation (2.19a) was not actually assembled into a global stiffness matrix. This is due to the fact that when assembled, the global stiffness matrix becomes banded. Storing the complete matrix in the computer, therefore, results in providing unnecessary storage space for the zero elements.

In order to avoid the extra storage space, the product of [K]{U} is substituted in Equation (2.19b) rather than the separate matrices. For example, the first element in the resulting vector is expressed as

$$\{KU\}_1 = [K^A]\{U_1\} - [K^B]\{U_2\} - [K^C]\{U_5\} \qquad (5.1)$$

and the corresponding expression for acceleration is

$$\{\ddot{U}_1\} = [M]^{-1} (\{F_1(t)\} - \{KU\}_1) \qquad (5.2)$$

For the purpose of detecting any cracks due to mortar or bond failure, the force in each spring is calculated by calling subroutine FORCES when new displacements are obtained after each time interval. Whenever the force in a given spring reaches the maximum allowable limit (calculated earlier in subroutine LINKEL), a failure code is assigned to that spring in array NSTAGE so that the stiffness is set to zero in subroutine STIF. For springs in compression, a similar approach is followed to assign a modulus of elasticity corresponding to the level of stress at the particular stage.

CHAPTER VI

SELECTED PROBLEMS

6.1 General

The principles discussed in the previous chapter will be employed here to analyze a number of beam and wall systems. The process should serve as a means of testing the performance of the computer program.

The behavior of the systems analyzed here will be discussed in Chapter VII.

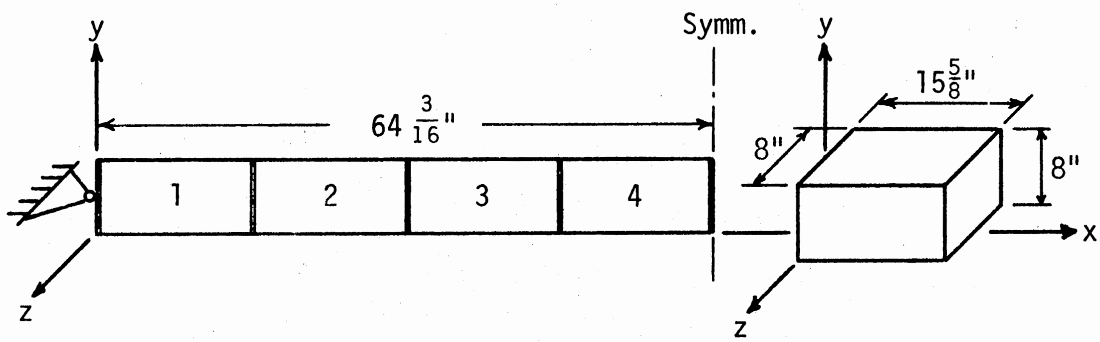
6.2 Simply Supported Beams

Two versions of a simply supported beam were analyzed. The first, shown in Figure 23(a), was made of eight clay bricks. The second version was the same as the first, except for substituting concrete blocks for the clay bricks, as shown in Figure 23(b). Due to symmetry, only the left half of each beam was analyzed.

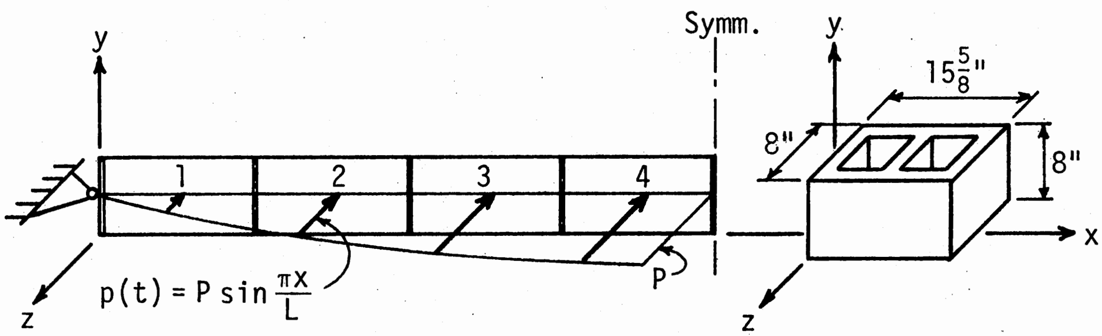
Both beams were subjected to a sinusoidal loading distribution. The pressure variation with time was according to the function shown in Figure 23(c).

Computer printouts of the input data and calculated stiffness used in both beams are presented in Appendix E.

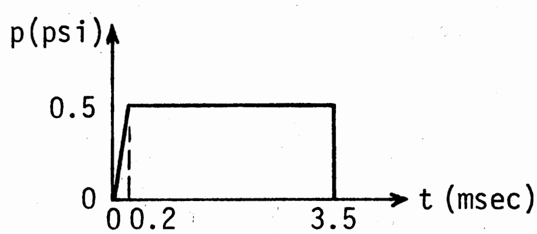
In order to compare the behavior of both beams, the centroidal displacement versus time of unit number 4 in each beam was plotted in Figure



(a) Eight-Unit Beam of Clay Bricks



(b) Eight-Unit Beam of Concrete Blocks With Typical Loading



(c) Pressure-Time History

Figure 23. Simply Supported Beams

24. A comparison of the behavior of the clay brick beam with that of a closed form, continuous elastic beam is presented in section 7.3.

6.3 Simply Supported Walls: Horizontal Stack Pattern

In order to investigate the behavior of simply supported walls, the system shown in Figure 25(a) was analyzed. Due to symmetry, only a quarter of the wall was analyzed. Again, two types of masonry units, namely, clay bricks (Figure 25(b)) and concrete blocks (Figure 25(c)), were used.

A sinusoidal loading distribution was applied to the wall. The pressure-time history is shown in Figure 25(d). Computer printouts of the input data and calculated stiffnesses for this problem are presented in Appendix E. The response of unit number 12 is shown in Figure 26 for both types of masonry units.

6.4 Simply Supported Walls: Running Bond Pattern

The simply supported wall with a horizontal stack pattern of the previous section is analyzed here as having a running bond pattern. The wall system, brick, and loading used are illustrated in Figure 27. Computer printouts of the input data and spring stiffnesses are presented in Appendix E.

It is interesting to compare the behavior of the two systems. Computer printouts of the results at 7.5 milliseconds are presented in Appendix E for both the horizontal stack and the running bond patterns. A comparison of the fourth row deflection in each system is shown in Figure 28. The displacement history of block number 29 in the running

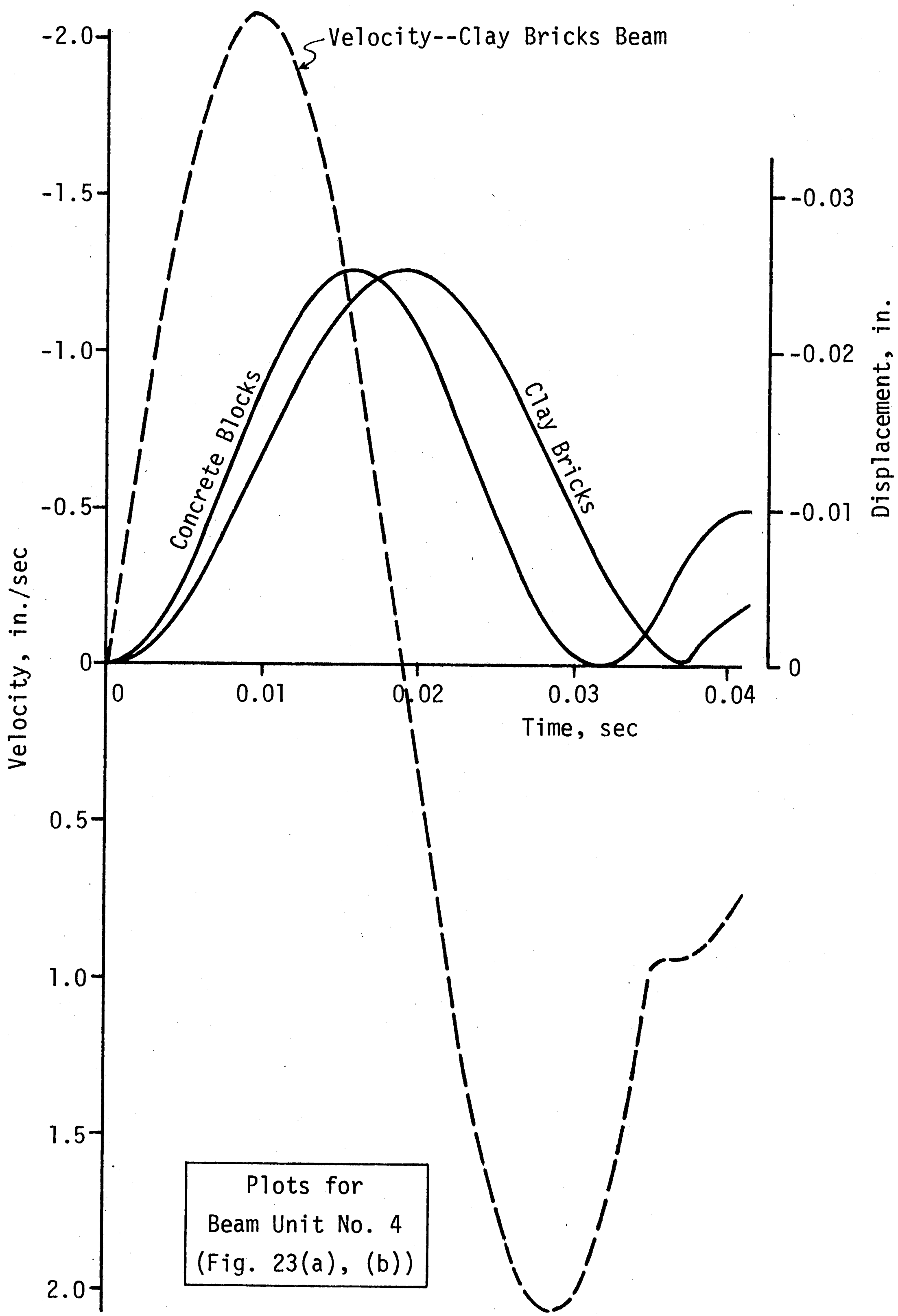


Figure 24. Displacement and Velocity in the z-Direction as a Function of Time for a Typical Beam Unit

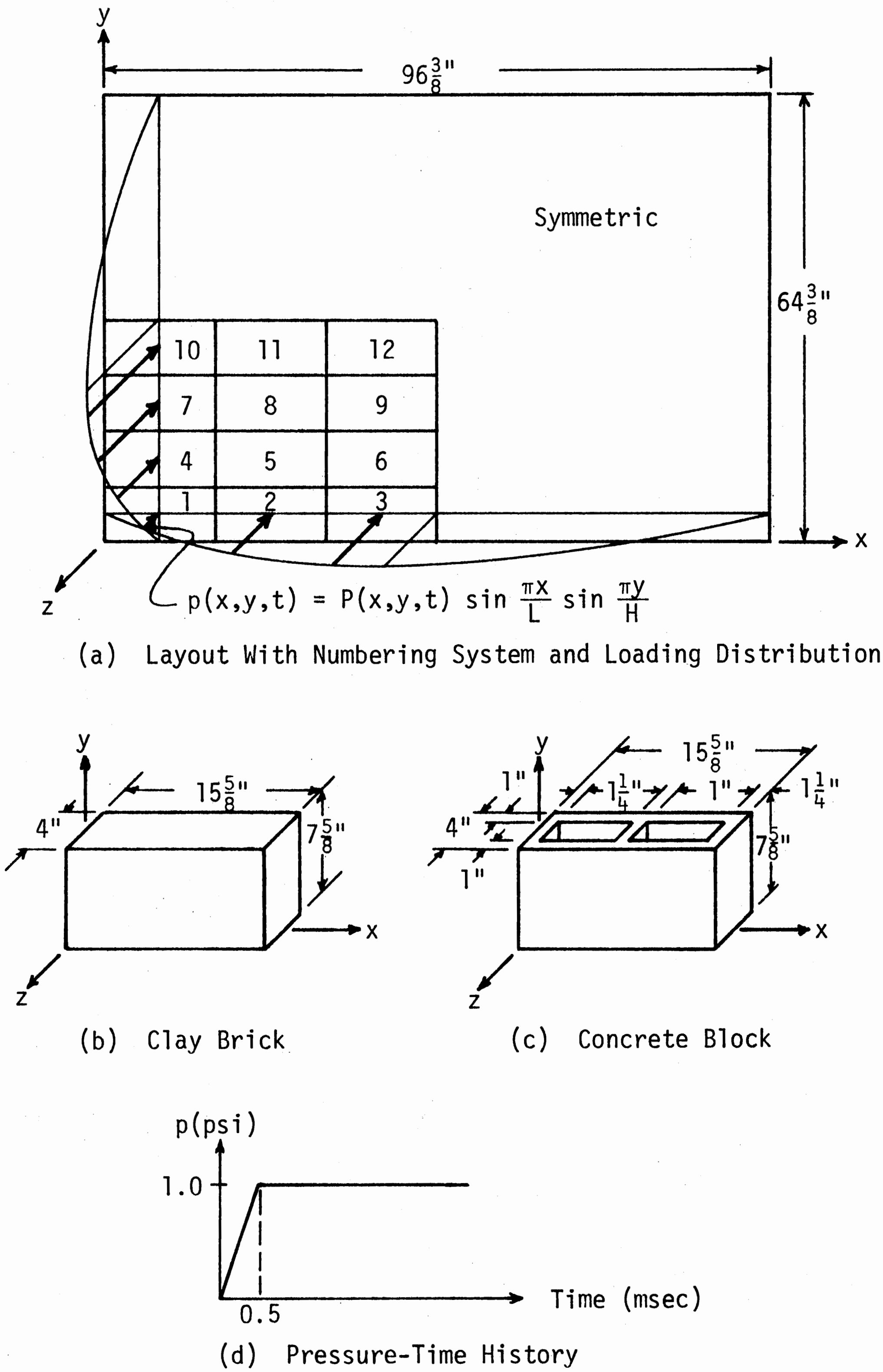


Figure 25. Simply Supported Wall With a Horizontal Stack Pattern

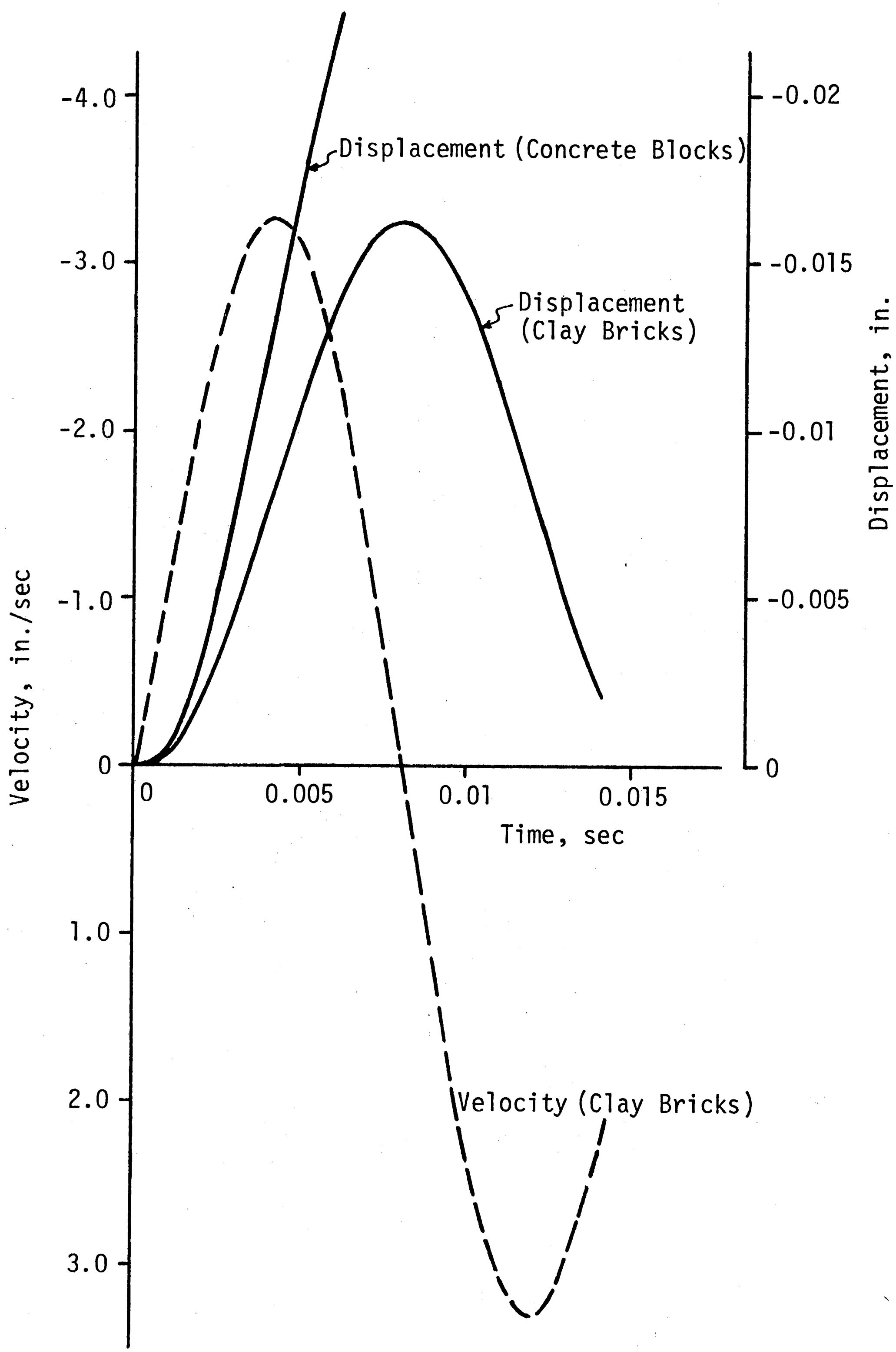
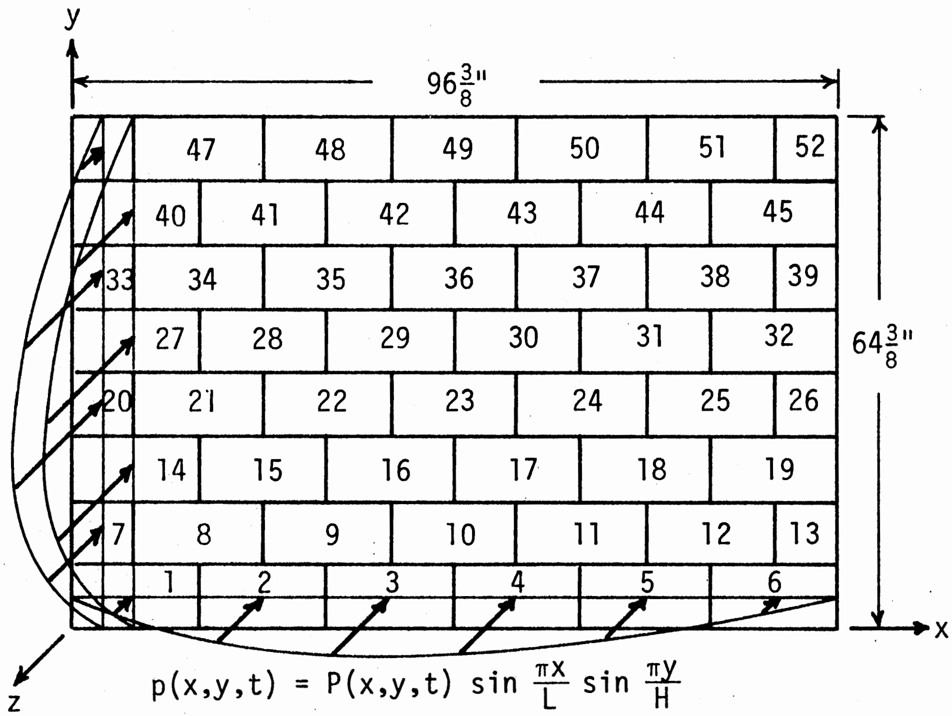
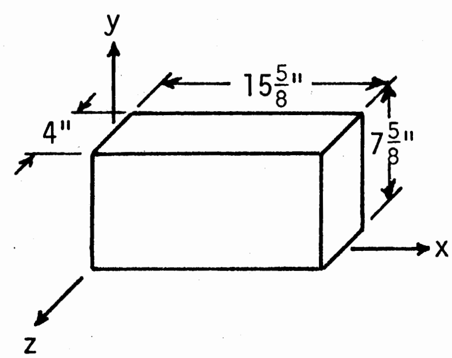


Figure 26. Displacement and Velocity in the z-direction as a Function of Time for a Typical Unit in the Horizontal Stack Wall

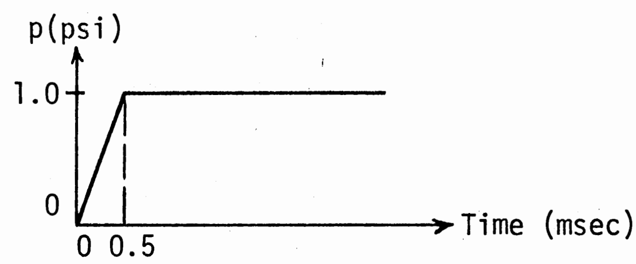


$$p(x,y,t) = P(x,y,t) \sin \frac{\pi x}{L} \sin \frac{\pi y}{H}$$

(a) Layout With Numbering System and Loading Distribution



(b) Clay Brick



(c) Pressure-Time History

Figure 27. Simply Supported Wall With a Running Bond Pattern

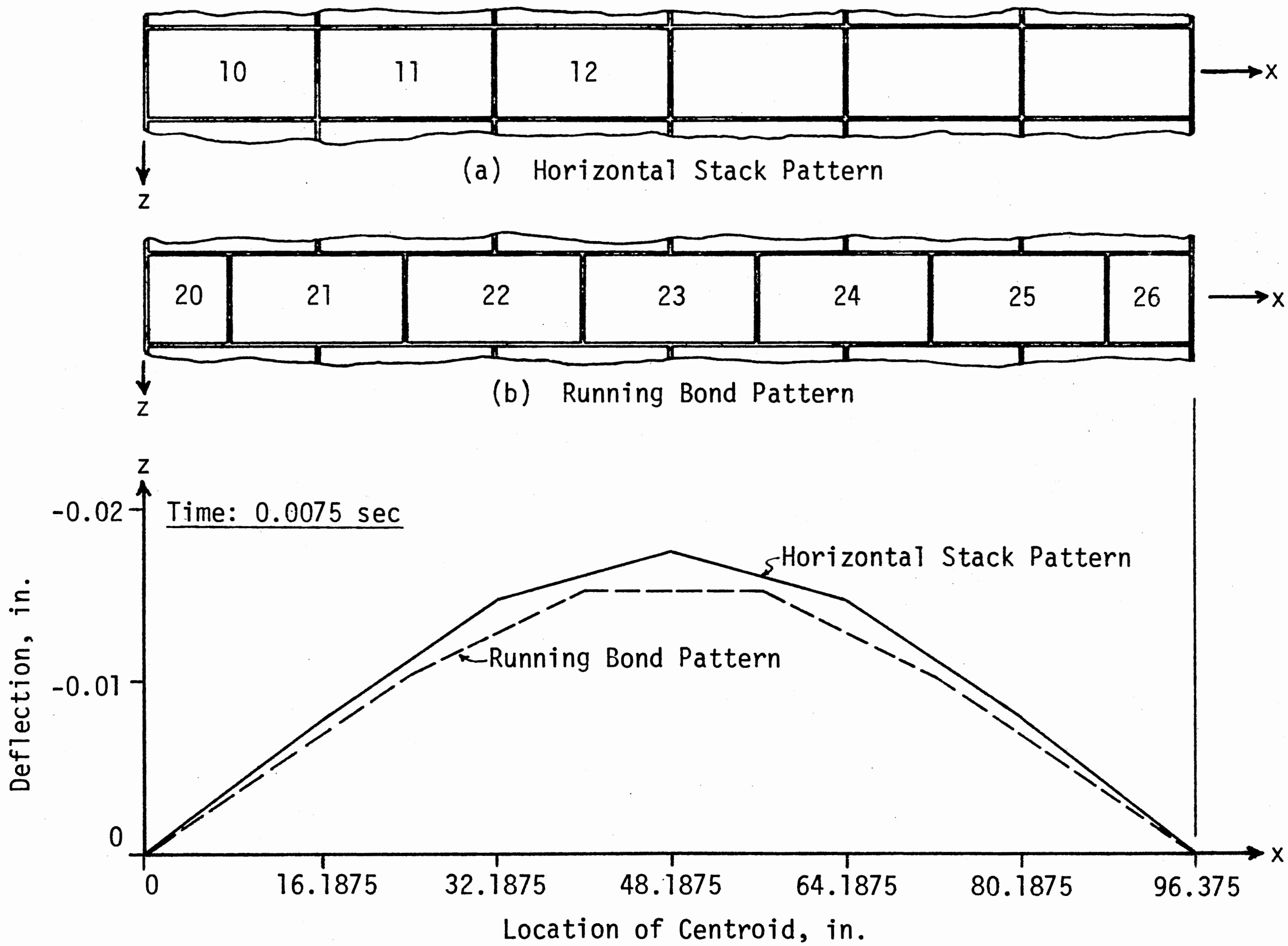


Figure 28. Comparison of Deflection Along a Row of Horizontal Stack and Running Bond Patterns

bond pattern and the corresponding brick (number 12) in the horizontal stack pattern are presented in Figure 29. Clearly, the row deflection and unit displacement differ slightly in magnitude. It is also apparent from Figure 28 that the row deflection of the running bond pattern follows a sine curve more closely than the corresponding row in the horizontal stack pattern at the given instant which is near the time of maximum deflection.

6.5 Comparison With Experimental Studies

Much of the experimental work on the dynamic behavior of masonry walls has been performed by the URS Research Corporation. One of the experimental walls that has been used in the literature for comparison purposes is a wall having a "common Flemish bond" construction with a bond course every sixth row. The wall is simply supported at the top and bottom and free on the sides; it is approximately 12 feet long, 8½ feet high, and 8 inches thick.

The wall was tested by Willoughby (39) at the URS Research Corporation with a uniformly distributed load having the load parameters given in Figure 30. The deflection at the center of the two panels tested are shown in Figure 31 by curves EA and EB.

In at least two cases this wall was used as an example. Wiehle (36) reduced the wall into a single-degree-of-freedom system and applied Newmark's β method in the analysis. The resistance function used was based on a beam strip of the wall. The response obtained is represented by curve W in Figure 31. On another occasion Summers analyzed a 12-inch wide strip of the wall as a beam with the "finite element grillage" representation. Curve S in Figure 31 was obtained by Summers' method.

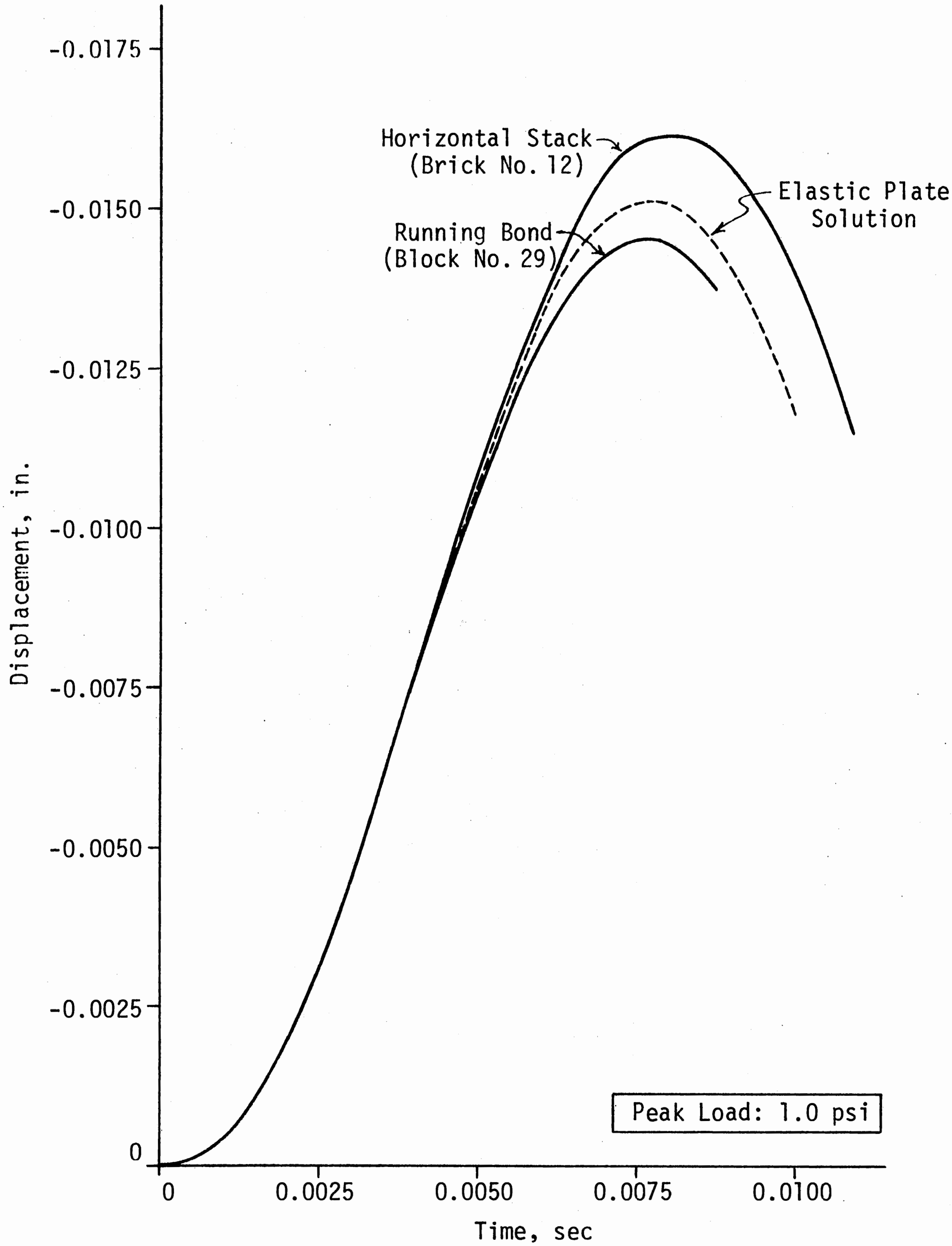
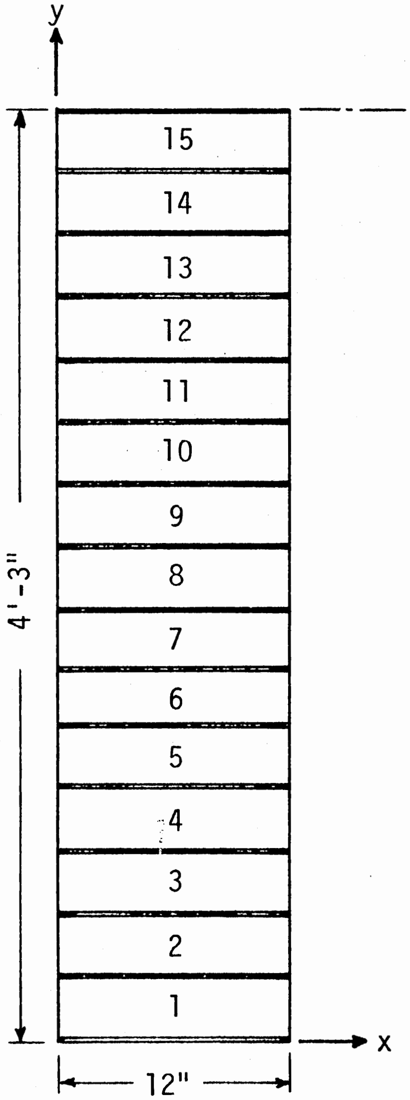
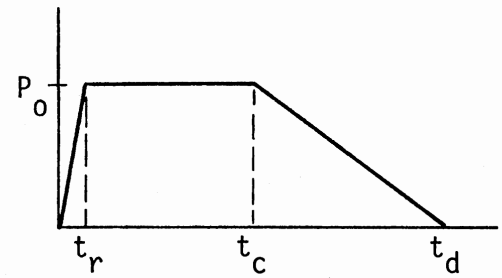


Figure 29. A Comparison of the Displacements of a Typical Unit in the Horizontal Stack and Running Bond Patterns





Load Parameters

P_o = 3 psi
t_r = 2 msec
t_c = 40 msec
t_d = 80 msec

Beam Parameters (Simply Supported)

Length = 8 ft, 6 in.
Width = 12 in.
Depth = 8 in.
Height of brick = 3 in.
Joint thickness = 3/8 in.

Mortar Properties

Tensile bond strength = 135 psi
Shear bond strength = 128 psi
Compressive strength = 2656 psi
Unit weight = 110 pcf
Modulus of Elasticity = 1.3×10^6 psi

Figure 30. Beam Model for Comparison With Experimental Brick Wall Panel

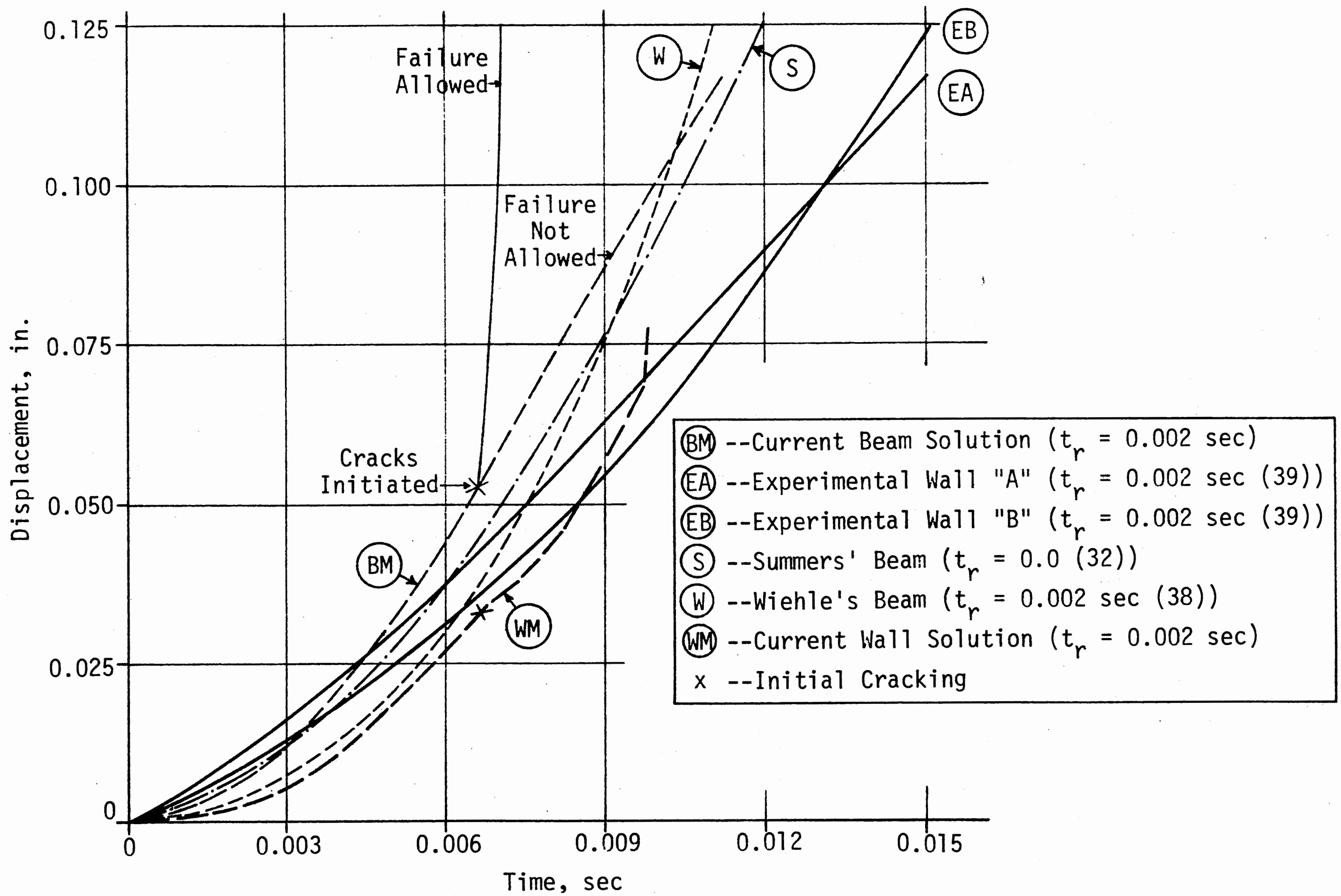


Figure 31. Comparison of Experimental and Various Theoretical Displacements Versus Time for Simply Supported Brick Wall Panel

In order to compare the response of the wall using the principles employed in this research with the previous results, a 12-inch wide beam was analyzed. The beam model, load parameters, beam parameters, and mortar properties are given in Figure 30. The response of unit number 15 is given by curve BM in Figure 31. Note that this curve represents the displacement at the center of unit number 15 rather than at the center of the beam. As the curve illustrates, the beam disintegrated rapidly following the development of tensile cracks at the top and bottom of unit 15. Had the bond between the mortar and the bricks been strong enough so that no failure occurs at the stress level attained, the beam deflection would have followed the general trend of the deflection curves obtained by Wiehle and Summers. Note that the forcing function used by Summers had no rise time.

The complete 12 x 8½ foot wall was also analyzed. As indicated earlier, this wall is simply supported at the top and bottom and is free on the sides. Load parameters and mortar properties were the same as those given in Figure 30. The bricks used were 19 inches long, 6.5 inches high, and 8 inches deep. As such, the bricks have approximately twice the dimensions of the bricks used in the experimental walls by Willoughby (39). The wall pattern used was version II of the running bond pattern (Figure 12(d)). The average displacement of seven bricks located near the center of the wall is represented in Figure 31 by curve WM.

When this curve is compared with experimental curves EA and EB, certain factors must be kept in mind. The current model employs a running bond pattern rather than the flemish bond, with bricks approximately twice as large as those used in the experimental walls. In addition, joint thicknesses and the modulus of elasticity of mortar were estimated on the basis of information in the URS report (39). A more important factor, however, is the fact that the experimental walls were built with bricks that are approximately 4 inches deep. Thus, the 8-inch thick experimental wall can be thought of as two 4-inch thick walls joined together by a mortar panel extending over the full length and height of the wall. Certainly, the stiffening effect of this panel cannot be ignored.

The responses shown in Figure 31 clearly indicate that the beam models follow the same general trend. Collectively, they seem to predict the wall response fairly well.

CHAPTER VII

COMPARATIVE STUDY AND DISCUSSION

7.1 General

In the previous chapter, different systems of beams and walls were treated. In addition, different masonry units were also used. The effect of these and other factors on the behavior of masonry walls will be discussed in this chapter.

7.2 Effect of the Type of Masonry Units

The two types of masonry units used, namely, clay bricks and two-core concrete blocks, produced different system behaviors. For the beam of Figure 23(a), substituting the 32 percent lighter concrete blocks for the clay bricks (Figure 23(c)) produced the same maximum displacement as shown in Figure 24. It did, however, raise the natural frequency of the beam by about 26 percent. This was not the case for the wall of Figure 25. Substituting concrete blocks for the bricks here resulted in higher displacement, as indicated in Figure 26.

In the case of the beam, the change in natural frequency is traced directly to the change in the unit's mass. This is also a contributor to the higher displacement of the wall. A more important factor for the wall, however, is the lower stiffness of the bed joint springs caused by the mortar covering only the face shells of the concrete blocks.

7.3 Comparison With Closed Form Solutions

In order to compare the response of the modeled masonry systems with equivalent elastic systems, the beam of Figure 23(a) and the walls of Figures 25(a) and 27(a) will be considered.

The displacement versus time of beam unit number 4 (shown in Figure 24) was reproduced in Figure 32. The dashed curve in Figure 32 represents the corresponding displacement obtained for an equivalent elastic beam. Clearly, the two solutions are very close, with a peak difference in the neighborhood of 1.85 percent.

For discussion of the simply supported walls, reference is made to Figure 29. The horizontal stack pattern resulted in a peak displacement which is over 7 percent higher than that of an equivalent elastic plate. Conversely, the peak displacement in the running bond pattern was about 5 percent lower than the elastic plate solution.

Inspection of these results shows that the response of the modeled masonry beam and walls is essentially the same as that of a comparable elastic beam and plate. Furthermore, the running bond pattern seems to develop a somewhat higher resistance than the horizontal stack pattern for dynamically loaded walls with simple supports on all sides.

7.4 Wall System

One of the objectives of this study was to compare the behavior of the horizontal stack and the running bond systems. In Chapter VI a wall was analyzed once as having a horizontal stack pattern and again with a running bond pattern, both under the same loading. Figures 28 and 29 show the deflection and block displacement for both types. The maximum

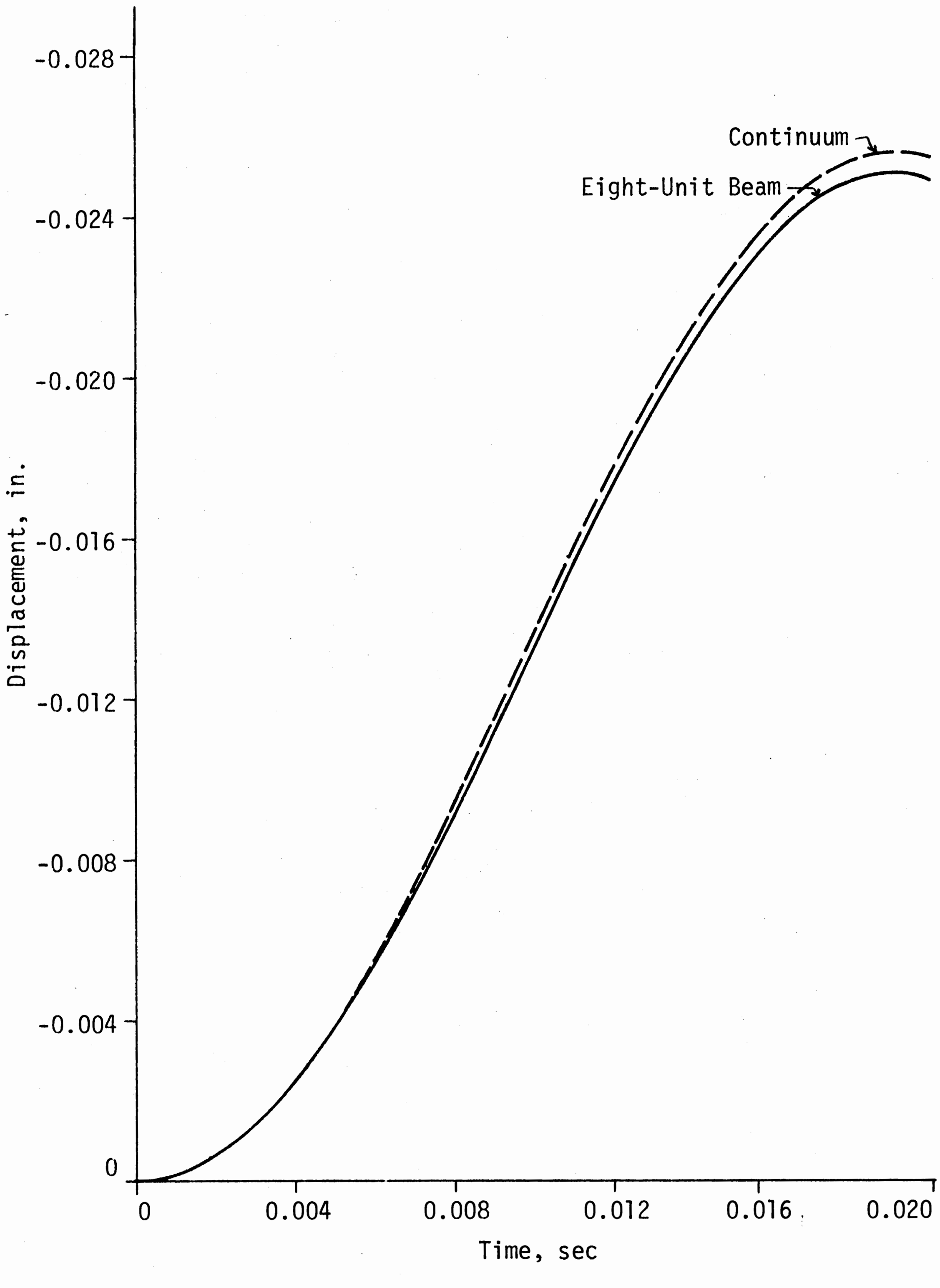


Figure 32. Comparison of Beam Deflection With the Closed Form Solution

deflection for the running bond pattern is approximately 12 percent lower than that of the horizontal stack pattern, which indicates that the running bond pattern has a slightly higher transverse resistance than the horizontal stack pattern.

7.5 Crack Development

Crack development is one of the important factors that affects the strength of masonry walls. A number of investigators have pointed to the bond failure between the masonry units and the mortar as the primary cause for crack development.

In order to study the development of cracks, the horizontal stack wall of Figure 25 was subjected to the same loading shown in Figure 25, but with peak pressures of 1, 2, 4, and 6 psi. Figure 34 shows the displacement of block number 12 versus time for all four peak pressures. No cracks developed when the 1 psi peak pressure was used. Under the other three peak pressures, tensile bond failure occurred at the upper bed joint of block number 12 (mid-height of the wall) when the block displacement reached approximately 0.027 inches. It is interesting to note that had there been no crack development, the 2, 4, and 6 psi curves would have followed a pattern similar to that of the 1 psi curve but with a higher amplitude. This is shown for the 2 psi pressure in Figure 33.

A clearer picture of crack development under the different peak pressures might be obtained from Figures 34, 35, and 36 which show the crack development after seven milliseconds of exposure to the peak pressures of 2, 4, and 6 psi, respectively. In Figure 34, virtually all the central area of the wall developed cracks, with the rear face having a larger share than the front face. Cracking was initiated by tensile

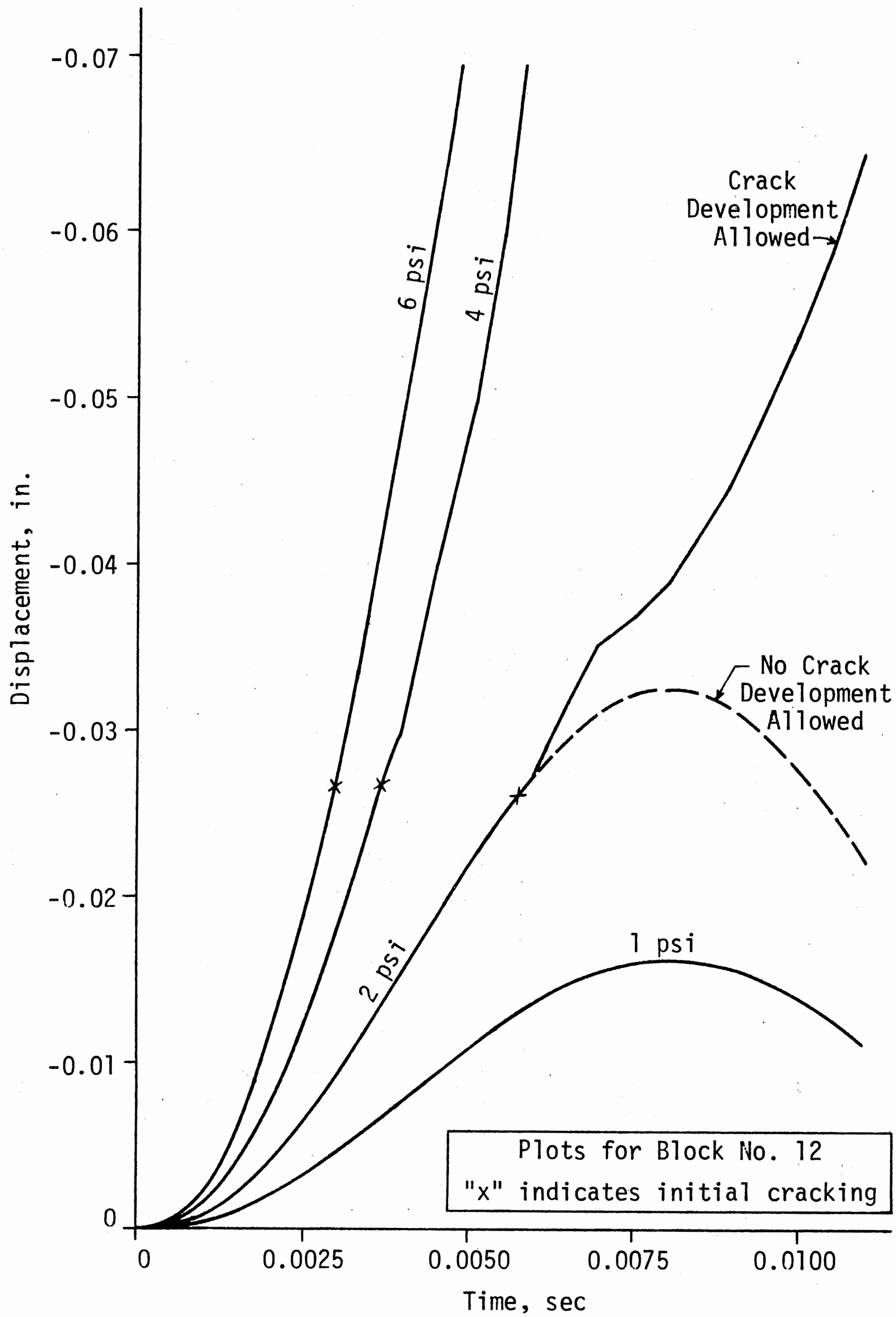
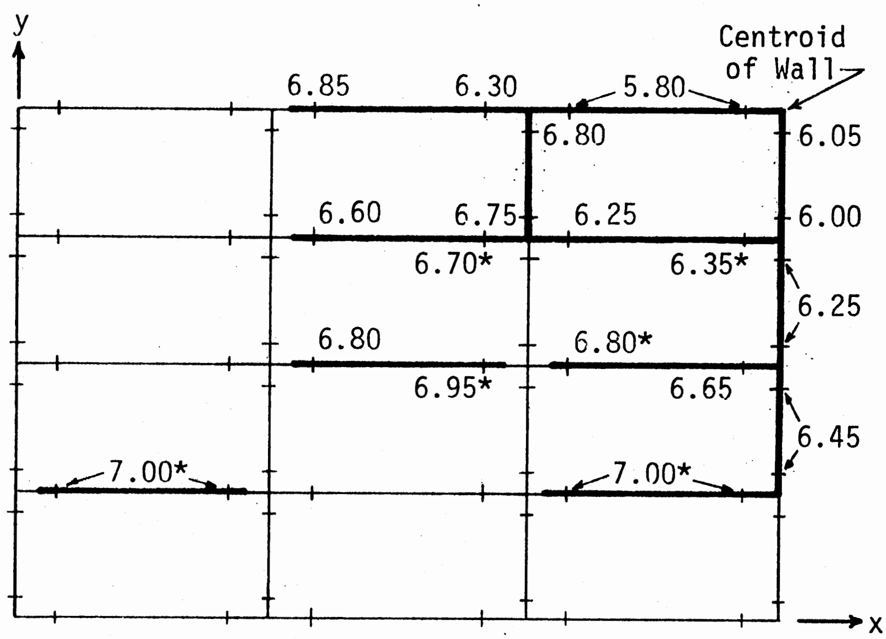


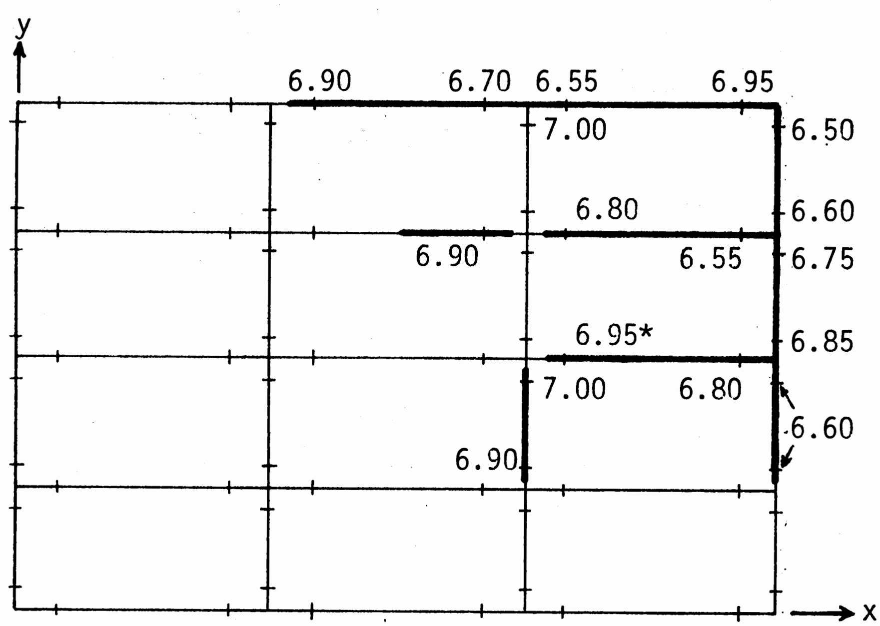
Figure 33. Displacement in the z-Direction as a Function of Time for a Typical Block for Various Load Peaks



(a) Rear Face

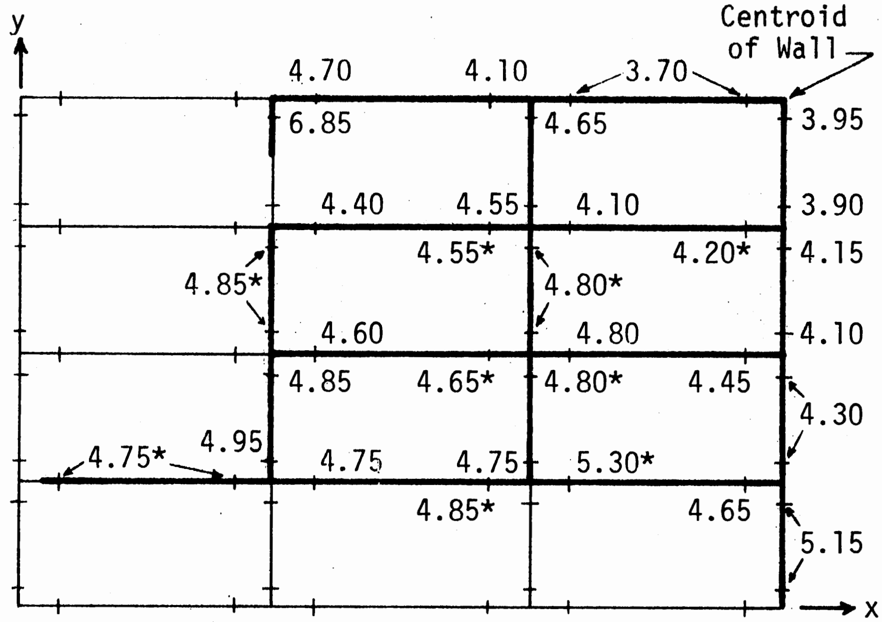
(Numbers indicate time of tensile bond failure)

*Time of shear bond failure



(b) Front Face

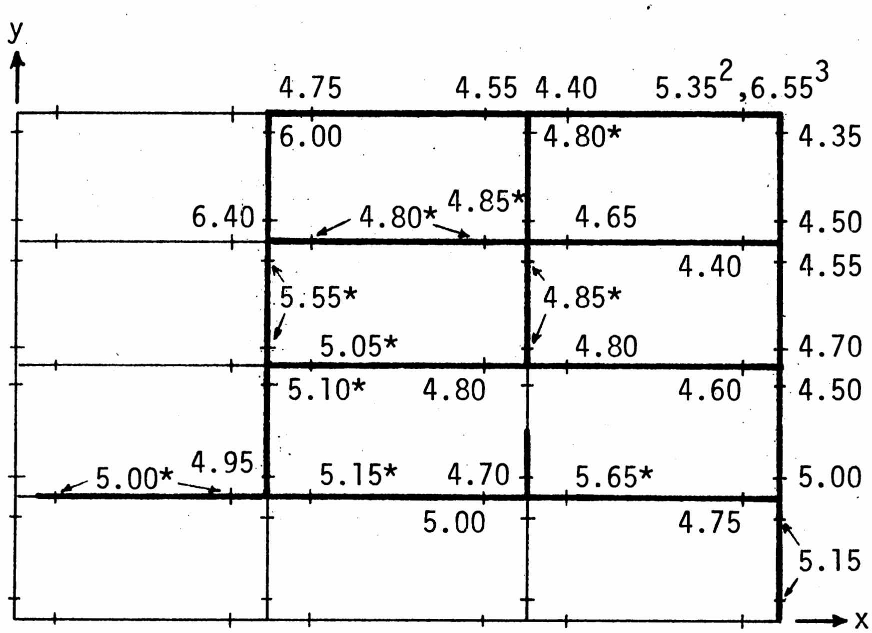
Figure 34. Crack Development in a Horizontal Stack Simply Supported Wall Subjected to a Peak Pressure of 2 psi for 7 msec.



(a) Rear Face

(Numbers indicate time of tensile bond failure)

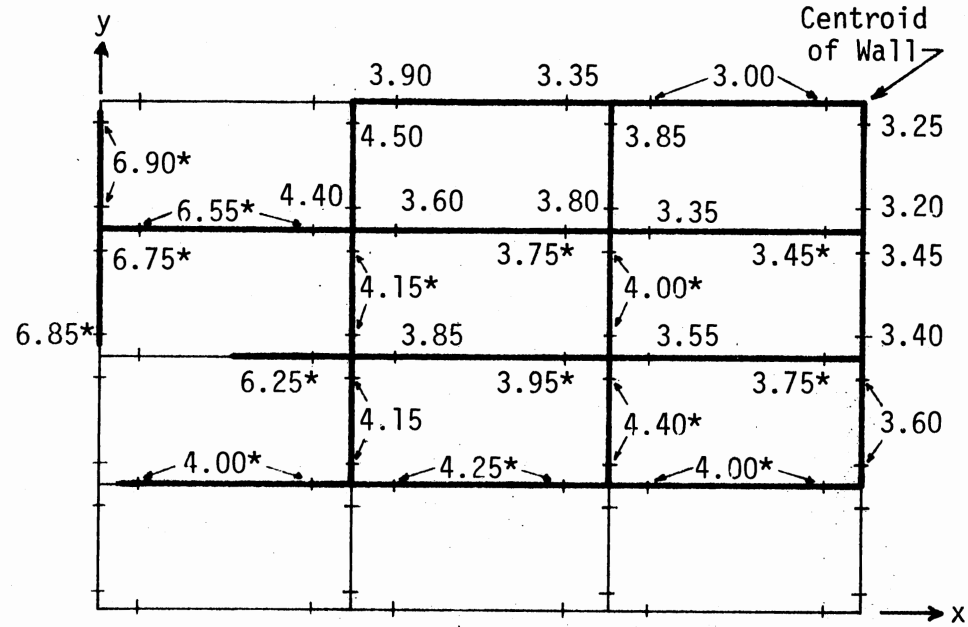
*Time of shear bond failure



(b) Front Face

Superscript numerals indicate segment number in the stress-strain curve for mortar to which a shift occurred at time shown

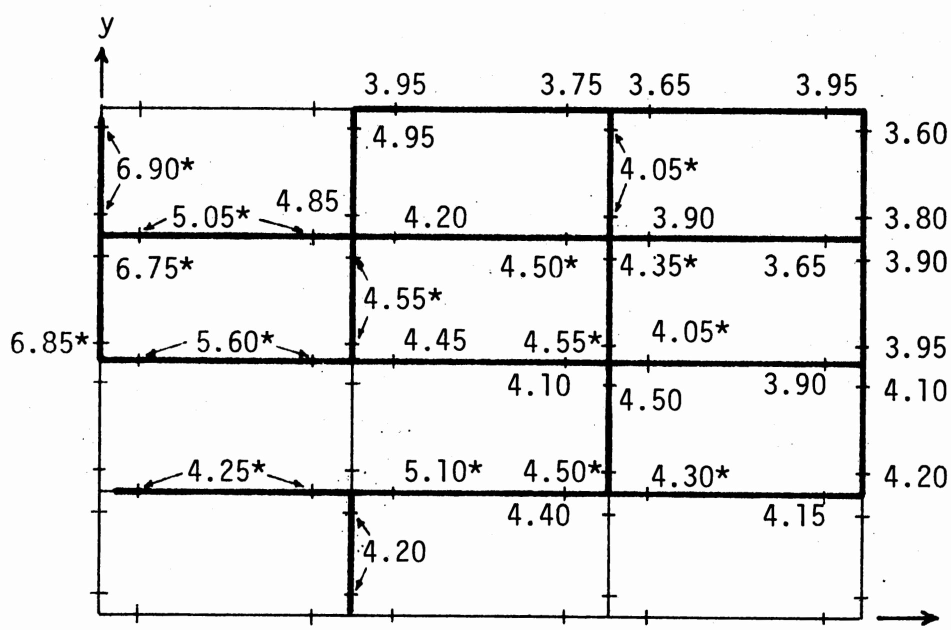
Figure 35. Crack Development in a Horizontal Stack Simply Supported Wall Subjected to a Peak Pressure of 4 psi for 7 msec



(a) Rear Face

(Numbers indicate time of tensile bond failure)

*Time of shear bond failure



(b) Front Face

Figure 36. Crack Development in a Horizontal Stack Simply Supported Wall Subjected to a Peak Pressure of 6 psi for 7 msec

failure near the center of the rear face. It was followed by successive failures at the other joints, starting with those along the symmetry lines. More extensive cracking developed when the wall was subjected to higher pressures, as illustrated in Figures 35 and 36. For the most part, cracking was due to tensile bond failure, although shear bond failures did occur. In addition, some joints experienced compressive stresses in the nonlinear region of the stress-strain curve, as indicated on the front face of the wall in Figure 35.

CHAPTER VIII

SUMMARY AND CONCLUSIONS

8.1 Summary

Research on the response of structures to blast loading has been underway for decades. Much of that work, however, was concentrated on the frame behavior. Since wall resistance can be significant, more attention has been given to wall behavior in recent years. The aim of this study was to develop a model for masonry walls that simulates their response to blast loading.

Previous investigations have pointed to the mortar joint as the weak link in masonry walls. Thus, the important role of the mortar joint was emphasized in the model developed. While masonry units were assumed to be rigid, mortar deformation was simulated by flexible three-dimensional "linkage" elements. Enclosed in each of these elements are three perpendicular springs to measure the axial, in-plane shear, and transverse shear stresses. Since the masonry units were assumed to be rigid, forces and deformations developed in the springs were conveniently related to the center of gravity of the masonry units, using the small displacement theory.

The horizontal stack and running bond wall patterns were studied; equations of motion were developed for each. Walls built with either clay bricks or concrete blocks were used in the study.

A computer program was written for solving the equations of motion, using Newmark's β method. The program can be used to analyze four versions of the running bond pattern, in addition to the horizontal stack pattern, with different boundary conditions. Furthermore, solid brick as well as two- and three-core concrete blocks are acceptable.

So far as the loading distribution is concerned, uniform, sinusoidal, or a combination of both may be specified. Pressure-time relationships can be any of four common pulses, in addition to the blast function.

The level of stress in each spring is tested in the program after each time interval. Thus, crack development and transition into the nonlinear region of the stress-strain curve are traced.

8.2 Conclusions

A number of conclusions may be drawn from this study. They are:

1. Based on the excellent agreement between the response obtained from the model and those from previous research and closed form solution, the model chosen seems to simulate the behavior of masonry walls quite well.

2. Assuming that the masonry units are rigid is a valid assumption, provided that proper stiffnesses are assigned to the springs.

3. Using the small displacement theory in relating nodal and centroidal parameters for the masonry units proved to be satisfactory.

4. The running bond pattern developed a somewhat higher resistance to the transverse dynamic load than the horizontal stack pattern for simply supported walls.

5. Walls constructed with clay bricks have a higher resistance to transverse dynamic loads than those constructed with concrete blocks.

6. The model facilitated the prediction of crack development and simulated wall response beyond the elastic limit.

7. Mortar properties and block dimensions have a great effect on the resistance of masonry walls.

8. The behavior of masonry walls is similar to that of plates, which suggests that the model developed here may be used for plates and beams.

8.3 Suggestions for Future Work

The work presented in this dissertation may be expanded to cover walls with openings. Additional wall patterns may also be studied.

BIBLIOGRAPHY

(1) ACI Committee 531. "Concrete Masonry Structures--Design and Construction." Proceedings, Journal of the American Concrete Institute, Vol. 67, No. 5 (May, 1967), pp. 380-403, and No. 6, (June, 1967), pp. 442-460.

(2) Amrhein, J. E., and W. L. Dickey. Masonry Design Manual. 2nd ed. Masonry Industry Advancement Committee, Masonry Institute, San Francisco, Calif., 1972.

(3) American Society for Testing and Materials. Annual Book of ASTM Standards. Part 16 (Specifications for C62, Building Brick; C90, Specification for Concrete Masonry Units; C129, Specification for Non-Load-Bearing Concrete Masonry Units; C270, Specification for Mortar for Masonry Units). Philadelphia, Pa., 1977.

(4) Anson, M., and K. Newman. "The Effect of Mix Proportions and Method of Testing on Poisson's Ratio for Mortars and Concretes." Magazine of Concrete Research, Vol. 18, No. 56 (September, 1966), pp. 115-129.

(5) Archer, J. S., and C. H. Samson, Jr. "Structural Idealization for Digital-Computer Analysis." Second Conference on Electronic Computation, ASCE, 1960, pp. 283-325.

(6) Benjamin, J. R., and H. A. Williams. "The Behavior of One-Story Brick Shear Walls." Journal of the Structural Division, ASCE, Vol. 84, No. ST4 (July, 1958), pp. 1723-1-1723-30.

(7) Baker, W. E. Explosions in Air. Austin: University of Texas Press, 1973.

(8) Blume, J. A. "Dynamic Characteristics of Multistory Buildings." Journal of the Structural Division, ASCE, Vol. 94, No. ST2 (February, 1968), pp. 377-402.

(9) Clough, R. W., and K. L. Benuska. "Nonlinear Earthquake Behavior of Tall Buildings." Journal of the Engineering Mechanics Division, ASCE, Vol. 93, No. EM3 (June, 1967), pp. 129-146.

(10) Copeland, R. E., and E. L. Saxer. "Tests of Structural Bond of Masonry Mortars to Concrete Block." Proceedings, Journal of the American Concrete Institute, Vol. 61, No. 11 (November, 1964), pp. 1411-1451.

(11) Dhir, R. K., C. M. Sangha, and J. G. L. Munday. "Strength and Deformation Properties of Autogenously Healed Mortars." Proceedings, Journal of the American Concrete Institute, Vol. 70, No. 3 (March, 1973), pp. 231-236.

(12) Fedorkiw, J. P., and M. A. Sozen. "A Lumped-Parameter Model to Simulate the Response of Reinforced Concrete Frames With Filler Walls." Civil Engineering Studies, Structural Research Series No. 338. Urbana: University of Illinois, June, 1968.

(13) Fishburn, C. C. "Effect of Mortar Properties on the Strength of Masonry." National Bureau of Standards, Monograph 36. Washington, D.C.: U.S. Government Printing Office, 1961.

(14) Gabrielsen, B., and C. Wilton. "Shock Tunnel Tests of Arched Wall Panels." Final report, URS 7030-19, URS Research Corporation, San Mateo, Calif., December, 1974.

(15) Gabrielsen, B., C. Wilton, and K. Kaplan. "Response of Arching Walls and Debris From Interior Walls Caused by Blast Loading." Final report, URS 7030-23, URS Research Corporation, San Mateo, Calif., February, 1975.

(16) Goodman, H. J. "Compiled Free-Air Blast Data on Bare Spherical Pentolite." Report No. 1092, Ballistic Research Laboratories, Aberdeen Proving Ground, Md., February, 1960.

(17) Goodman, R. E., R. L. Taylor, and T. L. Brekke. "A Model for the Mechanics of Jointed Rock." Journal of the Soil Mechanics and Foundations Division, ASCE, Vol. 94, No. SM3 (May, 1968), pp. 637-659.

(18) Grimm, C. T. "Strength and Related Properties of Brick Masonry." Journal of the Structural Division, ASCE, Vol. 101, No. ST1 (January, 1975), pp. 217-232.

(19) Hedstrom, R. O. "Load Tests of Patterned Concrete Masonry Walls." Proceedings, Journal of the American Concrete Institute, Vol. 57, Pt. 2 (April, 1961), pp. 1265-1286.

(20) Hedstrom, R. O., A. Litvin, and J. A. Hanson. "Influence of Mortar and Block Properties on Shrinkage Cracking of Masonry Walls." Journal of the PCA Research and Development Laboratories, Vol. 10, No. 1 (January, 1968), pp. 34-51.

(21) Hornbostel, D. Materials for Architecture. New York: Reinhold Publishing Company, 1961.

(22) Kinney, G. F. Explosive Shocks in Air. New York: The Macmillan Company, 1962.

(23) Merritt, M. L. "Blast Loading of Idealized Structures Using High Explosives." Proceedings, Symposium on Earthquake and Blast Effects on Structures. University of California at Los Angeles, June, 1952, pp. 74-93.

(24) Newmark, N. M. "Computation of Dynamic Structural Response in the Range Approaching Failure." Proceedings, Symposium on Earthquake and Blast Effects on Structures. University of California at Los Angeles, June, 1952, pp. 116-126.

(25) Newmark, N. M. "An Engineering Approach to Blast Resistant Design." Proceedings, ASCE, Vol. 79, Separate No. 306 (October, 1953), pp. 306-1-301-16.

(26) Newmark, N. M. "A Method of Computation for Structural Dynamics." Journal of the Engineering Mechanics Division, ASCE, Vol. 85, No. EM3 (July, 1959), pp. 67-94.

(27) Ngo, D., and A. C. Scordelis. "Finite Element Analysis of Reinforced Concrete Beams." Proceedings, Journal of the American Concrete Institute, Vol. 64, No. 3 (March, 1967), pp. 152-163.

(28) Nilson, A. H. "Nonlinear Analysis of Reinforced Concrete by the Finite Element Method." Proceedings, Journal of the American Concrete Institute, Vol. 65, No. 9 (September, 1968), pp. 757-766.

(29) Norris, C. H., R. J. Hansen, M. J. Holley, Jr., J. M. Biggs, S. Namyet, and J. K. Minami. Structural Design for Dynamic Loads. New York: McGraw-Hill Book Company, 1959.

(30) Selig, E. T. "Response of an Elastic Structure to Blast-Type Loading." Journal of the Boston Society of Civil Engineers, Vol. 47, No. 2 (April, 1960), pp. 169-182.

(31) Smith, B. S. "Lateral Stiffness of Infilled Frames." Journal of the Structural Division, ASCE, Vol. 88, No. ST6 (December, 1962), pp. 183-199.

(32) Smith, B. S. "Behavior of Square Infilled Frames." Journal of the Structural Division, ASCE, Vol. 92, No. ST1 (February, 1966), pp. 381-403.

(33) "A Simple Method for Evaluating Blast Effects on Buildings." Armour Research Foundation, Illinois Institute of Technology, July, 1954.

(34) Summers, L. H. "Dynamic Analysis of Frangible Walls Subjected to Blast Overpressures." (Unpublished Ph.D. thesis, University of Notre Dame, 1970.)

(35) Timoshenko, S., and S. Woinowsky-Kreiger. Theory of Plates and Shells. New York: McGraw-Hill Book Company, 1959.

(36) Volterra, E., and E. C. Zachmonoglou. Dynamics of Vibration. Columbus: Charles E. Merrill Books, Inc., 1965.

(37) Whitney, C. S., B. G. Anderson, and E. Cohen. "Design of Blast Resistant Construction for Atomic Explosions." Proceedings, Journal of the American Concrete Institute, Vol. 51 (March, 1955), pp. 589-673.

(38) Wiehle, C. K., and J. L. Bockholt. "Summary of Existing Structures Evaluation--Part I: Walls." Stanford Research Institute, Menlo Park, Calif., November, 1968.

(39) Willoughby, A. B., C. Wilton, and B. Gabrielsen. "Development and Evaluation of a Shock Tunnel Facility for Conducting Full-Scale Tests of Loading, Response, and Debris Characteristics of Structural Elements." Report No. URS 680-2, URS Research Company, San Mateo, Calif., December, 1967.

(40) Willoughby, A. B., C. Wilton, B. L. Gabrielsen, and J. B. Zaccor. "A Study of Loading, Structural Response, and Debris Characteristics of Wall Panels." Final report, URS 680-5, URS Research Company, San Mateo, Calif., July, 1969.

APPENDIX A

EXPRESSIONS FOR MASS AND MASS MOMENT OF INERTIA FOR CONCRETE BLOCKS

As mentioned earlier in connection with Equation (2.12) (see page 18), the mass and mass moment of inertia for concrete blocks are naturally different than those of solid bricks. The mass is calculated in a similar fashion to that of a solid brick, using the block weight in addition to a shell of mortar surrounding the block on its sides. No mortar is used on the webs of the block. Furthermore, mortar joints on the right and left sides are considered to have a width in the z-direction equivalent to that of the face shells for 3-core blocks, and the full block width for 2-core blocks.

The expressions for mass moment of inertia for full blocks are:

$$I_u = \{2W_s\ [w_j^2 + (2b + t_y)^2 + 12R_z^2] + W_{ri}\ (N_c - 1) + 2(W_{re} + W_{em})]\ [(2b)^2 + Z^2]\}/12g$$

$$I_v = \{2W_s\ [w_j^2 + (2a + t_x)^2 + 12R_z^2] + 2W_{re}\ (t_e^2 + Z^2 + 12R_{x1}^2) + W_{ri}\ (N_c - 1)(t_i^2 + Z^2 + 12R_{x2}^2) + 2W_{em}\ [(t_x/2)^2 + Z^2 + 12(a + t_x/4)^2]\}/12g$$

$$I_w = \{2W_s\ [(2a + t_x)^2 + (2b + t_y)^2] + 2W_{er}\ [t_e^2 + (2b)^2 + 12R_{x1}^2] + W_{ri}\ (N_c - 1)\ [t_i^2 + (2b)^2 + 12R_{x2}^2] + 2W_{em}\ [(t_x/2)^2 + (2b)^2 + 12\ (a + t_x/4)^2]\}/12g$$

where

N_c = number of cores;

a,b,c = half dimensions of the block;

W_s = weight of the face shell and mortar;

W_{ri} = weight of the interior rib;

W_{re} = weight of the exterior rib;

W_{em} = weight of end mortar attached to end rib (for 2-core blocks);

g = acceleration of gravity = 386.088 in./sec^2;

t_e, t_i, w_j, Z, R_z, R_{x1}, R_{x2} are shown in Figure 37; and t_x, t_y are joint thicknesses defined in Chapter II.

For the case where a portion of a block is necessary to complete a course for the running bond pattern, a new set of expressions for the mass moment of inertia must be developed. To begin with, the center of gravity for a portion of a block must be located. The expression for the center of gravity is given by

$$C_g = \{A_s\ (2a + t_x)/2 + A_r\ [E_h + (t_e + t_x)/2] + A_{t1}\ (X_2 + t_1/2)$$
$$+ A_{t2}\ (X_4 + t_2/2) + A_{t3}\ (X_6 + t_3/2) + A_{im}\ (t_x/4)$$
$$+ A_m\ (L_x + t_x/2)\}/A_T$$

where

A_s = area of the two face shells (including joint);

A_r = area of end web;

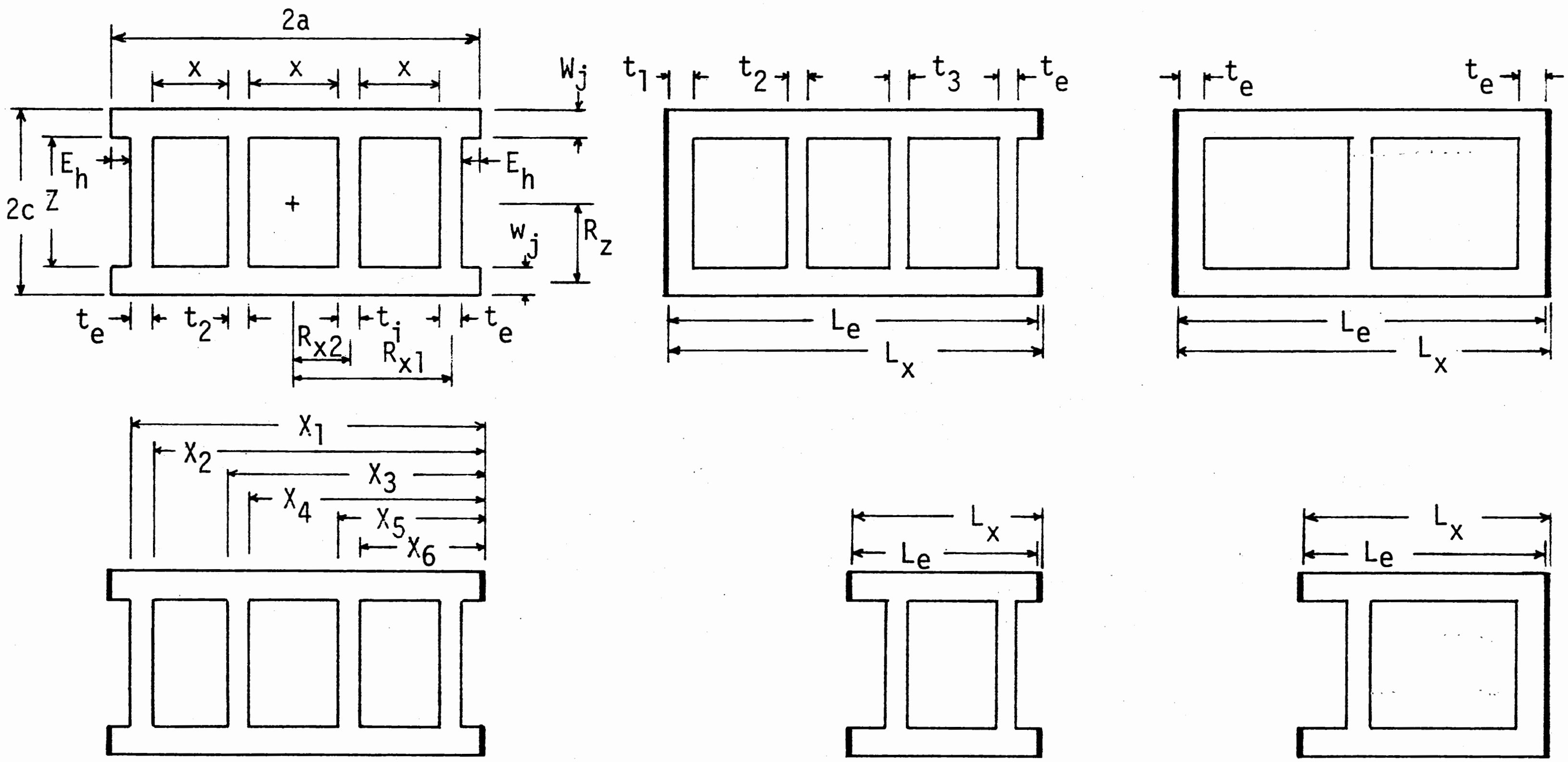
A_{t1}, A_{t2}, A_{t3} = areas of webs with widths t_1, t_2, and t_3, respectively (Figure 37);

A_{im} = area of mortar for 2-core blocks in excess of that for 3-core blocks = $Z\ t_x$;

A_m = equivalent to A_{im}; may be deleted when section at L_x does not pass through a web;

A_T = sum of the above areas.

The expressions for the mass moment of inertia are:



$R_z = c - w_j/2$

$R_{x1} = a - E_h - t_e/2$

$R_{x2} = a - E_h - t_e - x - t_i/2$

Portions of Blocks

(In all blocks shown in this figure the left joints are boundary joints)

Figure 37. Concrete Block Parameters for Calculation of the Mass Moment of Inertia

$$I_u = \{2W_s\ [w_j^2 + (2b + t_y)^2 + 12R_z^2]$$

$$+ [W_{re}\ (t_1 + t_e)/t_e + W_{ri}\ (t_2 + t_3)/t_i]$$

$$\cdot\ [(2b)^2 + Z^2]\}/12g$$

$$I_v = \{2W_s\ [w_j^2 + (2a + t_x)^2 + 12R_z^2]$$

$$+ W_{re}t_1\ [t_1^2 + Z^2 + 12\ (X_2 - C_g + t_1/2)^2]\ /t_e$$

$$+ W_{re}\ [t_e^2 + Z^2 + 12\ (C_g - t_x/2 - E_h - t_e/2)^2$$

$$+ W_{ri}t_2\ [t_2^2 + Z^2 + 12\ (X_4 - C_g - t_2/2)^2]/t_i$$

$$+ W_{ri}t_3\ [t_3^2 + Z^2 + 12\ (X_6 - C_g + t_3/2)^2]/t_i$$

$$+ W_{em}\ [(t_x/2)^2 + Z^2 + 12\ (L_x - C_g + t_x/4)^2]$$

$$+ W_{im}\ [(t_x/2)^2 + Z^2 + 12\ (C_g - t_x/4)^2\}/12g$$

$$I_w = \{2W_s\ [(2a + t_x)^2 + (2b + t_y)^2 + 12\ (L_x/2 - C_g)^2]$$

$$+ W_{re}t_1\ [t_1^2 + (2b)^2 + 12\ (X_2 - C_g + t_1/2)^2]/t_e$$

$$+ W_{re}\ [t_e^2 + (2b)^2 + 12\ (C_g - t_x/2 - E_h - t_e/2)^2]$$

$$+ W_{ir}t_2\ [t_2^2 + (2b)^2 + 12\ (X_4 - C_g + t_2/2)^2]/t_i$$

$$+ W_{ir}t_3\ [t_3^2 + (2b)^2 + 12\ (X_6 - C_g + t_3/2)^2]/t_i$$

$$+ W_{em}\ [(t_x/2)^2 + (2b)^2 + 12\ (L_x - C_g + t_x/4)^2]$$

$$+ W_{im}\ [(t_x/2)^2 + (2b)^2 + 12\ (C_g - t_x/4)^2]\}/12g.$$

APPENDIX B

STIFFNESS MATRICES

$$
[K^B] = \begin{bmatrix}
K_u^B & 0 & 0 & 0 & c(1-\lambda_w)\,K_{u\beta}^B & b(1-\lambda_v)\,K_{u\beta}^B \\
0 & K_v^B & 0 & c(1-\lambda_w)\,K_{v\theta}^B & 0 & -(a+t_x/2)\,K_v^B \\
0 & 0 & K_w^B & b(1-\lambda_v)\,K_{w\theta}^B & (a+t_x/2)\,K_w^B & 0 \\
0 & K_{24}^B & K_{34}^B & \begin{matrix} c^2(1-\lambda_w)^2\,K_v^B \\ +\,b^2(1-\lambda_v)^2\,K_w^B \end{matrix} & (a+t_x/2)\,K_{34}^B & -(a+t_x/2)\,K_{24}^B \\
K_{15}^B & 0 & -K_{35}^B & -K_{45}^B & \begin{matrix} c^2(1-\lambda_w)^2\,K_u^B \\ -(a+t_x/2)^2\,K_w^B \end{matrix} & c(1-\lambda_w)\,b(1-\lambda_v)\,K_{\beta\phi}^B \\
K_{16}^B & -K_{26}^B & 0 & -K_{46}^B & K_{56}^B & \begin{matrix} b^2(1-\lambda_v)^2\,K_u^B \\ -(a+t_x/2)^2\,K_v^B \end{matrix}
\end{bmatrix}
$$

$K_u^B = k_1 + k_7 + k_{13} + k_{19}$ $\qquad$ $K_{\beta\phi}^B = -k_1 + k_7 + k_{13} - k_{19}$ $\qquad$ $K_w^B = k_3 + k_9 + k_{15} + k_{21}$

$K_{u\beta}^B = k_1 + k_7 - k_{13} - k_{19}$ $\qquad$ $K_v^B = k_2 + k_8 + k_{14} + k_{20}$ $\qquad$ $K_{w\theta}^B = k_3 - k_9 + k_{15} - k_{21}$

$K_{u\phi}^B = -k_1 + k_7 - k_{13} + k_{19}$ $\qquad$ $K_{v\theta}^B = -k_2 - k_8 + k_{14} + k_{20}$

$$
[K^C] = \begin{bmatrix}
K_u^C & 0 & 0 & 0 & c(1-\lambda_w)\,K_{u\beta}^C & (b + t_y/2)\,K_u^C \\
0 & K_v^C & 0 & c(1-\lambda_w)\,K_{v\theta}^C & 0 & \begin{matrix} a(1-\lambda_{ur})\,K_{v\phi r}^C \\ + a(1-\lambda_{u\ell})\,K_{v\phi\ell}^C \end{matrix} \\
0 & 0 & K_w^C & -(b + t_y/2)\,K_w^C & \begin{matrix} -a(1-\lambda_{ur})\,K_{w\beta r}^C \\ -a(1-\lambda_{u\ell})\,K_{w\beta\ell}^C \end{matrix} & 0 \\
0 & K_{24}^C & -K_{34}^C & \begin{matrix} c^2(1-\lambda_w)^2\,K_v^C \\ -(b + t_y/2)^2\,K_W^C \end{matrix} & (b + t_y/2)\,K_{35}^C & \begin{matrix} c(1-\lambda_w)[a(1-\lambda_{ur})\,K_{\theta\phi r}^C \\ + a(1-\lambda_{u\ell})\,K_{\theta\phi\ell}^C] \end{matrix} \\
K_{15}^C & 0 & K_{35}^C & -K_{45}^C & \begin{matrix} c^2(1-\lambda_w)^2\,K_u^C \\ +a^2(1-\lambda_{ur})^2\,K_{wr}^C \\ +a^2(1-\lambda_{u\ell})^2\,K_{w\ell}^C \end{matrix} & (b + t_y/2)\,K_{15}^C \\
-K_{16}^C & K_{26}^C & 0 & K_{46}^C & -K_{56}^C & \begin{matrix} a^2(1-\lambda_{ur})^2\,K_{vr}^C \\ + a^2(1-\lambda_{u\ell})^2\,K_{v\ell}^C \\ - (b + t_y/2)^2\,K_u^C \end{matrix}
\end{bmatrix}
$$

$K_u^C = k_4 + k_{16} + k_{25} + k_{37}$ $\quad K_v^C = k_5 + k_{17} + k_{26} + k_{38}$ $\quad K_w^C = k_6 + k_{18} + k_{27} + k_{39}$

$K_{u\beta}^C = k_4 - k_{16} - k_{25} + k_{37}$ $\quad K_{v\theta}^C = -k_5 + k_{17} + k_{26} - k_{38}$ $\quad K_{w\beta}^C = k_6 + k_{18} - k_{27} - k_{39}$

$K_{vr}^C = k_5 + k_{17}$; $K_{v\ell}^C = k_{26} + k_{28}$ $\quad K_{v\phi}^C = k_5 + k_{17} - k_{26} - k_{38}$ $\quad K_{wr}^C = k_6 + k_{18}$; $K_{w\ell}^C = k_{27} + k_{39}$

$K_{v\phi r}^C = k_5 + k_{17}$; $K_{v\phi\ell}^C = -(k_{26} + k_{38})$ $\quad K_{\theta\phi}^C = -k_5 + k_{17} - k_{26} + k_{38}$ $\quad K_{w\beta r}^C = k_6 + k_{18}$; $K_{w\beta\ell}^C = -(k_{27} + k_{39})$

$K_{\theta\phi r}^C = -k_5 + k_{17}$; $K_{\theta\phi\ell}^C = -k_{26} + k_{38}$

$$
[K^D] = \begin{bmatrix}
K_u^D & 0 & 0 & 0 & c(1-\lambda_w)\,K_{u\beta}^D & b(1-\lambda_v)\,K_{u\phi}^D \\
0 & K_v^D & 0 & c(1-\lambda_w)\,K_{v\theta}^D & 0 & (a+t_x/2)\,K_v^D \\
0 & 0 & K_w^D & b(1-\lambda_v)\,K_{w\theta}^D & -(a+t_x/2)\,K_w^D & 0 \\
0 & K_{24}^D & K_{34}^D & \begin{matrix} c^2(1-\lambda_w)^2\,K_v^D \\ +b^2(1-\lambda_v)^2\,K_w^D \end{matrix} & -(a+t_x/2)\,K_{34}^D & (a+t_x/2)\,K_{24}^D \\
K_{15}^D & 0 & -K_{35}^D & -K_{45}^D & \begin{matrix} c^2(1-\lambda_w)^2\,K_u^D \\ -(a+t_x/2)^2\,K_w^D \end{matrix} & c(1-\lambda_w)\,b(1-\lambda_w)\,K_{\beta\phi}^D \\
K_{16}^D & -K_{26}^D & 0 & -K_{46}^D & K_{56}^D & \begin{matrix} b^2(1-\lambda_v)^2\,K_u^D \\ -(a+t_x/2)^2\,K_v^D \end{matrix}
\end{bmatrix}
$$

$$K_u^D = k_{28} + k_{31} + k_{40} + k_{43}$$

$$K_{u\beta}^D = -k_{28} - k_{31} + k_{40} + k_{43}$$

$$K_{u\phi}^D = -k_{28} - k_{31} - k_{40} + k_{43}$$

$$K_{\beta\phi}^D = k_{28} - k_{31} - k_{40} + k_{43}$$

$$K_v^D = k_{29} + k_{32} + k_{41} + k_{44}$$

$$K_{v\theta}^D = k_{29} + k_{32} - k_{41} - k_{44}$$

$$K_w^D = k_{30} + k_{33} + k_{42} + k_{45}$$

$$K_{w\theta}^D = k_{30} - k_{33} + k_{42} + k_{45}$$

$$
[K^E] = \begin{bmatrix}
K_u^E & 0 & 0 & 0 & c(1-\lambda_w)\,K_{u\beta}^E & -(b+t_y/2)\,K_u^E \\
0 & K_v^E & 0 & -c(1-\lambda_w)\,K_{v\theta}^E & 0 & a(1-\lambda_{ur})\,K_{v\phi r}^E + a(1-\lambda_{u\ell})\,K_{v\phi\ell}^E \\
0 & 0 & K_w^E & (b+t_y/2)\,K_w^E & -a(1-\lambda_{ur})\,K_{w\beta r}^E - a(1-\lambda_{u\ell})\,K_{w\beta\ell}^E & 0 \\
0 & K_{24}^E & -K_{34}^E & c^2(1-\lambda_w)^2\,K_v^E - (b+t_y/2)^2\,K_w^E & -(b+t_y/2)\,K_{35}^E & c(1-\lambda_w)[a(1-\lambda_{ur})\,K_{\theta\phi r}^E + a(1-\lambda_{u\ell})\,K_{\theta\phi\ell}^E] \\
K_{15}^E & 0 & K_{35}^E & -K_{45}^E & c^2(1-\lambda_w)^2\,K_u^E + a^2(1-\lambda_{ur})^2\,K_{wr}^E + a^2(1-\lambda_{u\ell})^2\,K_{w\ell}^E & -(b+t_y/2)\,K_{15}^E \\
-K_{16}^E & K_{26}^E & 0 & K_{46}^E & -K_{56}^E & a^2(1-\lambda_{ur})^2\,K_{vr}^E + a^2(1-\lambda_{u\ell})^2\,K_{v\ell}^E - (b+t_y/2)^2\,K_u^E
\end{bmatrix}
$$

$K_u^E = k_{10} + k_{22} + k_{34} + k_{46}$ $\quad K_v^E = k_{11} + k_{23} + k_{35} + k_{47}$ $\quad K_w^E = k_{12} + k_{24} + k_{36} + k_{48}$

$K_{u\beta}^E = k_{10} - k_{22} - k_{34} + k_{46}$ $\quad K_{v\theta}^E = k_{11} - k_{23} - k_{35} + k_{47}$ $\quad K_{w\beta}^E = k_{12} + k_{24} - k_{36} - k_{48}$

$K_{vr}^E = k_{11} + k_{23};\ K_{v\ell}^E = k_{35} + k_{47}$ $\quad K_{v\phi}^E = k_{11} + k_{23} - k_{35} - k_{47}$ $\quad K_{wr}^E = k_{12} + k_{24};\ K_{w\ell}^E = k_{36} + k_{48}$

$K_{v\phi r}^E = k_{11} + k_{23};\ K_{v\phi\ell}^E = -(k_{35} + k_{47})$ $\quad K_{\theta\phi}^E = -k_{11} + k_{23} - k_{35} + k_{47}$ $\quad K_{w\beta r}^E = k_{12} + k_{24};\ K_{w\beta\ell}^E = -(k_{36} + k_{48})$

$K_{\theta\phi r}^E = -k_{11} + k_{23};\ K_{\theta\phi\ell}^E = -k_{35} + k_{47}$

$$
[K^{CR}] = \begin{bmatrix}
K_u^{CR} & 0 & 0 & 0 & c(1-\lambda_w)\ K_{u\beta}^{CR} & (b + t_y/2)\ K_u^{CR} \\
0 & K_v^{CR} & 0 & -c(1-\lambda_w)\ K_{v\theta}^{CR} & 0 & -a(1-\lambda_{ur})\ K_v^{C} \\
0 & 0 & K_w^{CR} & -(b + t_y/2)\ K_w^{CR} & a(1-\lambda_{ur})\ K_w^{CR} & 0 \\
0 & K_{24}^{CR} & -K_{34}^{CR} & \begin{array}{c} c^2(1-\lambda_w)^2\ K_v^{CR} \\ -(b + t_y/2)^2\ K_w^{CR} \end{array} & (b + t_y/2)\ K_{35}^{CR} & -a(1-\lambda_{ur})K_{24}^{CR} \\
K_{15}^{CR} & 0 & -K_{35}^{CR} & K_{45}^{CR} & \begin{array}{c} c^2(1-\lambda_w)^2\ K_u^{CR} \\ -a^2(1-\lambda_{ur})^2\ K_w^{CR} \end{array} & (b + t_y/2)\ K_{15}^{CR} \\
-K_{16}^{CR} & -K_{26}^{CR} & 0 & -K_{46}^{CR} & -K_{56}^{CR} & \begin{array}{c} -(b + t_y/2)^2\ K_u^{CR} \\ -a^2(1-\lambda_{ur})^2\ K_v^{CR} \end{array}
\end{bmatrix}
$$

$K_u^{CR} = k_4 + k_{16}$ $\quad$ $K_v^{CR} = k_5 + k_{17}$ $\quad$ $K_w^{CR} = k_6 + k_{18}$

$K_{u\beta}^{CR} = k_4 - k_{16}$ $\quad$ $K_{v\theta}^{CR} = k_5 - k_{17}$

$$
[K^{ER}] = \begin{bmatrix}
K_u^{ER} & 0 & 0 & 0 & c(1-\lambda_w)\ K_{u\beta}^{ER} & (b + t_y/2)\ K_u^{ER} \\
0 & K_v^{ER} & 0 & c(1-\lambda_w)\ K_{v\theta}^{ER} & 0 & a(1-\lambda_{u\ell})\ K_v^{ER} \\
0 & 0 & K_w^{ER} & -(b + t_y/2)\ K_w^{ER} & -a(1-\lambda_{u\ell})\ K_w^{ER} & 0 \\
0 & K_{24}^{ER} & -K_{34}^{ER} & \begin{matrix} c^2(1-\lambda_w)^2\ K_v^{ER} \\ -(b + t_y/2)^2\ K_w^{ER} \end{matrix} & (b + t_y/2)\ K_{35}^{ER} & a(1-\lambda_{u\ell})\ K_{24}^{ER} \\
K_{15}^{ER} & 0 & -K_{35}^{ER} & K_{45}^{ER} & \begin{matrix} c^2(1-\lambda_w)^2\ K_u^{ER} \\ -a^2(1-\lambda_{u\ell})^2\ K_w^{ER} \end{matrix} & (b + t_y/2)\ K_{15}^{ER} \\
-K_{16}^{ER} & -K_{26}^{ER} & 0 & -K_{46}^{ER} & K_{56}^{ER} & \begin{matrix} -(b + t_y/2)^2\ K_u^{ER} \\ -a^2(1-\lambda_{u\ell})^2\ K_v^{ER} \end{matrix}
\end{bmatrix}
$$

$$
K_u^{ER} = k_{25} + k_{37} \qquad K_v^{ER} = k_{26} + k_{38} \qquad K_w^{ER} = k_{27} + k_{39}
$$

$$
K_{u\beta}^{ER} = -k_{25} + k_{37} \qquad K_{v\theta}^{ER} = k_{26} - k_{38}
$$

$$
[K^F] = \begin{bmatrix}
K_u^F & 0 & 0 & 0 & -c(1-\lambda_w)\,K_{u\beta}^F & -(b + t_y/2)\,K_u^F \\
0 & K_v^F & 0 & c(1-\lambda_w)\,K_{v\theta}^F & 0 & a(1-\lambda_{u\ell})\,K_v^F \\
0 & 0 & K_w^F & (b + t_y/2)\,K_w^F & -a(1-\lambda_{u\ell})\,K_w^F & 0 \\
0 & K_{24}^F & -K_{34}^F & \begin{matrix} c^2(1-\lambda_w)^2\,K_v^F \\ -(b + t_y/2)^2\,K_w^F \end{matrix} & -(b + t_y/2)\,K_{35}^F & a(1-\lambda_{u\ell})\,K_{24}^F \\
K_{15}^F & 0 & -K_{35}^F & K_{45}^F & \begin{matrix} c^2(1-\lambda_w)^2\,K_u^F \\ -a^2(1-\lambda_{u\ell})^2\,K_w^F \end{matrix} & -(b + t_y/2)\,K_{15}^F \\
-K_{16}^F & -K_{26}^F & 0 & -K_{46}^F & -K_{56}^F & \begin{matrix} -a^2(1-\lambda_{u\ell})^2\,K_v^F \\ -(b + t_y/2)^2\,K_u^F \end{matrix}
\end{bmatrix}
$$

$$K_u^F = k_{34} + k_{46} \qquad K_v^F = k_{35} + k_{47} \qquad K_w^F = k_{36} + k_{48}$$

$$K_{u\beta}^F = k_{34} - k_{46} \qquad K_{v\theta}^F = k_{35} - k_{47}$$

$$
[K^H] = \begin{bmatrix}
K_u^H & 0 & 0 & 0 & c(1-\lambda_w)\,K_{u\beta}^H & -(b + t_y/2)\,K_u^H \\
0 & K_v^H & 0 & c(1-\lambda_w)\,K_{v\theta}^H & 0 & -a(1-\lambda_{ur})\,K_v^E \\
0 & 0 & K_w^H & (b + t_y/2)\,K_w^H & a(1-\lambda_{ur})\,K_w^H & 0 \\
0 & K_{24}^H & -K_{34}^H & c^2(1-\lambda_w)^2\,K_v^H - (b + t_y/2)^2\,K_w^H & -(b + t_y/2)\,K_{35}^H & -a(1-\lambda_{ur})\,K_{24}^H \\
K_{15}^H & 0 & -K_{35}^H & K_{45}^H & c^2(1-\lambda_w)^2\,K_u^H - a^2(1-\lambda_{ur})^2\,K_w^H & -(b + t_y/2)\,K_{15}^H \\
K_{16}^H & -K_{26}^H & 0 & -K_{46}^H & -K_{56}^H & -a^2(1-\lambda_{ur})^2\,K_v^H - (b + t_y/2)^2\,K_u^H
\end{bmatrix}
$$

$K_u^H = k_{10} + k_{22}$ $\quad K_v^H = k_{11} + k_{23}$ $\quad K_w^H = k_{12} + k_{24}$

$K_{u\beta}^H = k_{10} - k_{22}$ $\quad K_{v\theta}^H = -k_{11} + k_{23}$

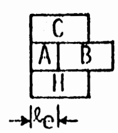
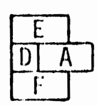
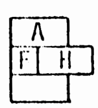
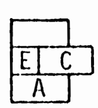
$$
[K^A] = \begin{bmatrix}
K_u^B + K_u^C + K_u^D + K_u^E & 0 & 0 & 0 & K_{15}^B + K_{15}^C + K_{15}^D + K_{15}^E & K_{16}^B + K_{16}^D - K_{16}^C - K_{16}^E \\
 & K_v^B + K_v^C + K_v^D + K_v^E & 0 & K_{24}^B + K_{24}^C + K_{24}^D + K_{24}^E & 0 & K_{26}^C + K_{26}^E - K_{26}^B - K_{26}^D \\
 & & K_w^B + K_w^C + K_w^D + K_w^E & K_{34}^B + K_{34}^D - K_{34}^C - K_{34}^E & K_{35}^C + K_{35}^E - K_{35}^B - K_{35}^D & 0 \\
 & & & \begin{gathered} c^2(1-\lambda_w)^2 K_{22}^A \\ + b^2(1-\lambda_v)^2 (K_w^B + K_w^D) \\ + (b + t_y/2)^2 (K_w^C + K_w^E) \end{gathered} & K_{45}^C + K_{45}^E - K_{45}^B - K_{45}^D & K_{46}^C + K_{46}^E - K_{46}^B - K_{46}^D \\
 & & & & \begin{gathered} c^2(1-\lambda_w)^2 K_{11}^A \\ + a^2(1-\lambda_{ur})^2 (K_{wr}^C + K_{wr}^E) \\ + a^2(1-\lambda_{u\ell})^2 (K_{w\ell}^C + K_{w\ell}^E) \\ + (a + t_x/2)^2 (K_w^B + K_w^D) \end{gathered} & K_{56}^B + K_{56}^D - K_{56}^C - K_{56}^E \\
 & \text{Symmetric} & & & & \begin{gathered} a^2(1-\lambda_{ur})^2 (K_{vr}^C + K_{vr}^E) \\ + a^2(1-\lambda_{u\ell})^2 (K_{v\ell}^C + K_{v\ell}^E) \\ + b^2(1-\lambda_v)^2 (K_u^B + K_u^D) \\ + (a + t_x/2)^2 (K_v^B + K_v^D) \\ + (b + t_y/2)^2 (K_u^C + K_u^E) \end{gathered}
\end{bmatrix}
$$

Boundary Modifications for Stiffness Matrices

Due to the geometry of the running bond pattern, certain courses start and/or end with a less than full masonry unit. In such cases, matrices $[K^A]$, $[K^B]$, $[K^{CR}]$, $[K^{ER}]$, $[K^D]$, $[K^F]$, and $[K^H]$ given on the previous pages of this appendix need to be modified. The particular cases where such modifications are necessary are tabulated in Table V on the following page, along with the required changes.

TABLE V

MODIFICATIONS IN STIFFNESS MATRICES FOR LEFT* BOUNDARY

Case		Matrix Elements Affected		
I		35,53 45,54 26,62 46,64 55 66	$[K^A]$	Replace a by ℓ_e†/2 $a(1-\lambda_{ur}) = 0$ $a(1-\lambda_{u\ell}) = 0$
		53 54 62 64	$[K^B]$	Replace a by $\ell_e/2$
		55		$\ldots -\frac{1}{2}(\ell_e + t_x)(a + t_x/2)\, K_w^B$
		66		$\ldots -\frac{1}{2}(\ell_e + t_x)(a + t_x/2)\, K_v^B$
		53 54 62 64	$[K^{CR}]$ and	Zero
		55 66	$[K^H]$	$a(1-\lambda_{ur}) = 0$
II		35 45 26 46	$[K^D]$	Replace a by $\ell_e/2$
		55		$\ldots -\frac{1}{2}(\ell_e + t_x)(a + t_x/2)\, K_w^D$
		66		$\ldots -\frac{1}{2}(\ell_e + t_x)(a + t_x/2)\, K_v^D$
III		35 45 26 46	$[K^F]$	Zero
		55 66		$a(1-\lambda_{u\ell}) = 0$
IV		35 45 26 46	$[K^{ER}]$	Zero
		55 66		$a(1-\lambda_{u\ell}) = 0$

*The same changes apply for the right boundary, except in Cases I and II, where $[K^B]$ changes apply to $[K^D]$ instead and vice versa.

†ℓ_e is the actual length of an end block.

APPENDIX C

PROGRAM "WALBLAST": LISTING

```
C       +--------------------------------------------------------------+
C       I                                                              I
C       I              DYNAMIC ANALYSIS OF MASONRY WALLS               I
C       I              ------- -------- -- ------- -----               I
C       I                                                              I
C       I                                                              I
C       I                       T. M. AL-ASWAD                         I
C       I                                                              I
C       I                SCHOOL OF CIVIL ENGINEERING                   I
C       I                OKLAHOMA STATE UNIVERSITY                     I
C       I                  STILLWATER, OKLAHOMA                        I
C       I                                                              I
C       I                                                              I
C       I WALL PATTERNS: HORIZONTAL STACK, RUNNING BOND                I
C       I                                                              I
C       I MASONRY UNITS: SOLID BRICKS, 2- AND 3-CORE CONCRETE BLOCKS   I
C       I                                                              I
C       I SUPPORT CONDITIONS: FREE, SIMPLE, FIXED, SYMMETRY            I
C       I                                                              I
C       I LOADING DISTRIBUTION: UNIFORM, SINUSOIDAL, OR A COMBINATION  I
C       I                                                              I
C       I FORCING FUNCTIONS: FIVE DIFFERENT PULSES                     I
C       I                                                              I
C       I                                                              I
C       I FOR ADDITIONAL INFORMATION, REFER TO ACCOMPANYING PH.D.      I
C       I DISSERTATION .                                               I
C       I                                                              I
C       +--------------------------------------------------------------+
```

```
C-->>     MAIN PROGRAM
C
C     ******************************************************************
C     *                                                                *
C     * FUNCTION: CALL MAJOR SUBROUTINES                               *
C     *                                                                *
C     ******************************************************************
C
      DIMENSION IDR(40), IDP(19)
      DATA  IBLANK /4H    /
C
C-->>   READ IDENTIFICATION OF RUN -- TWO CARDS
C
      READ 1000, IDR
      PRINT 2005
      PRINT 2010, IDR
  100     CONTINUE
C
C-->>   READ IDENTIFICATION OF PROBLEM -- ONE CARD
C
      READ 1000, NPROB, IDP
      PRINT 2020, NPROB, IDP
C
C-->>   CHECK PROBLEM NAME.  STOP IF BLANK
C
          IF (NPROB .EQ. IBLANK) GO TO 200
C
C-->>   CALL SUBROUTINE "ID" TO READ AND PRINT PROBLEM DATA
      CALL ID
C-->>   CALL SUBROUTINE "CORDS" TO ASSIGN NUMBERS TO AND CALCULATE
C-->>   COORDINATES OF BLOCKS
      CALL CORDS
C-->>   CALL SUBROUTINE "LINKEL" TO CALCULATE SPRING STIFFNESSES
      CALL LINKEL
C-->>   CALL SUBROUTINE "SOLVER" TO FORMULATE AND SOLVE
C-->>   EQUATIONS OF MOTION
      CALL SOLVER
C
          GO TO 100
  200 PRINT 3000
      PRINT 2005
 1000 FORMAT (20A4)
 2005 FORMAT (1H1)
 2010 FORMAT (1H , 20A4)
 2020 FORMAT (1H0, 3X,8HPROBLEM , A4, 2H:  , 19A4)
 3000 FORMAT (1H0, 20H  >>> END OF RUN <<< )
      STOP
      END
```

```
      SUBROUTINE ID
C
C     ****************************************************************
C     *                                                              *
C     * FUNCTION: READ AND PRINT PROBLEM DATA                        *
C     *                                                              *
C     ****************************************************************
C
      IMPLICIT REAL * 8 (A-H , O-Z)
      COMMON /ELAST / DPEAK, SPEAK, RTIME, CTIME, DTIME, TYPE,ILOAD,KDYN
      COMMON /BLCK  / A, B, C, WJ, TI, TE, EH, ELENTH
      COMMON /CURVE / NSTAGE(48,125), STRESS( 5), STRAIN( 5), E(5), NSEG
      COMMON /GLAMKS/ KS(48,125), G(48,6), LAMR, LAML, LAMV, LAMW, AKK
      COMMON /JOINTS/ TXJ, TYJ, TLJ, TRJ, TUJ, TBJ
      COMMON /STRNTH/ TSBOND, SSBOND, CSMORT
      COMMON /WALL  / L, L1, L2, H, H2, ILS, IRS, IUS, IBS, KODE
      COMMON /WEIGHT/ WT, MORTWT, IBLOCK
      DOUBLE PRECISION L, L1, L2, LI, H, HI, H2, KS
      DOUBLE PRECISION LAMR, LAML, LAMV, LAMW, MORTWT
      INTEGER  HF
      DATA  TWO, TWELVE /2.0D00, 12.0D00/
      DATA  TYP1, TYP2 /1HI, 2HII/
      DATA  FIX, SIM, FRE, SYM /4HFIXD, 4HSMPL, 4HFREE, 4HSYMM/
C
C-->>   READ WALL INFORMATION
C
      READ 1000, KODE, LF, LI, HF, HI
C
C-->>    READ SUPPORT CONDITIONS
C
      READ 1005, ILS, IRS, IUS, IBS
          IF (ILS .EQ. 0) SL = FRE
          IF (ILS .EQ. 1) SL = SIM
          IF (ILS .EQ. 2) SL = FIX
          IF (ILS .EQ. 3) SL = SYM
          IF (IRS .EQ. 0) SR = FRE
          IF (IRS .EQ. 1) SR = SIM
          IF (IRS .EQ. 2) SR = FIX
          IF (IRS .EQ. 3) SR = SYM
          IF (IUS .EQ. 0) SU = FRE
          IF (IUS .EQ. 1) SU = SIM
          IF (IUS .EQ. 2) SU = FIX
          IF (IUS .EQ. 3) SU = SYM
          IF (IBS .EQ. 0) SB = FRE
          IF (IBS .EQ. 1) SB = SIM
          IF (IBS .EQ. 2) SB = FIX
          IF (IBS .EQ  3) SB = SYM
C
C-->>   READ BLOCK INFORMATION
C
      READ 1010, IBLOK, A2, B2, C2, WT, WJ, TI, TE, EH
C
C-->>   READ SPRING LOCATIONS
C
      READ 1030, LAMR, LAML, LAMV, LAMW
C
C-->>   READ NO. OF SEGMENTS IN STRESS-STRAIN CURVE FOR MORTAR
C
      READ 1010, NSEG
C
C-->>   READ COORDINATES OF STRESS-STRAIN CURVE
C
         DO 150 I=1,NSEG
      READ 1050, STRESS(I), STRAIN (I)
  150    CONTINUE
C
C-->>   READ MORTAR PROPERTIES
C
      READ 1030, TSBOND, SSBOND, CSMORT, MORTWT
C
C-->>   READ INTERIOR AND BOUNDARY JOINT WIDTHS
C
      READ 1030, TXJ, TYJ, TLJ, TRJ, TUJ, TBJ
C
C-->>   READ LOADING INFORMATION
C
      READ 1040,  TYPE, ILOAD, DPEAK, SPEAK, RTIME, CTIME, DTIME
C
      PRINT 2020
      PRINT 2040
              IKODE = KODE + 1
         GO TO (300, 350, 350, 400, 400),IKODE
  300 PRINT 2060, LF, LI, HF, HI
         GO TO 450
  350 PRINT 2061, LF, LI, HF, HI, TYP1, KODE
         GO TO 450
  400    CONTINUE
      PRINT 2061, LF, LI, HF, HI, TYP2, KODE
  450 PRINT 2070
      PRINT 2075, SL, SR, SU, SB
      PRINT 2080
         IF (IBLOK .EQ. 1) GO TO 500
      PRINT 2100, A2, B2, C2, TI, TE, WJ, EH, WT, IBLOK
         GO TO 550
  500 PRINT 2101, A2, B2, C2, WT
  550 PRINT 2120
      PRINT 2140, TXJ, TYJ, TLJ, TRJ, TUJ, TBJ
      PRINT 2160
      PRINT 2180, TSBOND, SSBOND, CSMORT, MORTWT
      PRINT 2200
```

```
      PRINT 2220, LAMR, LAML, LAMV, LAMW
      PRINT 2260
      PRINT 2261, (STRESS(I), I=1,NSEG)
      PRINT 2262, (STRAIN(I), I=1,NSEG)
      PRINT 2280
      PRINT 2300,  TYPE, ILOAD, DPEAK, SPEAK, RTIME, CTIME, DTIME
      PRINT 2350
C
 1000 FORMAT (I1, I4, F10.0, I5, F10.0)
 1005 FORMAT (4I5)
 1010 FORMAT (I5, 4F10.0, 4F5.0)
 1030 FORMAT (6F10.0)
 1040 FORMAT (A4, 5X, I1, 5F10.0)
 1050 FORMAT (D10.3, 5X, D10.3)
 2020 FORMAT (1H0, 1(/), 35(1H*), 3X,  7HD A T A, 3X, 35(1H*),
     1        6X, 37(1H*), /, 89X, 1H*, 35X, 1H*, /, 89X, 1H*, 3X,
     2        19HPROGRAM  "WALBLAST", 8X, 1H*, /, 89X, 1H*, 14X,
     3        7HFOR THE, 14X, 1H*)
 2040 FORMAT (4X, 15(2H--), 2X, 11HWALL SYSTEM, 2X, 27(1H-),13X, 1H*,
     A        1X, 33HDYNAMIC ANALYSIS OF MASONRY WALLS, 1X, 1H*,
     B         1(/),  4X, 20HCLEAR LENGTH(FT--IN), 4X, 12HCLEAR HEIGHT,
     C        8H(FT--IN), 4X, 7HPATTERN, 3X, 4HTYPE, 3X, 7HVERSION,
     D        13X,1H*, 35X, 1H*)
 2060 FORMAT (7X, I3, 5X, F6.3, 10X, I3, 5X, F6.3, 7X, 8HH. STACK, 2X,
     1        14HNOT APPLICABLE, 13X, 1H*, 1X, 17HPROGRAMMER: T.M. ,
     2        8HAL-ASWAD, 9X, 1H*, /, 89X, 1H*, 35X, 1H*, /,89X,1H*,11X,
     3        12HSPRING, 1978, 12X, 1H*, /, 89X, 1H*, 35X, 1H*)
 2061 FORMAT (7X, I3, 5X, F6.3, 10X, I3, 5X, F6.3, 7X, 7HR. BOND, 4X,
     A        A2, 7X, I1, 16X, 1H*, 1X, 17HPROGRAMMER: T.M. ,
     B        8HAL-ASWAD, 9X, 1H*, /,89X,1H*, 35X,1H*, /,89X,1H*,11X,
     C        12HSPRING, 1978, 12X, 1H*, /, 89X, 1H*, 35X, 1H*)
 2070 FORMAT (17X, 17(1H-), 2X, 12HSUPPORT TYPE, 2X, 17(1H-), 22X, 1H*,
     1        4X, 27HSCHOOL OF CIVIL ENGINEERING, 4X, 1H*, /, 17X,
     2        4HLEFT, 10X, 5HRIGHT,12X, 5HUPPER, 9X, 5HLOWER, 22X, 1H*,
     3        5X, 25HOKLAHOMA STATE UNIVERSITY, 5X, 1H*)
 2075 FORMAT (17X, A4,10X, A4,13X, A4,10X, A4,23X,1H*, 7X,10HSTILLWATER,
     $        10H, OKLAHOMA, 8X,1H*,/, 89X,1H*, 35X,1H*,/, 89X, 37(1H*))
 2080 FORMAT (1H ,        4X, 17(2H--), 1X, 6HBLOCKS, 2X, 17(2H--), 1H ,
     D        2(/), 5X, 6HLENGTH, 4X, 6HHEIGHT, 4X, 5HDEPTH, 3X,
     E        24HI.WEB  E.WEB  FACE  HEEL, 4X, 6HWEIGHT, 9X, 4HTYPE,
     F        1(/), 1H , 20X, 23H(I   N   C   H   E   S),17X,5H(LBS),/)
 2100 FORMAT (5X, F6.3, 4X, F6.3, 4X, F6.3, 3X, 4(F5.3, 1X), 3X, F6.3,
     F        4X, I1, 6H-CORE , 8HCONC BLK)
 2101 FORMAT (5X, F6.3, 3X, F6.3, 4X, F6.3, 7X, 14HNOT APPLICABLE, 10X,
     G         F6.3, 5X, 11HSOLID BRICK)
 2120 FORMAT (1H0, 1(/), 10X, 7(2H--), 1X,26HJOINT THICKNESSES (INCHES),
     H        2X, 8(2H--), 1H , 1(/), 4X, 3(2H--), 8HINTERIOR, 1X,
     I        2(2H--), 26X, 6(2H--), 8HBOUNDARY, 1X, 6(2H--), 1H , 1(/),
     J        6X, 4HHEAD, 8X, 3HBED, 29X, 4HLEFT, 4X, 5HRIGHT, 4X,
     K        5HUPPER, 4X, 5HLOWER)
 2140 FORMAT (5X, F5.3, 7X, F5.3, 27X, 4(F5.3, 4X))
 2160 FORMAT (1H0, 1(/),  3X, 10(2H--), 2X, 17HMORTAR PROPERTIES, 2X,
     L        15(2H--), 1H , 1(/), 8X, 18HBOND STRENGTH(PSI), 21X,
     M        15HCOMPR. STRENGTH, 5X, 11HUNIT WEIGHT, 1H , 1(/), 5X,
     N        7HTENSILE, 12X, 6HSHEAR., 22X, 5H(PSI), 13X, 5H(PCF))
 2180 FORMAT (4X, F8.3, 11X, F8.3, 19X, F8.3, 11X, F7.3)
 2200 FORMAT (1H0, 1(/), 1X, 18(2H--), 2X, 7HSPRINGS, 2X, 17(2H--),
     O        1(/),  4X, 9(2H- ), 9HLOCATIONS, 1X,10(2H- ),
     P        1(/),  4X, 8HLAMBDA R, 5X, 8HLAMBDA L, 5X, 8HLAMBDA V,
     R        5X, 8HLAMBDA W)
 2220 FORMAT ( 4X, 4(F7.5, 6X), /)
 2260 FORMAT (1H , 3X, 32HSTRESS-STRAIN CURVE COORDINATES:,/,4X,33(1H-))
 2261 FORMAT (1H , 3X, 11HSTRESS(PSI), 2X, 5(2X, D11.4))
 2262 FORMAT (1H0, 3X, 13HSTRAIN(IN/IN), 5(2X, D11.4))
 2280 FORMAT (1H0, 1(/), 1X, 17(2H--), 1X, 13HBLAST LOADING, 2X,
     X        16(2H--), 2(/), 1X, 4HTYPE, 3X, 9HTIME CODE, 3X,
     Y        10HUNIF. PEAK, 3X, 9HSINE PEAK, 3X, 9HRISE TIME, 3X,
     Z        13HPEAK DURATION, 3X, 10HDECAY TIME, 1(/), 21X, 5H(PSI),
     ?        8X, 5H(PSI), 13X, 23H( S  E  C  O  N  D  S ),/ )
 2300 FORMAT (1X, A4, 6X, I1, 7X, F8.3, 6X, F8.3, 4X, F8.6, 5X,
     $        F8.6, 7X, F8.6)
 2350 FORMAT (1H1)
C
C-->>   CALCULATE HALF BLOCK DIMENSIONS
C
              A = A2 / TWO
              B = B2 / TWO
              C = C2 / TWO
              L = LF * TWELVE + LI
              H = HF * TWELVE + HI
              MORTWT = MORTWT / (TWELVE ** 3)
      RETURN
      END
```

```
      BLOCK DATA
C
C     ******************************************************************
C     *                                                                *
C     * PURPOSE: INITIALIZE ARRAYS IN COMMON BLOCKS                    *
C     *                                                                *
C     ******************************************************************
C
      IMPLICIT REAL * 8 (A-H , O-Z)
      COMMON /BORDER/ NLB(125), NRB(125)
      COMMON /CURVE / NSTAGE(48,125), STRESS( 5), STRAIN( 5), E(5), NSEG
      COMMON /GLAMKS/ KS(48,125), G(48,6), LAMR, LAML, LAMV, LAMW, AKK
      COMMON /MASLOD/ P(6,125), MASINV(6,125), BLOKWT(125)
      COMMON /RESULT/ FORCE(48,125), UC(6,125), POS(6,125), TIME
      COMMON /STFMAT/ AK(750,6), BK(750,6), CK(750,6), DK(750,6),
     $                EK(750,6), FK(750,6), HK(750,6)
      DOUBLE PRECISION KS, MASINV, LAMR, LAML, LAMV, LAMW
      DATA  FORCE/6000*0.0D00/, KS/6000*0.0D00/, POS/750*0.0D00/
      DATA  P/750*0.0D00/, UC/750*0.0D00/, NSTAGE/6000*1/,NLB,NRB/250*0/
      DATA  AK/4500*0.0D00/, BK/4500*0.0D00/, CK/4500*0.0D00/
      DATA  DK/4500*0.0D00/, EK/4500*0.0D00/, FK/4500*0.0D00/
      DATA  HK/4500*0.0D00/
      END
```

```
      SUBROUTINE CORDS
C
C     ******************************************************************
C     *                                                                *
C     * FUNCTIONS: 1) ASSIGN NUMBERS TO WALL UNITS                     *
C     *            2) CALCULATE COORDINATES OF WALL UNITS              *
C     *                                                                *
C     ******************************************************************
C
      IMPLICIT REAL * 8 (A-H , O-Z)
      COMMON /BLOK  / A, B, C, WJ, TI, TE, EH, ELENTH
      COMMON /BORDER/ NLB(125), NRB(125)
      COMMON /JOINTS/ TXJ, TYJ, TLJ, TRJ, TUJ, TBJ
      COMMON /NOBLOK/ AB(2,125), NB, NBX, NBY, NER, NOR, NBXX, NNBX
      COMMON /RESULT/ FORCE(48,125), UC(6,125), POS(6,125), TIME
      COMMON /WALL  / L, L1, L2, H, H1, ILS, IRS, IUS, IBS, KODE
      DIMENSION  SWITCH(35)
      DOUBLE PRECISION L, L1, L2
      DATA  HALF, TWO, THREE /5.0D-1, 2.0D00, 3.0D00/
C
              A2           = TWO * A
              B2 = TWO * B
              L2           = A2 + TXJ
C-->>  NUMBER OF FULL-LENGTH BLOCKS IN THE X-DIRECTION
              ABX          = (L + TXJ - TLJ - TRJ) / L2 + 0.001
              NBX = ABX
              H2           = TWO*B + TYJ
C-->>  NUMBER OF BLOCK ROWS IN THE Y-DIRECTION
              ABY          = (H + TYJ - TUJ - TBJ) / H2 + 0.001
              NBY = ABY
              NBX1         = NBX + 1
C-->>  CALCULATE NUMBER OF ODD ROWS (NOR) AND NUMBER OF EVEN ROWS (NER)
              N            = (NBY-1) / 2
              NOR          = N + 1
              NER          = NBY - NOR
              NBXX         = NBX
C
C-->>  KODE = 0  INDICATES HORIZONTAL STACK
C
         IF (KODE .EQ. 0) GO TO 210
C
C-->>  CLACULATE LENGTH OF PARTIAL END BLOCK FOR RUNNING BOND
C
              ABX          = NBX
              EXPR         = L - L2*ABX - TLJ - TRJ
              ELENTH       = HALF*EXPR + A
         IF (KODE .EQ. 3 .OR. KODE .EQ. 4) ELENTH = EXPR
              HAFEND      = HALF * ELENTH
              L1           = HAFEND + A + TXJ
              NBX2         = NBX + 2
```

```
           NBX3          = NBX + 3
           XLP           = HAFEND + TLJ
           XRP           = L - HAFEND - TRJ
           POS(1,1) = A + TRJ
           SWITCH(1)     = POS(1,1)
           INC           = 2 * NBX
C-->>   CALCULATE TOTAL NUMBER OF BLOCKS AND X-COORDINATES FOR
C-->>   THE BOTTOM TWO ROWS
        GO TO (100, 110, 130, 130),KODE
  100      NB            = NBX*NCR + NBX1*NER
           NXX           = NBX
        GO TO 120
  110      NB            = NBX1*NCR + NBX*NER
           NXX           = NBX1
  120      NBXX          = NBX
           INC1          = INC + 1
           IN            = INC
           POS(1,NBX1) = XLP
           POS(1,NBX2) = POS(1,NBX1) + L1
        IF (NBY .EQ. 1) GO TO 140
           POS(1,INC1) = XRP
        IF (NBXX .EQ. 1) GO TO 165
        GO TO 140
  130      NB            = NBX1 * NBY
           NBXX          = NBX1
           NXX           = NBX1
           INC           = INC + 2
           INC1          = INC
           IN            = INC - 1
           POS(1,NBX1) = XRP
           POS(1,NBX2) =  XLP
           POS(1,NBX3) = POS(1,NBX2) + L1
           SWITCH(NBX1)  = XRP
  140      ISP = NBXX + 3
        DO 150 I=2,NBX
           POS(1,I) = POS(1,I-1) + L2
           SWITCH(I)      = POS(1,I)
  150   CONTINUE
        IF (NBY .EQ. 1) GO TO 190
        DO 160 I = ISP,INC
           POS(1,I) = POS(1,I-1) + L2
  160   CONTINUE
  165   CONTINUE
        IF (KODE .EQ. 1  .OR.  KODE .EQ. 4) GO TO 190
        DO 170 I=1,NBX1
           POS(1,I) = POS(1,I+NBXX)
  170   CONTINUE
        DO 180 I=NBX2,INC1
           POS(1,I) = SWITCH(I-NBX1)
  180   CONTINUE
  190   CONTINUE
C-->>   DETERMINE BLOCK NUMBER FOR BLOCKS ALONG LEFT AND RIGHT
C-->>   BOUNDARIES
           INN = IN + 1
        DO 191 N = 1,NB,INN
           NLB(N)          = N
           NRB(N+NXX-1) = N + NXX - 1
           NA = NLB(N) + 1
        IF (NRB(N+NXX-1) .EQ. NB) GO TO 192
           NLB(N+NXX)     = N + NXX
           NRB(N+IN)      = N + IN
  191   CONTINUE
  192   CONTINUE
        IF (NBY .EQ. 1) GO TO 210
           INC2 = INC1 + 1
        DO 200 I=1,INC1
        IF (INC2 .GT. NB) GO TO 200
        DO 201 N=INC2,NB,INC1
           POS(1,N) = POS(1,I)
  201   CONTINUE
        IF (KODE .EQ. 2  .OR.  KODE .EQ. 4) GO TO 202
        IF (I .EQ. NBX) INC2 = NBX + INC1
        GO TO 203
  202   CONTINUE
        IF (I .EQ. NBX1) INC2 = NBX1 + INC1
  203      INC2           = INC2 + 1
  200   CONTINUE
        GO TO 250
  210   CONTINUE
C-->>   EXPAND X-COORDINATES IN THE Y-DIRECTION (TWO ROWS AT A TIME)
           INC            = NBX1
           NB             = NBX * NBY
           POS(1,1) = A + TLJ
           N              = 1
           NLB(1)         = 1
           NRB(NBX)       = NBX
  220   CONTINUE
        IF (NBY .EQ. 1) GO TO 235
        DO 230 I=INC, NB, NBX
           POS(1,I) = POS(1,N)
        IF (N .EQ. 1) NLB(I) = I
        IF(N .EQ. NBX) NRB(I) = I
  230   CONTINUE
  235   CONTINUE
        IF (N .EQ. NBX) GO TO 240
           POS(1,N+1) = POS(1,N) + L2
           INC            = INC + 1
           N              = N + 1
        GO TO 220
  240   CONTINUE
```

```
          INC1         = 2 * NBX
  250   CONTINUE
        IF (KODE .EQ. 2) NBXX = NBX1
C
C-->>   CALCULATE MNBX: BLOCK NUMBER FOR LAST BLOCK BEFORE THE UPPER ROW
C
          MNBX = NB - NBXX
        IF (KODE .EQ. 1) MNBX = MNB) + NOR - NER - 1
        IF (KODE .EQ. 2) MNBX = MNB) - NOR + NER + 1
C
C-->>   CALCULATE THE Y-COORDINATES
C
          K            = 1
          NN           = NBX1
        IF (KODE .EQ. 0  .OR.  KODE .EQ. 1) NN = NBX
          MM           = NN
          EXPR         = B + TBJ
        DO 260 LY=1,2
          DO 261 MY=K,NB,INC1
          POS(2,MY)      = EXPR
        IF (NB .EQ. 1) GO TO 265
          J            = MY + 1
           DO 262 I =J,MM
          POS(2,IY)      = POS(2,MY)
  262      CONTINUE
        IF (MBY .EQ. 1) GO TO 265
          MM           = MM + INC1
          EXPR         = EXPR + TWO*H2
  261     CONTINUE
          K            = NN + 1
          MM           = INC1
          EXPR         = THREE * E + TYJ + TBJ
  260   CONTINUE
  265   CONTINUE
C
C-->>   CALCULATE BLOCK EFFECTIVE LENGTH AND WIDTH
C-->>   (ADD HALF A JOINT WIDTH ON EACH SIDE)
C
        IF (KODE .NE. 0) GO TO 310
        DO 300 JB = 1,NB
          HJ = TXJ
          VJ = TYJ
        IF (JB .EQ. NLB(JB)) HJ = TLJ
        IF (JB .EQ. NRB(JB)) HJ = TRJ
        IF (JB .GT. MNBX) VJ = TUJ
        IF (JB .LE. NBXX) VJ = TBJ
          AB(1,JB) = A2 + (TXJ+HJ) / TWO
          AB(2,JB) = B2 + (TYJ+VJ) / TWO
  300   CONTINUE
        GO TO 500
  310   DO 400 J = 1,NB
          PORT = A2
        IF (J .NE. NLB(J)) GO TO 320
          TXX = TLJ
        IF (POS(1,J) .LT. A) PORT = ELENTH
        GO TO 340
  320   IF (J .NE. NRB(J)) GO TO 330
          TXX = TRJ
        IF ((L-POS(1,J)) .LT. A) PORT = ELENTH
        GO TO 340
  330     TXX = TXJ
          TYY = TYJ
  340   IF (J .GT. MNBX) TYY = TLJ
        IF (IUS .EQ. 3) TYY = TYJ
        IF (J .LE. NBXX) TYY = TBJ
          AB(1,J) = PORT + HALF * (TXJ + TXX)
          AB(2,J) = B2 + HALF * (TYJ + TYY)
  400   CONTINUE
  500   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE  LINKEL
C
C     ***************************************************************
C     *                                                             *
C     * FUNCTIONS: 1) CALCULATE MODULI OF ELASTICITY                *
C     *            2) CALCULATE SPRING STIFFNESSES FOR EACH BLOCK    *
C     *            3) CALCULATE MAXIMUM ALLOWABLE FORCES IN SPRINGS  *
C     *                                                             *
C     ***************************************************************
C
      IMPLICIT REAL * 8 (A-H , O-Z)
      COMMON /BLCK  / A, B, C, WJ, TI, TE, EH, ELENTH
      COMMON /BORDER/ NLB(125), NRE(125)
      COMMON /CURVE / NSTAGE(48,125), STRESS( 5), STRAIN( 5), E(5), NSEG
      COMMON /GLAMKS/ KS(48,125), G(48,6), LAMR, LAML, LAMV, LAMW, AKK
      COMMON /INJULT/ COMPHD( 5), COMPBD( 5),TENHED,TENBED,SHRHED,SHRBED
      COMMON /JOINTS/ TXJ, TYJ, TLJ, TRJ, TUJ, TBJ
      COMMON /NOBLOK/ AB(2,125), NB, NBX, NBY, NEP, NOP, NBXX, MNBX
      COMMON /RESULT/ FORCE(40,125), UC(6,125), POS(6,125), TIME
      COMMON /STRNTH/ TSBOND, SSBOND, CSMORT
      COMMON /WALL  / L, L1, L2, H, H2, ILS, IRS, IUS, IBS, KODE
      COMMON /WEIGHT/ WT, MORTWT, IBLCK
      DOUBLE PRECISIO  L, L1, L2, KS, LAMR,LAML, LAMV, LAMW, MORTWT
      DOUBLE PRECISION DSIN
      DATA  XNU /1.5D-1/, PI /3.141592654/
      DATA  ZERO, ONE, TWO /0.0D00, 1.0D00, 2.0D00/
      DATA  SKHY, SKHZ, SKBX, SKBZ /4*0.0D0C/
      DATA  AKH, AKL, AKR, AKB, AKU, AKD, SKUU /7*0.0D00/
C
              A2 = TWO * A
              B2 = TWO * B
              C2 = TWO * C
C-->>    CALCULATE AREA OF CONTACT BETWEEN UNIT AND MORTAR
              WJH = C
              WJB = C
         IF (IBLCK .GE. 2) WJB = WJ
         IF (IBLCK .EQ. 3) WJH = WJ
              BDAREA = WJB * A
              HDAREA = WJH * B
C
C-->>    CALCULATE MODULI OF ELASTICITY: E
C
              E(1) = STRESS(1) / STRAIN(1)
         IF (NSEG .EQ. 1) GO TO 110
         DO 100 I = 2,NSEG
              DSTRS = STRESS(I) - STRESS(I-1)
              DSTRN = STRAIN(I) - STRAIN(I-1)
              E(I) = DSTRS / DSTRN
  100    CONTINUE
  110    CONTINUE
C-->>    CALCULATE SHEAR MODULUS (GG) AND THE EFFECTIVE LENGTH AND HEIGHT
              GG = ONE / (TWO * (ONE+XNU))
              EFFL = L
         IF (IRS .EQ. 3) EFFL = TWO * EFFL - TXJ
              EFFH = H
         IF (IUS .EQ. 3) EFFH = TWO * EFFH - TYJ
C
C-->>    SPRING STIFFNESSES FOR INTERIOR JOINTS
C
C-->>    HEAD JOINTS
C-->>    AXIAL, SHEAR
C
              FACTH = C2 * C2 / (H2 * H2)
              FACTB = C2 * C2 / (L2 * L2)
         IF (NBXX .LE. 2  .OR.  NBY .LE. 2) FACTH = ONE
         IF (NBXX .LE. 2  .OR.  NBY .LE. 2) FACTB = ONE
              FACTH = FACTH * TWO / 3.0D00
              FACTB = FACTB * TWO / 3.0D00
         IF (TXJ .EQ. ZERO) GO TO 111
              YL = L2
              ZLH = L2
              SKHY = HDAREA * GG / YL
              SKHZ = HDAREA * GG / ZLH
              SKHZ = FACTH * SKHZ
              AKH = HDAREA / L2
  111    IF (TLJ .EQ. ZERO) GO TO 112
              AKL = HDAREA / L2
  112    IF (TRJ .EQ. ZERO) GO TO 113
              AKR = HDAREA / L2
C
C-->>    BED JOINTS
C-->>    AXIAL, SHEAR
C
  113    CONTINUE
         IF (TYJ .EQ. ZERO) GO TO 114
              XL = H2
              ZLB  = H2
              SKBX = BDAREA * GG / XL
              SKBZ = BDAREA * GG / ZLB
              SKBZ = FACTB * SKBZ
              AKB = BDAREA / H2
  114    IF (TUJ .EQ. ZERO) GO TO 115
              AKU = BDAREA / H2
  115    IF (TBJ .EQ. ZERO) GO TO 116
              AKD = BDAREA / H2
  116    CONTINUE
C
C-->>    SPRING STIFFNESSES FOR BOUNDARY JOINTS
C-->>    BOUNDARY CODES: 0-FREE, 1-SIMPLE, 2-FIXED, 3-SYMMETRY
C
```

```
            ILS1 = ILS + 1
            IRS1 = IRS + 1
            IUS1 = IUS + 1
            IBS1 = IBS + 1
C-->>   LEFT BOUNDARY
          GO TO (120, 140, 160), ILS1
C-->>   FREE SUPPORT
  120       BAKL = ZERO
            BSKLY = ZERO
            BSKLZ = ZERO
          GO TO 200
C-->>   SIMPLE SUPPORT
  140       BAKL = ZERO
            BSKLY = SKHY
            BSKLZ = SKHZ * (A2 + TLJ) / (A + TLJ)
          GO TO 200
C-->>   FIXED SUPPORT
  160       BAKL = 1.0003
            BSKLY = 1.0003
            BSKLZ = 1.0003
C-->>   RIGHT BOUNDARY
  200     GO TO (220, 240, 260, 280), IRS1
C-->>   FREE SUPPORT
  220       BAKR = ZERO
            BSKRY = ZERO
            BSKRZ = ZERO
          GO TO 300
C-->>   SIMPLE SUPPORT
  240       BAKR = ZERO
            BSKRY = SKHY
            BSKRZ = SKHZ * (A2 + TRJ) / (A + TRJ)
          GO TO 300
C-->>   FIXED SUPPORT
  260       BAKR = 1.0003
            BSKRY = 1.0003
            BSKRZ = 1.0003
          GO TO 300
C-->>   SYMMETRY
  280       BAKR = AKH
            BSKRY = SKHY
            BSKRZ = SKHZ
C-->>   UPPER BOUNDARY
  300     GO TO (320, 340, 360, 380), IUS1
C-->>   FREE SUPPORT
  320       BAKU = ZERO
            BSKUX = ZERO
            BSKUZ = ZERO
          GO TO 400
C-->>   SIMPLE SUPPORT
  340       BAKU = ZERO
            BSKUX = SKBX
            BSKUZ = SKBZ * (B2+TUJ) / (B+TUJ)
          GO TO 400
C-->>   FIXED SUPPORT
  360       BAKU = 1.0003
            BSKUX = 1.0003
            BSKUZ = 1.0003
          GO TO 400
C-->>   SYMMETRY
  380       BAKU = AKB
            BSKUX = SKBX
            BSKUZ = SKBZ
C-->>   LOWER BOUNDARY
  400     GO TO (420, 440, 460), IBS1
C-->>   FREE SUPPORT
  420       BAKB = ZERO
            BSKBX = ZERO
            BSKBZ = ZERO
          GO TO 500
C-->>   SIMPLE SUPPORT
  440       BAKB = ZERO
            BSKBX = SKBX
            BSKBZ = SKBZ * (B2 + TBJ) / (B+TBJ)
          GO TO 500
C-->>   FIXED SUPPORT
  460       BAKB = 1.0003
            BSKBX = 1.0003
            BSKBZ = 1.0003
  500     CONTINUE
            XX = TLJ + A/TWO - AB(1,1)
            YY = TBJ + B/TWO - AB(2,1)
          DO 800 K = 1,NB
C
C-->>   SPRING STIFFNESSES FOR HEAD JOINTS
C
            KL = 45
            M = 9
          IF (K .NE. NLB(K)) GO TO 502
            YY = YY + AB(2,K)
            Y = YY
            YC = Y + B/TWO
  502     CONTINUE
          DO 600 N = 1,2
            BSKLL = BSKLZ
            SKHH = SKHZ
          IF (K .NE. NLB(K)) GO TO 520
          IF (KODE .EQ. 0) GO TO 510
          IF (ILS .EQ. 0) GO TO 505
            AR = A + TLJ - POS(1,K)
          IF (AR .LT. ZERO) AR = - AR
```

```
      IF (AR .LT. 0.1000) GO TO 510
C
C-->>   ADJUST SHEAR STIFFNESS FOR LEFT BOUNDARY (FOR PARTIAL UNITS)
C
           BSKLL = SKHZ * (A2+TLJ) / (ELENTH/TWO + TLJ)
  505   CONTINUE
C
C-->>   ADJUST SHEAR STIFFNESS FOR HEAD JOINTS OF PARTIAL BLOCKS IN
C-->>   THE RUNNING BOND PATTERN
C
           SKHH = SKHZ * L2 / L1
           SKLLB = BSKLL
           SHHK = SKHH
      IF (ILS .EQ. 2) SKLLB = BSKLZ
  510   CONTINUE
      IF (ILS .NE. 1) GO TO 515
      IF (NBY .LE. 2) GO TO 515
           BSKLL = BSKLL = DSIN(PI*Y/EFFH)
     $           /(ONE - AB(1,K)*PI*DSIN(PI*YC/EFFH)/(TWO*EFFL))
C
C-->>   ADJUST SHEAR STIFFNESS FOR LEFT BOUNDARY ACCORDING TO
C-->>   SINUSOIDAL VARIATION (FOR SIMPLE SUPPORT ONLY)
C
  515   CONTINUE
C
C-->>   ASSIGN SPRING STIFF. FOR LEFT HEAD JOINTS OF BLOCKS ALONG
C-->>   THE LEFT BOUNDARY
C
           KS(KL  ,K) = BSKLL
           KS(KL-1,K) = BSKLY
           KS(KL-2,K) = BAKL
  520   CONTINUE
      IF (NBXX .EQ. 1) GO TO 525
      IF (K .EQ. NB) GO TO 525
      IF ((K+1) .EQ. NRB(K+1)  .AND.  AE(1,K+1) .LT. A2) SKHH = SKHZ
     $     * L2 / L1
  525   CONTINUE
C
C-->>   ASSIGN SPRING STIFFNESSES FOR HEAD JOINTS
C
           KS(M  ,K) = SKHH
           KS(M-1,K) = SKHY
           KS(M-2,K) = AKH
C
C-->>   ADJUST SHEAR STIFFNESSES FOR RIGHT BOUNDARY
C
  530      BSKRR = BSKRZ
      IF (K .NE. NRB(K)) GO TO 540
      IF (KODE .EQ. 0) GO TO 535
      IF (IRS .EQ. 0) GO TO 532
           AL = L - POS(1,K)- A - TRJ
      IF (AL .LT. ZERO) AL = - AL
      IF (AL .GT. 0.10000)BSKRR = SKHZ * (A2 + TRJ) /
     $                                   (ELENTH/TWO + TRJ)
  532   CONTINUE
           SKRRB = BSKRR
      IF (IRS .EQ. 2) SKRRB = BSKRZ
  535   CONTINUE
C
C-->>   ASSIGN SPRING STIFF. FOR RIGHT HEAD JOINTS OF BLOCKS ALONG
C-->>   THE RIGHT BOUNDARY
C
           KS(M  ,K) = BSKRR
           KS(M-1,K) = BSKRY
           KS(M-2,K) = BAKR
  540      KS(M+12,K) = KS(M   ,K)
           KS(M+11,K) = KS(M-1,K)
           KS(M+10,K) = KS(M-2,K)
      IF (K .EQ. NLB(K)) GO TO 560
C
C-->>   SET STIFFNESSES FOR SPRINGS ON LEFT HEAD JOINT OF A BLOCK
C-->>   EQUAL TO THOSE ON THE RIGHT OF THE PREVIOUS BLOCK
C
           KS(KL  ,K) = KS(M   ,K-1)
           KS(KL-1,K) = KS(M-1,K-1)
           KS(KL-2,K) = KS(M-2,K-1)
  560      KS(KL-12,K) = KS(KL  ,K)
           KS(KL-13,K) = KS(KL-1,K)
           KS(KL-14,K) = KS(KL-2,K)
           KL = 42
           M = 3
           Y = Y + B
  600   CONTINUE
C-->>   SPRING STIFFNESSES FOR BED JOINTS
      IF (NBY .EQ. 1) GO TO 800
           MBXO = NBX
           MBX12 = NBX + 1
      IF (K .NE. NLB(K)) GO TO 610
           MBX34 = NBX + 1
      IF (AB(1,K) .LT. A2) MBX34 = NBX + 2
           M34 = MBX34
  610   CONTINUE
           KL = 48
           M = 39
           I = 1
           XX = XX + AB(1,K)
           X = XX
           XC = X + A/TWO
      DO 700 N = 1,2
      IF (K .GT. NBXX) GO TO 620
```

```
      IF (AB(1,K) .GE. A2) GO TO 615
      IF (N .EQ. 1  .AND.  K .EQ. NLB(K)) GO TO 630
      IF (N .EQ. 2  .AND.  K .EQ. NRB(K)) GO TO 630
  615 CONTINUE
C
C-->>   CALCULATE FACTORS FOR SINUSOIDAL VARIATION OF SHEAR
C-->>   STIFFNESSES (FOR SIMPLE SUPPORTS)
C
          SUP = ONE
      IF (IBS .NE. 1) GO TO 616
      IF (NBXX .LE. 2) GO TO 616
          SUP = DSIN(PI*X/EFFL)
     a        /(ONE-AB(2,K)*PI*DSIN(PI*XC/EFFL)/(TWO*EFFH))
C
C-->>   ASSIGN STIFF. FOR BOTTOM SPRINGS OF BLOCKS ALONG LOWER BOUNDARY
C
  616     KS(KL  ,K) = BSKBZ * SUP
          KS(KL-1,K) = BAKB
          KS(KL-2,K) = BSKBX
  620 CONTINUE
C
C-->>   ADJUST SHEAR STIFF. FOR BED JOINT SPRINGS BETWEEN FULL BLOCKS
C-->>   OF THE RUNNING BOND PATTERN
C
          SKBB = SKBZ
      IF (KODE .EQ. 0) GO TO 627
      IF (AB(1,K) .GE. A2) GO TO 625
      IF (K .EQ. NLB(K)  .AND.  N .EQ. 1) GO TO 630
      IF (K .EQ. NRB(K)  .AND.  N .EQ. 2) GO TO 630
          SKBB = SKBZ*H2 / ((A-ELENTH/TWO)**2 + H2*H2)**0.5
          SBBK = SKBB
      GO TO 627
  625 CONTINUE
          SKBB = SKBZ*H2 / ((L2/TWO)**2 + H2*H2)**0.5
          SBBK = SKBB
      IF (K .EQ. NLB(K) .AND. N .EQ. 1) SKBB = SKBZ*H2 / ((A-ELENTH
     $    / TWO)**2 + H2 * H2)**0.5
      IF (K .EQ. NRB(K)  .AND.  N .EQ. 2) SKBB = SKBZ * H2 /
     $ ((A-ELENTH/TWO)**2 + H2*H2)**0.5
  627 CONTINUE
C
C-->>   ASSIGN STIFFNESSES FOR BED JOINT SPRINGS
C
          KS(M  ,K) = SKBB
          KS(M-1,K) = AKB
          KS(M-2,K) = SKBX
  630 IF (K .LE. MNBX) GO TO 640
          SKUU = BSKUZ
      IF (IUS .EQ. 3) SKUU = SKBB
      IF (AB(1,K) .GE. A2) GO TO 635
      IF (N .EQ. 1  .AND.  K .EQ. NLB(K)) GO TO 640
      IF (N .EQ. 2  .AND.  K .EQ. NRB(K)) GO TO 640
  635 CONTINUE
C-->>  ASSIGN STIFF. FOR BED JOINT SPRINGS ALONG UPPER BOUNDARY
          KS(M  ,K) = SKUU
          KS(M-1,K) = BAKU
          KS(M-2,K) = BSKUX
  640     MI = M-I*12
          KI = KL-I*12
          KS(MI  ,K) = KS(M  ,K)
          KS(MI-1,K) = KS(M-1,K)
          KS(MI-2,K) = KS(M-2,K)
      IF (K .LE. MBXX) GO TO 660
      IF (KODE .EQ. 0) MBX = MBX0
      IF (KODE .EQ. 1  .OR.  KODE .EQ. 2) MBX = MBX12
      IF (KODE .GT. 2) MBX = MBX34
      IF ((K-MBX) .EQ. 0) GO TO 660
          MM = M
      IF (KODE .EQ. 0) GO TO 650
      IF (N .EQ. 1) MM = 6
      IF (N .EQ. 2) MM = 39
  650 CONTINUE
      IF (K .NE. NLB(K)) GO TO 655
      IF (AB(1,K) .LT. A2  .AND.  N .EQ. 1) GO TO 665
      GO TO 656
  655 IF (K .NE. NRB(K)) GO TO 656
      IF (AB(1,K) .LT. A2  .AND.  N .EQ. 2) GO TO 665
  656 CONTINUE
C
C-->>  SET STIFF. ON BOTT. OF A BLOCK EQUAL THOSE ON TOP OF BLOCK BELOW
C
          KS(KL  ,K) = KS(MM  ,K-MBX)
          KS(KL-1,K) = KS(MM-1,K-MBX)
          KS(KL-2,K) = KS(MM-2,K-MBX)
  660     KS(KI  ,K) = KS(KL  ,K)
          KS(KI-1,K) = KS(KL-1,K)
          KS(KI-2,K) = KS(KL-2,K)
  665 CONTINUE
          I = - I
          KL = 12
          M = 6
          MBX12 = NBX
          MBX34 = M34 - 1
      IF (N .EQ. 2) MBX34 = M34
          X = X + A
  700 CONTINUE
  800 CONTINUE
    PRINT 1900
C
C-->>  MAXIMUM ALLOWABLE STRESSES
```

```
C
          DO 900  I = 1,NSEG
                COMPHD(I) = STRESS(I) * HDAREA
                COMPBD(I) = STRESS(I) * BDAREA
      PRINT 1950, I, E(I)
  900     CONTINUE
                TENBED = TSBOND * BDAREA
                TENHED = TSBOND * HDAREA
                SHRBED = SSBOND * BDAREA
                SHRHED = SSBOND * HDAREA
C-->>   MULTIPLY STIFFNESSES BY E(1) FOR PRINTING PURPOSES
                AKH   = AKH   * E(1)
                SKHY  = SKHY  * E(1)
                SKHZ  = SKHZ  * E(1)
                AKB   = AKB   * E(1)
                SKBX  = SKBX  * E(1)
                SKBZ  = SKBZ  * E(1)
                BAKL  = BAKL  * E(1)
                BSKLY = BSKLY * E(1)
                BSKLZ = BSKLZ * E(1)
                BAKR  = BAKR  * E(1)
                BSKRY = BSKRY * E(1)
                BSKRZ = BSKRZ * E(1)
                BAKU  = BAKU  * E(1)
                BSKUX = BSKUX * E(1)
                SKUU  = SKUU  * E(1)
                BAKB  = BAKB  * E(1)
                BSKBX = BSKBX * E(1)
                BSKBZ = BSKBZ * E(1)
          IF (KODE .EQ. 0) GO TO 950
                SHHK  = SHHK  * E(1)
                SBKB  = SBKB  * E(1)
                SBBK  = SBBK  * E(1)
                SKLLB = SKLLB * E(1)
                SKRRB = SKRRB * E(1)
  950     CONTINUE
      PRINT 2000
      PRINT 2010
          IF (KODE .NE. 0) GO TO 1000
      PRINT 2020, AKH, AKB
      PRINT 2030, SKHY, SKBX
      PRINT 2040, SKHZ, SKBZ
          GO TO 1100
 1000 PRINT 2050, AKH, AKB, AKH, AKB
      PRINT 2060, SKHY, SKBX, SKHY, SKBX
      PRINT 2070, SKHZ, SBKB, SHHK, SBBK
 1100 PRINT 2080
          IF (KODE .NE. 0) GO TO 1200
      PRINT 2090, BAKL, BAKR, BAKU, BAKB
      PRINT 2100, BSKLY, BSKRY, BSKUX, BSKBX
      PRINT 2110, BSKLZ, BSKRZ, SKUU, BSKBZ
          GO TO 1300
 1200 PRINT 2120, BAKL, BAKR, BAKU, BAKB, BAKL, BAKR
      PRINT 2130, BSKLY, BSKRY, BSKUX, BSKBX, BSKLY, BSKRY
          IF (IUS .EQ. 3) SKUU = SBKB
      PRINT 2140, BSKLZ, BSKRZ, SKUU, BSKBZ, SKLLB, SKRRB
 1300 PRINT 2150
 1900 FORMAT (1H0, 2(/),1X, 21HMODULI OF ELASTICITY:, /, 1X,21(1H-), /)
 1950 FORMAT (1H0, 11HSEGMENT NO., I2, 3X, D20.13)
 2000 FORMAT (1H0, 1(/), 44(1H*), 3X, 30HS P R I N G   S T I F F N E S ,
     1        5HS E S, 3X, 44(1H*))
 2010 FORMAT (1H0, 2(/), 33X, 15(1H-), 1X, 15HINTERIOR JOINTS, 1X,
     2        19(1H-),/, 27X,  5(1H-), 1X, 11HFULL BLOCKS, 1X,  4(1H-),
     3        16X, 4(1H-), 1X, 14HPORTION BLOCKS, 1X, 4(1H-), /,
     4        27X, 4HHEAD,15X, 3HBED, 16X, 4HHEAD, 17X, 3HBED, 1(/))
 2020 FORMAT (1H ,9HA X I A L, 9X, 2(2X, D17.10), 13X,14HNOT APPLICABLE)
 2030 FORMAT (1H ,14HIN-PLANE SHEAR, 4X, 2(2X, D17.10), 13X,
     5        14HNOT APPLICABLE)
 2040 FORMAT (1H , 14HZ-DIREC. SHEAR, 4X, 2(2X, D17.10), 13X,
     6        14HNOT APPLICABLE)
 2050 FORMAT (1H , 9HA X I A L, 9X, 4(2X, D17.10))
 2060 FORMAT (1H , 14HIN-PLANE SHEAR, 4X, 4(2X, D17.10))
 2070 FORMAT (1H , 14HZ-DIREC. SHEAR, 4X, 4(2X, D17.10))
 2080 FORMAT (1H0, 2(/), 52X, 35(1H-), 1X, 15HBOUNDARY JOINTS, 1X,
     7        17(1H-), /, 27X, 24(1H-), 1X, 11HFULL BLOCKS, 1X, 25(1H-),
     8        14X, 4(1H-), 14HPORTION BLOCKS, 1X, 4(1H-), /, 27X,
     9        4HLEFT, 15X, 5HRIGHT, 14X, 5HUPPER, 14X, 5HLOWER,
     $        14X, 4HLEFT, 15X, 5HRIGHT, 1(/))
 2090 FORMAT (1H ,9HA X I A L, 9X, 4(2X,D17.10), 13X, 14HNOT APPLICABLE)
 2100 FORMAT (1H , 14HIN-PLANE SHEAR, 4X, 4(2X, D17.10), 13X,
     A        14HNOT APPLICABLE)
 2110 FORMAT (1H , 14HZ-DIREC. SHEAR, 4X, 4(2X, D17.10), 13X,
     B        14HNOT APPLICABLE)
 2120 FORMAT (1H , 9HA X I A L, 9X, 6(2X, D17.10))
 2130 FORMAT (1H , 14HIN-PLANE SHEAR, 4X, 6(2X, D17.10))
 2140 FORMAT (1H , 14HZ-DIREC. SHEAR, 4X, 6(2X, D17.10))
 2150 FORMAT (1H1)
      RETURN
      END
```

```
      SUBROUTINE SOLVER
C
C     ****************************************************************
C     *                                                              *
C     * FUNCTIONS: 1) FORMULATE AND SOLVE EQUATIONS OF MOTION        *
C     *            2) TRACE CRACK DEVELOPMENT AND NON-LINEAR BAHAVIOR *
C     *                                                              *
C     * OTHER SUBROUTINES CALLED: "LOAD", "MASS", "STIF", "FORCES"    *
C     *                                                              *
C     ****************************************************************
C
      IMPLICIT REAL * 8 (A-H , O-Z)
      COMMON /BLAST / DPEAK, SPEAK, RTIME, CTIME, DTIME, TYPE,ILOAD,KDYN
      COMMON /BLCK  / A, B, C, AJ, TI, TE, EH, ELENTH
      COMMON /BORDER/ NLB(125), NRB(125)
      COMMON /CURVE / NSTAGE(48,125), STRESS( 5), STRAIN( 5), E(5), NSEG
      COMMON /INJULT/ COMPHD( 5), COMPBD( 5),TENHED,TENBED,SHRHED,SHRBED
      COMMON /JOINTS/ TXJ, TYJ, TLJ, TRJ, TUJ, TBJ
      COMMON /MASLOD/ P(6,125), MASINV(6,125), BLCKWT(125)
      COMMON /NOBLCK/ AB(2,125), NB, NBX, NBY, NER, NCR, NBXX, MNBX
      COMMON /RESULT/ FORCE(48,125), UC(6,125), PCS(6,125), TIME
      COMMON /STFMAT/ AK(750,6), BK(750,6), CK(750,6), DK(750,5),
     $                EK(750,6), FK(750,6), HK(750,6)
      COMMON /WALL  / L, L1, L2, H, H2, ILS, IRS, IUS, IBS, KODE
      DIMENSION ACC(6,125), VEL(6,125), KU(6,125)
      DIMENSION UCT(6,125), VCT(6,125), ACE(6,125), ASSUME(6,125)
      DOUBLE PRECISION L, L1, L2, KU, KS, MASINV, DASS
      DATA  ITD, IRC /2*0/
      DATA  KU/750*0.0D00/, BETA/6.0D00/
      DATA  ACC/750*0.0D00/, VEL/750*0.0D00/
      DATA  ZERO/0.0D00/, HALF/5.0D-1/, ONE/1.0D00/, TWO/2.0D00/
C
C-->>   SET TIME TO ZERO
C
               TIME      = ZERO
      PRINT 5000, TIME
C
C-->>   CALL SUBROUTINES LOAD, MASS, AND STIFF AT TIME = 0
C
      CALL LOAD
      CALL MASS
      CALL STIF
C
C-->>   CALCULATE TIME INCREMENT: DT
C
               L = 3
               M = 4
               N = 5
               N3 = 1
          IF (IBS .NE. 2  .AND.  ILS .NE. 2) GO TO 50
               N3 = NBXX + 2
               L = N3 * 6 + 3
               M = L + 1
               N = L + 2
 50       CONTINUE
               SMALL3 = AK(L,3) * MASINV(3,N3)
               SMALL4 = AK(M,4) * MASINV(4,N3)
               SMALL5 = AK(N,5) * MASINV(5,N3)
          DO 100 I = 1,NB
               K3 = 6 * I - 3
               K4 = 6 * I - 2
               K5 = 6 * I - 1
               TINY3 = AK(K3,3) * MASINV(3,I)
          IF (TINY3 .LT. SMALL3) GO TO 90
               SMALL3 = TINY3
               II = I
 90            TINY4 = AK(K4,4) * MASINV(4,I)
          IF (TINY4 .LT. SMALL4) GO TO 110
               SMALL4 = TINY4
               I4 = I
 110           TINY5 = AK(K5,5) * MASINV(5,I)
          IF (TINY5 .LT. SMALL5) GO TO 100
               SMALL5 = TINY5
               I5 = I
 100      CONTINUE
               DETE = (BETA/SMALL3)**0.5
               DET4 = (BETA/SMALL4)**0.5
               DET5 = (BETA/SMALL5)**0.5
               NK = 0
               DET = DETE
          IF (DET4 .LT. DET) DET = DET4
          IF (DET5 .LT. DET) DET = DET5
               DUM = DET
 120      CONTINUE
               NK = NK + 1
               DUM = DUM * 10.0D00
               NUM = DUM
          IF (NUM .NE. 0) GO TO 155
          GO TO 120
 155           AD = NUM
               ADT = AD / 4.0D00
               DT = ADT * ((10.0D00)**(-NK))
      PRINT 6000
C
               A2 = TWO * A
               B2 = TWO * B
          DO 200 J=1,NB
      PRINT 6010, J, (UC(IB,J), IB=1,6), (VEL(IB,J), IB=1,6)
               PCS(2,J) = PCS(2,J) + UC(2,J)
          DO 230 I=1,6
```

```
C
C-->>   RESTRAIN IMPROPER ROTATION ALONG SUPPORT FOR SIMPLE SUPPORT
C
        IF (I .LE. 3  .OR.  I .EQ. 5) GO TO 191
        IF (ILS .EQ. 1  .AND.  J .EQ. NLE(J)) GO TO 230
        IF (IRS .EQ. 1  .AND.  J .EQ. NRB(J)) GO TO 230
  191   IF (I .LE. 4) GO TO 192
        IF (IBS .EQ. 1  .AND.  J .LE. NBXX) GO TO 230
        IF (IUS .EQ. 1  .AND.  J .GT. MNBX) GO TO 230
  192   CONTINUE
        IF (KDYN .EQ. 0  .AND.  I .EQ. 2) GO TO 225
        IF (KDYN .EQ. 0  .AND.  I .EQ. 6) GO TO 225
C
C-->>   CALCULATE ACCELERATIONS AT TIME = 0
C
             ACC(I,J) = MASINV(I,J)*(P(I,J)-KU(I,J))
        GO TO 200
  225        ACC(I,J) = ZERO
  230   CONTINUE
  200   CONTINUE
C
C-->>   CALL SUBROUTINE "FORCES" TO INITIATE FORCES AT TIME = 0
C
      CALL FORCES
        DO 250 J = 1,NE
        DO 250 I = 1,6
             UC(I,J) = ZERO
  250   CONTINUE
C
C-->>   DETERMINE NUMBER OF TIME STEPS (INCREMENTS)
C
             STEP      = (RTIME + CTIME + DTIME) / DT + 1.
             NSTEP     = STEP
             NSTEP = NSTEP + 150
             TENDT = 10.0000 * DT
             PPNT = TENDT
             CHEK = ZERO
             NITER  = 3 * NB
        DO 800 ITIME = 2,NSTEP
C
C-->>   INCREMENT TIME BY DT
C
             TIME = TIME + DT
C
C-->>   ADJUST TIME IF ABRUPT CHANGE IN SLOPE OF FORCING
C-->>   FUNCTION OCCURS WITHIN DT
C
        IF (ITD .EQ. 0) GO TO 260
             TIME = TIME - TD
             ITD = 0
        GO TO 300
  260   IF (IRC .EQ. 0) GO TO 300
             TIME = TIME - RC
             IRC = 0
  300   CONTINUE
C
C-->>   CALL SUBROUTINE "LOAD" TO CALCULATE DYNAMIC LOAD ON
C-->>   EACH BLOCK AT GIVEN TIME
C
      CALL LOAD
             ITER = 1
C
C-->>   INITIALIZE CENTROIDAL DISPL. (UCT), VEL. (VCT), AND ACCL. (ACE)
C
        DO 400 JB = 1,NB
             P(2,JB) = ZERO
        DO 400 I = 1,6
             UCT(I,JB) = UC (I,JB)
             VCT(I,JB) = VEL(I,JB)
             ACE(I,JB) = ACC(I,JB)
  400   CONTINUE
  430   CONTINUE
        DO 450 JB = 1,NB
        DO 440 I = 1,6
        IF (I .LE. 3  .OR.  I .EQ. 5) GO TO 431
        IF (ILS .EQ. 1  .AND.  JB .EQ. NLE(JB)) GO TO 440
        IF (IRS .EQ. 1  .AND.  JB .EQ. NRB(JB)) GO TO 440
  431   IF (I .LE. 4) GO TO 432
        IF (IBS .EQ. 1  .AND.  JB .LE. NBXX) GO TO 440
        IF (IUS .EQ. 1  .AND.  JB .GT. MNBX) GO TO 440
  432   CONTINUE
C
C-->>   CALCULATE DISPL. (UC) & VELOCITY (VEL) BASED ON ASSUMED ACCELER.
C
             ASSUME(I,JB) = ACC(I,JB)
             UC(I,JB) = UCT(I,JB) + DT*VCT(I,JB) + (HALF - ONE/BETA)
     1                  *(DT*DT)*ACE(I,JB) + (DT*DT) * ACC(I,JB)/BETA
             VEL(I,JB) = VCT(I,JB) + HALF*DT*(ACC(I,JB) + ACE(I,JB))
  440   CONTINUE
  450   CONTINUE
C
C-->>   MULTIPLY STIFFNESSES BY DISPL. VECTORS TO OBTAIN RESISTANCE
C-->>   VECTOR KU FOR EACH BLOCK
C
             LN = MNBX + 1
             INDEX = -1
             INDX  = 1
      DO 1888 J = 1,NB
C
C-->>   MULTIPLY KA BY UC FOR BLOCK A
```

```
C
      DO 150 IK = 1,6
  150    KU(IK,J) = ZERO
         NA = 6 * (J-1)
      DO 1055 IA = 1,6
         MA = IA + NA
      DO 1055 JA = 1,6
         KU(IA,J) = KU(IA,J) + AK(MA,JA) * UC(JA,J)
 1055 CONTINUE
      IF (IRS .EQ. 3  .AND.  J .EQ. NRB(J)) GO TO 1060
      IF (J .EQ. NRB(J)) GO TO 1110
         JP1     = J + 1
         SS = ONE
      GO TO 1070
 1060    JP1 = J
         SS = - ONE
 1070 CONTINUE
C
C-->>  MULTIPLY KB BY UC OF BLOCK B AND SUBTRACT FROM KU
C
      DO 1100 IA = 1,6
         MA = IA + NA
      DO 1100 JA = 1,6
         AS = ONE
      IF (JA .EQ. 5) AS = SS
         KU(IA,J) = KU(IA,J) - BK(MA,JA) * UC(JA,JP1) * AS
 1100 CONTINUE
 1110 IF (IUS .NE. 3) GO TO 1120
      IF (J .LE. MNBX) GO TO 1150
         JPX = J
         SS = - ONE
      GO TO 1160
 1120 IF (J .EQ. LN) GO TO 1475
      IF (J .GT. LN .AND. J .LE. NB) GO TO 1300
 1150    JPX     = J + NBX
         SS = ONE
      IF (NBY .EQ. 1) GO TO 1275
 1160 IF ( KODE .NE. 0) GO TO 1210
C
C-->>  MULTIPLY KC BY UC OF BLOCK C AND SUBTRACT FROM KU (FOR H. STACK)
C
      DO 1200 IA = 1,6
         MA = IA + NA
      DO 1200 JA = 1,6
         AS = ONE
      IF (JA .EQ. 4) AS = SS
         KU(IA,J) = KU(IA,J) - CK(MA,JA) * UC(JA,JPX) * AS
 1200 CONTINUE
      GO TO 1275
 1210 CONTINUE
         JPX1 = JPX + 1
      IF (J .EQ. NLB(J)) INDEX = - INDEX
      IF (KODE .NE. 3) GO TO 1212
      IF (INDEX) 1215, 1215, 1220
 1212 IF (KODE .NE. 4) GO TO 1220
      IF (INDEX) 1220, 1220, 1215
 1215    JPX1 = JPX + 2
         JPX  = JPX + 1
         SS = ONE
 1220 IF (AB(1,J) .LT. A2  .AND.  J .EQ. NRB(J)) GO TO 1230
C
C-->>  MULTIPLY KC BY UC OF BLOCK C AND SUBTRACT FROM KU (FOR R. BOND)
C
      IF (IUS .NE. 3) GO TO 1222
      IF (J .LE. MNBX) GO TO 1222
         JPX1 = J
         JPX  = J
         SS = - ONE
 1222 CONTINUE
      DO 1225 IA = 1,6
         MA = IA + NA
      DO 1225 JA = 1,6
         AS = ONE
      IF (JA .EQ. 4) AS = SS
         KU(IA,J) = KU(IA,J) - CK(MA,JA) * UC(JA,JPX1) * AS
 1225 CONTINUE
      IF (AB(1,J) .LT. A2  .AND.  J .EQ. NLB(J)) GO TO 1275
C
C-->>  MULTIPLY KE BY UC OF BLOCK E AND SUBTRACT FROM KU (FOR R. BOND)
C
 1230 CONTINUE
      DO 1250 IA = 1,6
         MA = IA + NA
      DO 1250 JA = 1,6
         AS = ONE
      IF (JA .EQ. 4) AS = SS
         KU(IA,J) = KU(IA,J) - EK(MA,JA) * UC(JA,JPX) * AS
 1250 CONTINUE
 1275 CONTINUE
      IF (J .EQ. 1) GO TO 1650
      IF (J .EQ. NLB(J)) GO TO 1475
C
C-->>  MULTIPLY KD BY UC OF BLOCK D AND SUBTRACT FROM KU
C
 1300    JM1     = J - 1
      DO 1350 IA = 1,6
         MA = IA + NA
      DO 1350 JA = 1,6
         KU(IA,J) = KU(IA,J) - DK(MA,JA) * UC(JA,JM1)
 1350 CONTINUE
```

```
      IF ( J.LE. NBXX) GO TO 1650
 1475      JMX = J - NBX
      IF (KODE .NE. 0) GO TO 1500
      IF (NBY .EQ. 1) GO TO 1575
C
C-->>  MULTIPLY KE BY UC OF BLOCK E AND SUBTRACT FROM KU (FOR H. STACK)
C
      DO 1600 IA = 1,6
           MA = IA + NA
      DO 1600 JA = 1,6
           KU(IA,J) = KU(IA,J) - EK(MA,JA) * UC(JA,JMX)
 1600 CONTINUE
      GO TO 1575
 1500 CONTINUE
           JMX1 = JMX - 1
      IF (J .EQ. NLB(J)) INDX = - INDX
      IF (KODE .NE. 3) GO TO 1510
      IF (INDX)  1525, 1525, 1520
 1510 IF (KODE .NE. 4) GO TO 1525
      IF (INDX)  1520, 1520, 1525
 1520      JMX1 = JMX - 2
           JMX  = JMX - 1
 1525 IF (AB(1,J) .LT. A2   .AND.  J .EQ. NLB(J)) GO TO 1550
C
C-->>  MULTIPLY KF BY UC OF BLOCK F AND SUBTRACT FROM KU (FOR R. BEND)
C
      DO 1540 IA = 1,6
           MA = IA + NA
      DO 1540 JA = 1,6
           KU(IA,J) = KU(IA,J) - FK(MA,JA) * UC(JA,JMX1)
 1540 CONTINUE
      IF (AB(1,J) .LT. A2   .AND.  J .EQ. NRB(J)) GO TO 1575
C
C-->>  MULTIPLY KH BY UC OF BLOCK H AND SUBTRACT FROM KU (FOR R. BEND)
C
 1550 CONTINUE
      DO 1560 IA = 1,6
           MA = IA + NA
      DO 1560 JA = 1,6
           KU(IA,J) = KU(IA,J) - H(MA,JA) * UC(JA,JMX)
 1560 CONTINUE
 1575 CONTINUE
 1650 CONTINUE
 1888 CONTINUE
C
C-->>  SOLVE EQUATIONS OF MOTION FOR ACCELERATIONS
C
      DO 480 JB = 1,NB
      DO 460 I = 1,6
      IF (I .LE. 3  .OR.  I .EQ. 5) GO TO 451
      IF (ILS .EQ. 1  .AND.  JB .EQ. NLB(JB)) GO TO 460
      IF (IRS .EQ. 1  .AND.  JB .EQ. NRB(JB)) GO TO 460
  451 IF (I .LE. 4) GO TO 452
      IF (IBS .EQ. 1  .AND.  JB .LE. NBXX) GO TO 460
      IF (IUS .EQ. 1  .AND.  JB .GT. MNBX) GO TO 460
  452 CONTINUE
           ACC(I,JB)= MASINV(I,JB)*(P(I,JB)-KU(I,JB))
  460 CONTINUE
  480 CONTINUE
      DO 485 JB = 1,NB
C
C-->>  SET "TOLRNS" VALUE FOR ALLOWABLE ERROR
C
           TOLRNS = (5.0D-4) * MASINV(3,JB) * P(3,JB)
      IF (TOLRNS .LT. ZERO) TOLRNS = - TOLRNS
      IF (P(3,JB) .EQ. ZERO) TOLRNS = 0.05
      DO 484 I = 1,6
      IF (I .LE. 3  .OR.  I .EQ. 5) GO TO 481
      IF (ILS .EQ. 1  .AND.  JB .EQ. NLB(JB)) GO TO 484
      IF (IRS .EQ. 1  .AND.  JB .EQ. NRB(JB)) GO TO 484
  481 IF (I .LE. 4) GO TO 482
      IF (IBS .EQ. 1  .AND.  JB .LE. NBXX) GO TO 484
      IF (IUS .EQ. 1  .AND.  JB .GT. MNBX) GO TO 484
  482 CONTINUE
C
C-->>  CHECK IF CALCULATED ACCELER. IS WITHIN ALLOWABLE
C-->>  ERROR OF ASSUMED ACCELERATION
C
           DIFF = ASSUME(I,JB) - ACC(I,JB)
           DIFA = DIFF
      IF (DIFF .LT. ZERO) DIFA = - DIFF
      IF (DIFA .LE. TOLRNS) GO TO 484
           ITER = ITER + 1
      IF (ITER .LE.NITER) GO TO 430
    PRINT 4000, NITER
    STOP
  484 CONTINUE
  485 CONTINUE
C
C-->>  SELECT TIME INTERVAL FOR PRINTING OUTPUT
C
      IF (TIME .LE. TENDT) GO TO 490
           TIMP = (TIME - PRNT) / TENDT + 1.0D-7
           NT = TIMP
           PROD = CHEK * VEL(3,1)
      IF (PROD .LT. ZERO .AND. NT.NE. 1) GO TO 490
      IF (NT .NE. 1) GO TO 510
           PRNT = PRNT + TENDT
  490 CONTINUE
    PRINT 5000, TIME
```

```
      PRINT 6000
          DO 500 JB = 1,NB
      PRINT 6010, JB, (UC(IB,JB), IB=1,6), (VEL(IB,JB), IB=1,6)
  500     CONTINUE
  510     CONTINUE
          DO 520 JB = 1,NB
          DO 520 I = 1,6
               POS(I,JB) = POS(I,JB) + UC(I,JB)
  520     CONTINUE
C
C-->>   CALL SUBROUTINE "FORCES" TO CALCULATE NODAL FORCES IN SPRINGS
      CALL FORCES
C
C-->>   CHECK FORCE IN EACH SPRING:
C-->>     1) FOR SPRINGS IN COMPRESSION, PROCEED TO THE NEXT
C-->>        SEGMENT OF THE STRESS-STRAIN CURVE WHEN HIGHEST
C-->>        STRESS LEVEL IS REACHED IN A GIVEN SEGMENT
C-->>     2) IF FORCE IN ANY SPRING HAS REACHED THE ULTIMATE STRENGTH
C-->>        IN COMPRESSION, TENSION, OR SHEAR, RELEASE ALL SPRINGS
C-->>        OF THE LINKAGE ELEMENT AT THE PARTICULAR NODE
C
          DO 730 J = 1,NB
C
C-->>   HEAD JOINTS
C
               LAST = 19
          IF (J .EQ. NLB(J)) LAST = 43
          DO 660 M = 1,LAST, 6
               I = M
          IF (M .EQ. 25   .OR.   M .EQ. 37) I = M + 3
               I1 = I + 1
               I2 = I + 2
          IF (I .GT. 19) GO TO 610
C
C-->>   CHECK WHETHER FORCE IN A SPRING IS TENSILE, COMPRESSIVE, OR ZERO
C
          IF (FORCE(I,J)) 615, 611, 630
  610     IF (FORCE(I,J)) 630, 611, 615
  611     IF (NSTAGE(I,J) .GT. NSEG) GO TO 660
  615     CONTINUE
C
C-->>   IN THE FOLLOWING STEPS:
C-->>     MAX. NUMBER OF SEGMENTS IN THE STRESS-STRAIN CURVE = 5
C-->>     ARRAY NSTAGE IS USED  TO KEEP TRACK OF THE STATUS OF
C-->>        EACH SPRING IN THE WALL.  WHEN A SPRING FAILS, ONE OF
C-->>        THE FOLLOWING FAILURE CODES IS ASSIGNED TO THAT SPRING IN
C-->>        ARRAY "NSTAGE" FOR PROPER ACTION IN SUBROUTINE "STIF":
C-->>        6: COMPR. FAILURE; 7: SHEAR FAILURE; 8: TENSILE FAILURE
C
C-->>   COMPRESSION
               DFORCE = FORCE(I,J)
          IF (FORCE(I,J) .LT. ZERO) DFORCE = - FORCE(I,J)
          IF (DFORCE .LT. COMPHD(NSEG)) GO TO 620
               NSTAGE(I,J) = 6
               NSTAGE(I1,J) = 7
               NSTAGE(I2,J) = 7
      PRINT 9100, I, J, TIME
          GO TO 640
  620          NS        = NSTAGE(I,J)
               DFORCE = FORCE(I,J)
          IF (FORCE(I,J) .LT. ZERO) DFORCE = - FORCE(I,J)
          IF (DFORCE .GT. COMPHD(NS)) NSTAGE(I,J) = NS+1
          IF (NSTAGE(I,J) .GT. NS) PRINT  9150, NSTAGE(I,J), I, J, TIME
          GO TO 640
C-->>   TENSION
  630          DFORCE = FORCE(I,J)
          IF (FORCE(I,J) .LT. ZERO) DFORCE = - FORCE(I,J)
          IF (DFORCE .LT. TENHED) GO TO 640
               NSTAGE(I,J) = 8
               NSTAGE(I1,J) = 7
               NSTAGE(I2,J) = 7
      PRINT 9200, I, J, TIME
C-->>   SHEAR
  640          DFORC1 = FORCE(I1,J)
          IF (FORCE(I1,J) .LT. ZERO) DFORC1 = - FORCE(I1,J)
               DFORC2 = FORCE(I2,J)
          IF (FORCE(I2,J) .LT. ZERO) DFORC2 = - FORCE(I2,J)
          IF (DFORC1 .LT. SHRHED) GO TO 650
               NSTAGE(I,J) = 8
               NSTAGE(I1,J) = 7
               NSTAGE(I2,J) = 7
      PRINT 9300, I1, J, TIME
  650     IF (DFORC2 .LT. SHRHED) GO TO 660
               NSTAGE(I,J) = 8
               NSTAGE(I1,J) = 7
               NSTAGE(I2,J) = 7
      PRINT 9300, I2, J, TIME
  660     CONTINUE
          IF (J .EQ. NLB(J)) GO TO 690
C
C-->>   SET STATUS OF SPRINGS ON THE LEFT SIDE OF A BLOCK EQUAL
C-->>   TO THAT ON THE RIGHT SIDE OF THE PREVIOUS BLOCK
C
          DO 680 M = 1,3
               NSTAGE(27+M,J) = NSTAGE(12+M,J-1)
               NSTAGE(30+M,J) = NSTAGE(18+M,J-1)
               NSTAGE(39+M,J) = NSTAGE(   M,J-1)
               NSTAGE(42+M,J) = NSTAGE( 6+M,J-1)
  680     CONTINUE
  690     CONTINUE
```

```
C
C-->>   BED JOINTS
C
              N = - 1
          DO 760 M = 5,47,6
              I = M
          IF (M .EQ. 29  .OR.  M .EQ. 41) I = M-3
              I1 = I + 1
              I2 = I - 1
              N = - N
          IF (N .EQ. 1) GO TO 710
C-->>   CHECK IF FORCE IN A SPRING IS TENSILE, COMPRESSIVE OR ZERO
          IF (FORCE(I,J)) 730, 711, 715
  710     IF (FORCE(I,J)) 715, 711, 730
  711     IF (NSTAGE(I,J) .GT. NSEG) GO TO 760
  715     CONTINUE
C
C-->>   REFER TO COMMENTS ON HEAD JOINTS ABOVE FOR "NSTAGE".
C-->>   AND FAILURE CODES
C
C-->>   COMPRESSION
C
              DFORCE = FORCE(I,J)
          IF (FORCE(I,J) .LT. ZERO) DFORCE = - FORCE(I,J)
          IF (DFORCE .LT. COMPBD(NSEG)) GO TO 720
              NSTAGE(I,J) = 6
              NSTAGE(I1,J) = 7
              NSTAGE(I2,J) = 7
        PRINT 9100, I, J, TIME
          GO TO 740
  720         NS        = NSTAGE(I,J)
              DFORCE = FORCE(I,J)
          IF (FORCE(I,J) .LT. ZERO) DFORCE = - FORCE(I,J)
          IF (DFORCE .GT. COMPBD(NS)) NSTAGE(I,J) = NS+1
          IF (NSTAGE(I,J) .GT. NS) PRINT  9150, NSTAGE(I,J), I, J, TIME
          GO TO 740
C-->>   TENSION
  730         DFORCE = FORCE(I,J)
          IF (FORCE(I,J) .LT. ZERO) DFORCE = - FORCE(I,J)
          IF (DFORCE .LT. TENBED) GO TO 740
        PRINT 9200, I, J, TIME
              NSTAGE(I,J) = 8
              NSTAGE(I1,J) = 7
              NSTAGE(I2,J) = 7
C-->>   SHEAR
  740         DFORC1 = FORCE(I1,J)
          IF (FORCE(I1,J) .LT. ZERO) DFORC1 = - FORCE(I1,J)
              DFORC2 = FORCE(I2,J)
          IF (FORCE(I2,J) .LT. ZERO) DFORC2 = - FORCE(I2,J)
          IF (DFORC1 .LT. SHRBED) GO TO 750
              NSTAGE(I,J) = 8
              NSTAGE(I1,J) = 7
              NSTAGE(I2,J) = 7
        PRINT 9300, I1, J, TIME
  750     IF (DFORC2 .LT. SHRBED) GO TO 760
              NSTAGE(I,J) = 8
              NSTAGE(I1,J) = 7
              NSTAGE(I2,J) = 7
        PRINT 9300, I2, J, TIME
  760     CONTINUE
  790     CONTINUE
C
C-->>   IF FAILURE OR A SHIFT TO A HIGHER SEGMENT OF THE STRESS-STRAIN
C-->>   CURVE HAD OCCURED, CALL SUBROUTINE "STIF" TO ASSIGN PROPER "E"
C-->>   OR SET STIFFNESS TO ZERO, DEPENDING ON THE CASE
C
          DO 791 I = 1,48
          DO 791 J = 1,NB
          IF (NSTAGE(I,J) .GT. 1) GO TO 792
  791     CONTINUE
          GO TO 793
  792     CONTINUE
        CALL STIF
  793     CONTINUE
              CHEK = VEL(3,1)
C
C-->>   DETERMINE CODE FOR TIME ADJUSTMENT IF CHANGE IN SLOPE OF THE
C-->>   FORCING FUNCTION OCCURS WITHIN A TIME ICREMENT DT
C
              TDIF = RTIME - TIME
              RCDIF = RTIME + CTIME - TIME
          IF (TDIF) 798, 800, 797
  797     IF (TDIF .GE. DT) GO TO 800
          IF (DABS(TDIF) .LE. 1.0D-8) GO TO 800
              TIME = TIME + TDIF
              TD = TDIF
              ITD = 1
          GO TO 300
  798     IF (RCDIF) 800, 800, 799
  799     IF (RCDIF .GE. DT) GO TO 800
          IF (DABS(RCDIF) .LE. 1.0D-8) GO TO 800
              TIME = TIME + RCDIF
              RC = RCDIF
              IRC = 1
          GO TO 300
  800     CONTINUE
 4000 FORMAT (//, 14H**** MORE THAN, I5, 27H ITERATIONS NEEDED ... WILL,
     1           10H STOP ****)
 5000 FORMAT (1H0, 2(/), 1X, 7HTIME = , F9.7, 9H  SECONDS)
 6000 FORMAT (1H0, 31(1H-),  1X, 13HDISPLACEMENTS, 1X, 31(1H-),
```

```
1         3X, 19(1H-), 1X, 10HVELOCITIES, 1X, 21(1H-), /,
2         1H , 5HBLOCK, 6X, 1HU, 11X, 1HV, 11X, 1HW, 9X, 5HTHETA,
3         8X, 4HBETA, 8X, 3HPHI,10X, 1HU, 8X, 1HV, 8X, 1HW,
4         6X, 5HTHETA, 4X, 4HBETA, 6X, 3HPHI)
6010 FORMAT (1H , I3, 2X, 6(1X,D11.4), 1X, 6(1X,F8.4) )
9100 FORMAT (1H0, 39H$$$$$ COMPRESSION FAILURE AT SPRING NO., I4,
1         16H . . . BLOCK NO., I5, 5X, 7HTIME = , F10.7,8H SECONDS)
9150 FORMAT (1H0,51H$$$$$ SHIFT IN MODULUS OF ELASTICITY TO SEGMENT NO.
+              , 1X, I3, 15H... SPRING NO. , I3, 12H, BLOCK NO. ,
+              I5, 5X, 8H TIME = , F10.7, 8H SECONDS)
9200 FORMAT (1H0, 35H>>>>> TENSION FAILURE AT SPRING NO., I4,
2         16H . . . BLOCK NO., I5, 5X, 7HTIME = , F10.7,8H SECONDS)
9300 FORMAT (1H0, 33H***** SHEAR FAILURE AT SPRING NO., I4,
3         16H . . . BLOCK NO., I5, 5X, 7HTIME = , F10.7,8H SECONDS)
     RETURN
     END
```

```
      SUBROUTINE LOAD
C
C     ****************************************************************
C     *                                                              *
C     * FUNCTION: CALCULATE DYNAMIC LOAD ON EACH BLOCK               *
C     *                                                              *
C     * OTHER SUBROUTINES CALLED: "STAHS" OR "STARB" FOR STATIC LOAD *
C     *                                                              *
C     ****************************************************************
C
      IMPLICIT REAL * 8 (A-H , O-Z)
      COMMON /BLAST / DPEAK, SPEAK, RTIME, CTIME, DTIME, TYPE,ILOAD,KDYN
      COMMON /BLCK  / A, B, C, WJ, TI, TE, EH, ELENTH
      COMMON /JOINTS/ TXJ, TYJ, TLJ, TRJ, TUJ, TBJ
      COMMON /MASLOD/ P(6,125), MASINV(6,125), BLCKWT(125)
      COMMON /NOBLOK/ AB(2,125), NB, NBX, NBY, NER, NOP, NBXX, MNBX
      COMMON /RESULT/ FORCE(48,125), LC(6,125), POS(6,125), TIME
      COMMON /WALL  / L, L1, L2, H, H2, ILS, IRS, IUS, IBS, KODE
      DIMENSION PEAKP (125)
      DOUBLE PRECISION L, L1, L2, MASINV, KS
      DATA  U, S /4HUNIF, 4HSINE/
      DATA  PI/3.14159265358979D00/, PEAKD,PEAKS/2*0.0D00/
      DATA  ZERO, ONE, TWO /0.0D00, 1.0D00, 2.0D00/
C
C-->>   CALCULATE EFFECTIVE LENGTH AND HEIGHT
C
        IF (TIME .NE. ZERO) GO TO 260
            EFFL = L
        IF (IRS .EQ. 3) EFFL = TWO * EFFL - TXJ
            PIL = PI / EFFL
            EFFH = H
        IF (IUS .EQ. 3) EFFH = TWO * EFFH - TYJ
            PIH = PI / EFFH
            KDYN = 0
            ABX  = NBX
            ABY  = NBY
            XB   = NB
            A2 = TWO * A
            MNBX1 = MNBX + 1
C
C-->>   CALCULATE STATIC LOAD: CALL SUBROUTINE "STARB" FOR RUNNING BOND
C-->>   AND SUBROUTINE "STAHS" FOR HORIZONTAL STACK
C
        IF (KODE .EQ. 0) GO TO 150
      CALL STARB
        GO TO 200
  150 CALL STAHS
  200   CONTINUE
C
C-->>   CACULATE DYNAMIC LOAD
```

```
C
      DO 500 J = 1,NB
           PEE2 = P(2,J)
           RCTIME = RTIME + CTIME
      IF (CTIME .EQ. ZERO) GO TO 210
           RATIO = (TIME - RCTIME) / DTIME
           DRATIO = ONE - RATIO
  210 IF (TYPE  .EQ. S) GO TO 220
C
C-->>  UNIFORMLY DISTRIBUTED LOAD
C
           AREA = AB(1,J) * AB(2,J)
           PEAKD = DPEAK * AREA
      IF (TYPE .EQ. U) GO TO 240
C
C-->>  SINOSOIDAL LOAD
C
  220 CONTINUE
           TYU = TYJ
           TYB = TYJ
      IF (J .LE. NBXX) TYB = TBJ
      IF (J .GT. MNBX) TYU = TUJ
           Y1 = POS(2,J) - B - TYB / TWC
           Y2 = POS(2,J) + B + TYU / TWC
  230      PTOT1 = - (DCOS(PIH*Y2)-DCOS(PIH*Y1))/ PIH
      IF (NBY .EQ. 1) PTOT1 = AB(2,J)
           X1 = POS(1,J) - AB(1,J) / TWC
           X2 = POS(1,J) + AB(1,J) / TWC
           PTOT = - (DCOS(PIL*X2) - DCOS(PIL*X1)) / PIL
      IF (NBXX .EQ. 1) PTOT = AB(1,J)
           PEAKS = SPEAK * PTOT * PTOT1
C
C-->>  COMBINE UNIFORM AND SINOSOIDAL PEAK LOAD ON A BLOCK
C
  240      PEAKP(J) = PEAKD + PEAKS
  500 CONTINUE
  260 CONTINUE
C
C-->>  CALCULATE DYNAMIC LOAD ON A BLOCK AT GIVEN TIME
C
      DO 400 J =1,NB
           P(3,J) = PEAKP(J)
      IF (TIME - RCTIME) 250, 400, 275
  250 IF (TIME .GT. RTIME) GO TO 400
      GO TO (400, 400, 300, 300, 350), ILOAD
  275 CONTINUE
      GO TO (375, 325, 325, 375, 350), ILOAD
  300      P(3,J) = PEAKP(J)*TIME / RTIME
      GO TO 400
  325      P(3,J) = PEAKP(J) * DRATIO
      GO TO 400
  350      P(3,J) = PEAKP(J)*DRATIO * DEXP(-RATIO)
      GO TO 400
  375      P(3,J) = ZERO
      IF (P(2,J) .NE. PEE2) KDYN = 1
  400 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE STAHS
C
C     ******************************************************************
C     *                                                                *
C     *            >> FOR THE HORIZONTAL STACK PATTERN <<              *
C     *                                                                *
C     * FUNCTIONS: 1) CALCULATE STATIC LOAD ON EACH BLOCK              *
C     *            2) CALCULATE WEIGHT OF EACH BLOCK                   *
C     *                                                                *
C     ******************************************************************
C
      IMPLICIT REAL * 8 (A-H , O-Z)
      COMMON /BLCK  / A, B, C, WJ, TI , TE, EH, ELENTH
      COMMON /BORDER/ NLB(125), NRB(125)
      COMMON /JOINTS/ TXJ, TYJ, TLJ, TRJ, TUJ, TBJ
      COMMON /MASLOD/ P(6,125), MASINV(6,125), BLCKWT(125)
      COMMON /NOBLOK/ AB(2,125), NB, NBX, NBY, NER, NOR, NBXX, MNBX
      COMMON /WALL  / L, L1, L2, H, H2, ILS, IRS, IUS, IBS, KODE
      COMMON /WEIGHT/ WT, MORTWT, IBLCK
      DOUBLE PRECISION L, L1, L2, KU, KS, MASINV, MORTWT
      DATA  ONE, TWO / 1.0D00, 2.0D00/
C
              A2 = TWO * A
              B2 = TWO * B
              WJH = C
              WJB = C
          IF (IBLCK .GE. 2) WJB = WJ
          IF (IBLOK .EQ. 3) WJH = WJ
C
C-->>   CALCULATE WEIGHT OF MORTAR JOINTS
C
              COMB = TWO = WJB * MORTWT
              COMH = TWO * WJH * MORTWT
              BHORZ = A2 * TYJ * COMB
              BUPER = A2 * TUJ * COMB
              WVERT = (H-TBJ) * TXJ * COMH
              BVERT = WVERT / NBY
C
C-->>   CALCULATE BLOCK WEIGHT AND STATIC LOAD ON EACH BLOCK
C
          DO 100 JB = 1,NB
              NROW = JB / NBX
              RONO = NBY - NROW
          IF (IUS .EQ. 3) RONO = TWO * NBY - NROW
          IF (JB .EQ. NRB(JB)) RONO = RONO + 1
              P(2,JB) =-(RONO*(WT+BVERT) + (RONO-ONE)*BHORZ + BUPER)
              AREA = AB(1,JB) * AB(2,JB) - A2 * B2
              BLOKWT(JB) = WT + AREA * COMB
          IF (IBLCK .EQ. 2) BLOKWT(JB) = BLOKWT(JB) + (AB(1,JB)-A2)*B2
     a                                   * (COMH - COMB)
  100     CONTINUE
          RETURN
          END
```

```
      SUBROUTINE STARB
C
C     ****************************************************************
C     *                                                              *
C     *          >> FOR THE RUNNING BOND PATTERN <<                  *
C     *                                                              *
C     * FUNCTIONS: 1) CALCULATE STATIC LOAD ON EACH BLOCK            *
C     *            2) CALCULATE WEIGHT OF EACH BLOCK                 *
C     *                                                              *
C     ****************************************************************
C
      IMPLICIT REAL * 8 (A-H , O-Z)
      COMMON /BLOK  / A, B, C, WJ, TI, TE, EH, ELENTH
      COMMON /BORDER/ NLB(125), NRB(125)
      COMMON /JOINTS/ TXJ, TYJ, TLJ, TRJ, TUJ, TBJ
      COMMON /MASLOD/ P(6,125), MASINV(6,125), BLOKWT(125)
      COMMON /NOBLOK/ AB(2,125), NB, NBX, NBY, NER, NOR, NBXX, MNBX
      COMMON /WALL  / L, L1, L2, H, H2, ILS, IPS, IUS, IBS, KODE
      COMMON /WEIGHT/ WT, MORTWT, IBLOCK
      DOUBLE PRECISION MASINV, L, L1, L2, MORTWT
      DATA  ONE, TWO /1.0D00, 2.0D00/
C
              A2 = TWO * A
              B2 = TWO * B
              NROW = 0
              WJH = 0
              WJB = 0
C
C-->>  CALCULATE WEIGHT FACTORS FOR JOINTS
C
         IF (IBLOCK .GE. 2) WJB = WJ
         IF (IBLOCK .EQ. 3) WJH = WJ
              COMB = TWO * WJB * MORTWT
              COMH = TWO * WJH * MORTWT
         DO 1000 J = 1,NB
C
C-->>  CALCULATE BLOCK WEIGHT
C
              Q = ONE
              AA = A2
              AREAB = A2 * B2
         IF (AB(1,J) .GT. A2) GO TO 400
              AA = ELENTH
              AREAB = ELENTH * B2
              Q = ELENTH / A2
  400         BLOKWT(J) = Q*WT + (AB(1,J)*AB(2,J) - AREAB) * COMB
         IF (IBLOCK .EQ. 2) BLOKWT(J) = BLOKWT(J) + (AB(1,J) - AA) * B2
     a                                  * (COMH - COMB)
C
C-->>  CALCULATE STATIC LOAD ON EACH BLOCK
C
         IF (J .NE. NLB(J)) GO TO 500
              NROW = NROW + 1
              AOR = NOR - NROW/2
              AER = NER - (NROW-1)/2
         IF (IUS .NE. 3) GO TO 500
              AOR = AOR + NER
              AER = AER + NOR
         GO TO 550
  500    IF (J .NE. NRB(J)) GO TO 600
  550         VERTM = TXJ * S * COMH
              VERTB = TXJ * WT / (TWO * A2)
              S = ONE
         IF (AB(1,J) .GE. A2) S = - ONE
              P(2,J) = - AOR*BLOKWT(J) - AER*(BLOKWT(J) - S*VERTM + S*
     1                   VERTB)
         GO TO 700
  600         P(2,J) = - BLOKWT(J) * (AOR + AER)
  700    IF (NROW .NE. 1) GO TO 800
              AR = TBJ / TWO
         IF (TYJ .NE. TBJ) AR = (TYJ - TBJ) * NBY
              P(2,J) = P(2,J) + AR * AB(1,J) * COMB
  800    IF (TYJ .NE. TUJ) P(2,J) = P(2,J) + (TYJ-TUJ)*AB(1,J)*COMB/TWO
 1000    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MASS
C
C     ******************************************************************
C     *                                                                *
C     * FUNCTION: CALCULATE THE MASS AND MASS MOMENT OF INERTIA        *
C     *           FOR EACH MASONRY UNIT                                *
C     *                                                                *
C     ******************************************************************
C
      IMPLICIT REAL * 8 (A-H , O-Z)
      COMMON /BLCK  / A, B, C, WJ, TI, TE, EH, ELENTH
      COMMON /BORDER/ NLB(125), NRB(125)
      COMMON /JOINTS/ TXJ, TYJ, TLJ, TRJ, TUJ, TBJ
      COMMON /MASLOD/ R(6,125), MASINV(6,125), BLOKWT(125)
      COMMON /NOBLOK/ AB(2,125), NB, NBX, NBY, NER, NOR, NBXX, MNBX
      COMMON /WALL  / L, L1, L2, H, H2, ILS, IPS, IUS, IBS, KODE
      COMMON /WEIGHT/ WT, MORTWT, IBLCK
      DOUBLE PRECISION L, L1, L2, MASINV, MORTWT
      DATA  GRAVTY/386.088D00/
      DATA  FOUR, TWELVE /4.0D00, 12.0D00/
      DATA  ZERO, HALF, ONE, TWO /0.0D00, 5.0D-1, 1.0D00, 2.0D00/
C
C-->>  THE  I N V E R S E  OF THE (M) MATRIX WILL BE DEVELOPED DIRECTLY
C
             A2     = A * TWO
             B2     = B * TWO
             C2     = C * TWO
             A2SQ          = A2 * A2
             B2SQ   = B2 * B2
             C2SQ   = C2 * C2
             MNBX1  = MNBX + 1
         DO 100 J=1,NB
C-->>  CALCULATE THE INVERSE OF THE MASS OF THE BLOCK
C
             XMAS        = GRAVTY / BLOKWT(J)
C
C-->>  ASSIGN VALUES TO FIRST 3 ELEMENTS OF INVERSE MASS MATRIX
C
             MASINV(1,J) = XMAS
             MASINV(2,J) = XMAS
             MASINV(3,J) = XMAS
  100    CONTINUE
C
C-->>  CHECK IF MASONRY UNIT IS SOLID BRICK OR CONCRETE BLOCK
C
         IF (IBLOCK .NE. 1) GO TO 250
         DO 200 J=1,NB
C
C-->>  CALCULATE THE INVERSE OF THE MASS MOMENT OF INERTIA
C-->>  FOR SOLID BRICKS
C
             ABSQ1 = AB(1,J) * AB(1,J)
             ABSQ2 = AB(2,J) * AB(2,J)
             XMAS        = GRAVTY / BLOKWT(J)
             MASINV(4,J) = TWELVE * XMAS / (ABSQ2 +C2SQ)
             MASINV(5,J) = TWELVE * XMAS / (ABSQ1 + C2SQ)
             MASINV(6,J) = TWELVE * XMAS / (ABSQ1 + ABSQ2)
  200    CONTINUE
         GO TO 700
  250    CONTINUE
C
C-->>  THE REMAINING PORTION OF THIS SUBROUTINE IS FOR CONCR. BLOCKS
C
             KORES = IBLOCK
             CORES  = KORES
             COREM1 = CORES - ONE
             RW = C - WJ
             XHOLE = (A2-TWO*(TE+EH) - COREM1 * TI) / CORES
             YHOLE = TWO * RW
             RY = C - WJ/TWO
             RX1 = A - EH - TE/TWO
             RX2 = A - EH - TE - XHOLE - TI/TWO
C
C-->>  CALCULATE VOLUMES FOR ELEMENTS OF A CONCRETE BLOCK
C
             VSHEL = A2 * B2 * WJ
             VRIBI = YHOLE * B2 * TI
             VRIBE = YHOLE * B2 * TE
             VTOT = COREM1 * VRIBI + TWO * (VRIBE + VSHEL)
C
C-->>  CALCULATE WEIGHT OF ELEMENTS OF A CONCRETE BLOCK
C-->>  AND SURROUNDING MORTAR OF A HALF JOINT
C
             UWT = WT / VTOT
             WSHELL = VSHEL * WT / VTOT
             WRIBI = VRIBI * WT / VTOT
             WRIBE = VRIBE * WT / VTOT
             WEBMOR = TXJ * B2 * RW * MORTWT
         IF (KORES .EQ. 3) WEBMOR = ZERO
         DO 600 J = 1,NB
             ABSQ1 = AB(1,J) * AB(1,J)
             ABSQ2 = AB(2,J) * AB(2,J)
             WMORT = HALF*(BLOKWT(J) - WT) - WEBMOR
             WSHEL = WSHELL + WMORT
         IF (KODE .NE. 0  .AND.  AB(1,J) .LT. A2) GO TO 650
C
C-->>  FULL CONCRETE BLOCKS
C
C-->>  CALCULATE THE INVERSE OF THE MASS MOMENT OF INERTIA
C
```

```
            MASINV(4,J) = TWELVE * GRAVTY/
     A                    (WSHEL*((WJ*WJ+ABSQ2) + TWELVE*RY*RY) * TWO
     B                    + (WRIBI * COREM1 + TWO*(WRIBE+WEBMCR))
     C                    * (B2SQ + YHOLE**2) )
            MASINV(5,J) = TWELVE * GRAVTY /
     D                    (WSHEL*(WJ*WJ+ABSQ1+TWELVE*RY*RY) * TWO
     E                    +WRIBE *(TE*TE+YHOLE**2+TWELVE*RX1**2)*TWO+
     F                    WRIBI*(TI*TI+YHOLE**2+TWELVE*RX2**2)*COREM1
     G                    +WEBMCR * ( (TXJ/TWO)**2 + YHOLE**2
     H                    +TWELVE * (A+TXJ/FOUR)**2) * TWO)
            MASINV(6,J) = TWELVE * GRAVTY /
     I                    (  WSHEL * (ABSQ1 + ABSQ2) * TWO
     J                    +WRIBE * (TE*TE+B2SQ+TWELVE*RX1**2) * TWO
     K                    +WRIBI * (TI*TI+B2SQ+TWELVE*RX2**2)*COREM1
     L                    +WEBMCR * (   (TXJ/TWO)**2 + B2SQ
     M                    +TWELVE * (A+TXJ/FOUR)**2) * TWO)
        GO TO 600
C
C-->>   PARTIAL CONCRETE BLOCKS
C
  650       XLENTH = ELENTH + HALF * TXJ
C
C-->>   ESTABLISH LOCATION OF RIBS WITH RESPECT TO REFERENCE
C
            X6 = TXJ/TWO + TE + XHOLE
            X5 = X6 + TI
            X4 = X5 + XHOLE
            X3 = X4 + TI
        IF (KORES .EQ. 2) X3 = X4 + TE
            X2 = X3 + XHOLE
            X1 = X2 + TE
            T1      = TE
            T2      = TI
            T3      = TI
C-->>   DETERMINE WHICH RIBS EXIST IN THE PARTIAL BLOCK
C
        IF (XLENTH .GT. X1) GO TO 475
        IF (XLENTH .LT. X2) GO TO 425
            T1 = XLENTH - X2
        GO TO 475
  425   IF (XLENTH .GT. X3) GO TO 463
        IF (XLENTH .LT. X4) GO TO 450
            T2 = XLENTH - X4
        GO TO 463
  450   IF (XLENTH .GT. X5) GO TO 462
        IF (XLENTH .LT. X6) GO TO 461
            T3 = XLENTH - X6
        GO TO 462
  461       T3 = ZERO
  462       T2 = ZERO
  463       T1 = ZERO
  475   CONTINUE
            AM = YHOLE * TXJ/TWO * MORTWT/UWT
            AIM = AM
        IF (KORES .EQ. 3) AM = ZERO
        IF (XLENTH .LT. X2  .AND.  XLENTH .GT. X3) AM = ZERO
        IF (XLENTH .LT. X4  .AND.  XLENTH .GT. X5) AM = ZERO
        IF (XLENTH .LT. X6) AM = ZERO
C
C-->>   DETERMINE AREAS OF WEBS AND FACE SHELLS
C
            ASHEL = TWO * AB(1,J) * WJ
            ARIB = YHOLE * TE
            AT1  = YHOLE * T1
            AT2  = YHOLE * T2
            AT3  = YHOLE * T3
            ATOT = ASHEL + ARIB + AT1 + AT2 + AT3 + AIM + AM
C
C-->>   CALCULATE WEIGHT OF MORTAR ATTACHED TO FACE SHELL
C
            WMORT = HALF * (BLOKWT(J) - WT*ELENTH/A2) - WEBMCR
C
C-->>   ADD ATTACHED MORTAR TO WEIGHT OF SHELL
C
            WSHEL = WSHELL * ELENTH/A2 + WMORT
C
C-->>   DETERMINE WEIGHT OF MORTAR ATTACHED TO EXTERIOR RIBS
C
            WEBMEX = AM * B2 * MORTWT
C
C-->>   DETERMINE THE CENTER OF GRAVITY OF PARTIAL BLOCKS
C
            CG = (ASHEL * HALF * AB(1,J) + ARIB * (EH +HALF*(TE+TXJ))
     1               + AT1 * (X2 + HALF * T1)
     2               + AT2 * (X4 + HALF * T2)
     3               + AT3 * (X6 + HALF * T3)
     4               + AIM * TXJ/FOUR + AM * (XLENTH+TXJ/TWO)) / ATOT
C
C-->>   CALCULATE THE INVERSE OF THE MASS MOMENT OF INERTIA
C
            MASINV(4,J) = TWELVE * GRAVTY /
     A                      (WSHEL * (WJ*WJ+ABSQ2+TWELVE*RY*RY)  * TWO
     B                    + (WRIBE * (T1+TE)/TE + WRIBI * (T2+T3)/TI)
     C                      * (B2SQ + YHOLE * YHOLE))
            WEBM5 =            WEBMCR * (   (TXJ/TWO)**2 + YHOLE**2
     a                                  + TWELVE*(CG-TXJ/FOUR)**2)
            MASINV(5,J) = TWELVE * GRAVTY / ( WEBM5
     1                    +  WSHEL * (ABSQ1 + WJ*WJ+TWELVE*RY*RY)*TWO
     2                    + WRIBE * (T1*(T1*T1+YHOLE**2+TWELVE*(X2-CG
     3                            +T1/TWO)**2)/TE + TE*TE+YHOLE**2
```

```
     4                                 +TWELVE*(CG-TXJ/TWO-EH-TE/TWO)**2)
     5                       + WRIBI * (T2*(T2*T2+YHOLE**2+TWELVE*(X4-CG
     6                                 +T2/TWO)**2) + T3*(T3*T3+YHOLE**2
     7                                 +TWELVE*(X6-CG+T3/TWO)**2)) / TI
     8                       + WEBMEX * (   (TXJ/TWO)**2 + YHOLE**2
     9                                  + TWELVE*(XLENTH-CG+TXJ/FOUR)**2))
           WEBM6 =          WEBMOR * (   (TXJ/TWO)**2 + B2SQ
     $                                 +TWELVE*(CG - TXJ/FOUR)**2)
           MASINV(6,J) = TWELVE * GRAVTY /  ( WEBM6
     A                       +  WSHEL * ( A2SQ1 + A2SQ2
     B                                  + TWELVE * (XLENTH/TWO - CG)**2)
     C                       + WRIBE * (T1 * (T1*T1+B2SQ + TWELVE*(X2-CG
     D                                 +T1/TWO)**2)/TE + TE*TE+B2SQ
     E                                 + TWELVE*(CG-TXJ/TWO-EH-TE/TWO)**2)
     F                       + WRIBI * (T2 * (T2*T2+B2SQ+ TWELVE*(X4-CG
     G                                 + T2/TWO)**2) + T3 * (T3*T3 + B2SQ
     H                                 + TWELVE * (X6-CG+T3/TWO)**2)) / TI
     I                       + WEEMEX * (   (TXJ/TWO)**2 + B2SQ
     J                                  + TWELVE*(XLENTH-CG+TXJ/FOUR)**2))
 600      CONTINUE
 700      CONTINUE
      RETURN
      END
```

```
      SUBROUTINE STIF
C
C     *******************************************************************
C     *                                                                 *
C     * FUNCTION: SET UP THE STIFFNESS MATRICES FOR EACH BLOCK          *
C     *           AND THE SURROUNDING BLOCKS                            *
C     *                                                                 *
C     *******************************************************************
C
      IMPLICIT REAL * 8 (A-H , O-Z)
      COMMON /BLAST / DPEAK, SPEAK, RTIME, CTIME, DTIME, TYPE,ILOAD,KDYN
      COMMON /BLOK  / A, B, C, WJ, TI, TE, EH, ELENTH
      COMMON /BORDER/ NLB(125), NRB(125)
      COMMON /CURVE / NSTAGE(48,125), STRESS( 5), STRAIN( 5), E(5), NSEG
      COMMON /GLAMKS/ KS(48,125), C(48,6), LAMR, LAML, LAMV, LAMW, AKK
      COMMON /JOINTS/ TXJ, TYJ, TLJ, TRJ, TUJ, TBJ
      COMMON /MASLOD/ P(6,125), MASINV(6,125), BLOKWT(125)
      COMMON /NOBLOK/ AR(2,125), NB, NBX, NBY, NBR, NOR, NBXX, NNBX
      COMMON /RESULT/ FORCE(48,125), UC(6,125), POS(6,125), TIME
      COMMON /STFMAT/ AK(750,6), BK(750,6), CK(750,6), DK(750,6),
     $                EK(750,6), FK(750,6), HK(750,6)
      COMMON /WALL  / L, L1, L2, H, H2, ILS, IRS, IUS, IBS, KODE
      DIMENSION ST(48,125), LAM(2)
      DOUBLE PRECISION LAM, L, L1, L2
      DOUBLE PRECISION KS, MASINV, LAMR, LAML, LAMV, LAMW
      DATA  ZERO, ONE, TWO /0.0D00, 1.0D00, 2.0D00/
C
              A2       = TWO * A
              KR = - 1
              KM = 2
              KN = 1
          DO 100 J = 1,NB
          IF (J .NE. NLB(J)) GO TO 120
              KR = -KR
              KM = KM - KR
              KN = KN + KR
              LAM(KM) = LAMR
              LAM(KN) = LAML
  120     CONTINUE
C
C-->>  MULTIPLY SPRING STIFFNESSES CALCULATED IN SUBROUTINE "LINKEL"
C-->>  BY THE PROPER 'E' VALUE.  SPRING STATUS IS UPDATED ACCORDING
C-->>  TO ARRAY "NSTAGE" SET UP IN SUBROUTINE "SOLVER"
C
          DO 200 I = 1,48
              NS = NSTAGE(I,J)
          IF (NS .GT. NSEG) GO TO 150
          IF (TIME .EQ. ZERO) ST(I,J) = KS(I,J)
              KS(I,J) = ST(I,J) * E(NS)
          GO TO 200
```

```
  150       KS(I,J) = ZERO
  200     CONTINUE
          IF (TIME .NE. 0) GO TO 100
             YH = H
          IF (IUS .EQ. 3) YH = TWO * H - TUJ - TBJ + TYJ
             AKK = TWO * C * (A2+TXJ) * E(1) / YH
  100     CONTINUE
 9998     CONTINUE
             ALMR   = A * (ONE - LAM(1))
             ALML   = A * (ONE - LAM(2))
             BLAMV  = B * (ONE - LAMV)
             CLAMW  = C * (ONE - LAMW)
             ATX = A + TXJ/TWO
             BTY    = B + TYJ / TWO
             ALLSQ = ALML * ALML
             ARLSQ = ALMR * ALMR
             BLAMSQ = BLAMV * BLAMV
             CLAMSQ = CLAMW * CLAMW
             ATSQ = ATX * ATX
             BTSQ   = BTY * BTY
C
C-->>                                        6
C-->>     SET UP STIFFNESS MATRIX (K  )
C
          DO 450 J = 1,NB
             NA = 6 * (J-1)
             N1 = NA + 1
             N2 = NA + 2
             N3 = NA + 3
             N4 = NA + 4
             N5 = NA + 5
             N6 = NA + 6
             AA      = A
          IF (AB(1,J) .LT. A2) AA = ELENTH / TWO
             AATX    = AA + TXJ / TWO
             AATSQ   = AATX * AATX
             ALAMR = ALMR
             ALAML = ALML
             ALLMSQ = ALLSQ
             ARLMSQ = ARLSQ
          IF (AB(1,J) .GE. A2) GO TO 300
             ALAMR = ZERO
             ALAML = ZERO
  300     CONTINUE
             AKT1    = ZERO
             AKS1    = ZERO
             AKS2    = ZERO
             AKT2    = ZERO
             AKY3    = ZERO
             AKX3    = ZERO
C-->>     SUM SPRING STIFFNESSES FOR RIGHT HALF OF BLOCK
           DO 500 I = 1,19,6
                AKT1    = AKT1 + KS(I,J)
                AKS1    = AKS1 + KS(I+3,J)
                AKS2    = AKS2 + KS(I+1,J)
                AKT2    = AKT2 + KS(I+4,J)
                AKY3    = AKY3 + KS(I+2,J)
                AKX3    = AKX3 + KS(I+5,J)
  500      CONTINUE
C-->>     SUM SPRING STIFFNESSES FOR LEFT HALF OF BLOCK
                AKX3R = AKX3
                AKT2R = AKT2
                KK      = 28
                MM      = 31
                N       = 3
          DO 525 LL = 1,2
             DO 550 I = KK,MM,3
                AKT1    = AKT1 + KS(I,J)
                AKS1    = AKS1 + KS(I-N,J)
                AKS2    = AKS2 + KS(I+1,J)
                AKT2    = AKT2 + KS(I-N+1,J)
                AKY3    = AKY3 + KS(I+2,J)
                AKX3    = AKX3 + KS(I-N+2,J)
                N       = -N
  550        CONTINUE
                KK      = 40
                MM      = 43
  525     CONTINUE
                AKX3L = AKX3 - AKX3R
                AKT2L = AKT2 - AKT2R
                AK(N1,1)= AKT1 + AKS1
                AK(N2,2)= AKS2 + AKT2
                AK(N3,3)= AKY3 + AKX3
                AK(N4,4) = CLAMSQ*AK(N2,2) + BLAMSQ*AKY3 + BTSQ*AKX3
                AK(N5,5) = CLAMSQ*AK(N1,1) + ALLMSQ*AKX3L + ARLMSQ*AKX3R
     $                     +AATSQ * AKY3
                AK(N6,6) = ALLMSQ*AKT2L + ARLMSQ*AKT2R + BLAMSQ*AKT1
     $                     + AATSQ*AKS2 + AKS1 * BTSQ
                AK(N1,5) = ZERO
                AK(N1,6)= ZERO
                AK(N2,4) = ZERO
                AK(N2,6) = ZERO
                AK(N3,4) = ZERO
                AK(N3,5) = ZERO
                AK(N4,5) = ZERO
                KI = 1
                NI = 1
                MI = -1
          DO 560 II = 1,43,6
                I = II
```

```
      IF (II .EQ. 25  .OR.  II .EQ. 37) I = II + 3
      IF (I .GT. 19) KI = - 1
      IF (I .GT. 7) NI = -1
      IF (I .GT. 37) NI = 1
      IF (I .GT. 19) GO TO 555
         KSI4 = KS(I+4,J)
         KSI5 = KS(I+5,J)
555   CONTINUE
         MK = NI * KI
         AK(N1,5) = AK(N1,5)+ NI * KS(I,J) * CLAMW
         AK(N1,6) = AK(N1,6) + NI * KS(I,J) * BLAMV
         AK(N2,4) = AK(N2,4) - NI * KS(I+1,J) * CLAMW
         AK(N2,6) = AK(N2,6) + MI * KS(I+1,J) * AATX
         AK(N3,4) = AK(N3,4) - MI * KS(I+2,J) * BLAMV
         AK(N3,5) = AK(N3,5) - MI * KS(I+2,J) * AATX
         AK(N4,5) = AK(N4,5) + MK * KS(I+2,J) * BLAMV * AATX
      IF (II .EQ. 25  .OR.  II .EQ. 37) I = II - 3
         IF (II .GT. 19) GO TO 556
         KI4 = KS(I+4,J)
         KI5 = KS(I+5,J)
556   CONTINUE
         AK(N1,5) = AK(N1,5) + NI * KS(I+3,J) * CLAMW
         AK(N1,6) = AK(N1,6) + MI * KS(I+3,J) * BTY
         AK(N2,4) = AK(N2,4) - MI * KS(I+4,J) * CLAMW
         AK(N2,6) = AK(N2,6) + MI * (KSI4*ALMR + KI4*ALML)
         AK(N3,4) = AK(N3,4) - MI * KS(I+5,J) * BTY
         AK(N3,5) = AK(N3,5) - MI * (KSI5*ALMR + KI5*ALML)
         AK(N4,5) = AK(N4,5) + MK * BTY * (KSI5*ALAMR+KI5*ALAML)
         MI = - MI
560   CONTINUE
         AK46 = AATX* (-KS(2,J) - KS(8,J) + KS(14,J) + KS(20,J)
     $                 -KS(26,J) - KS(32,J) + KS(41,J) + KS(44,J))
         AK46 = AK46 +(- KS(5,J) - KS(11,J) + KS(17,J) + KS(23,J))
     $          * ALAMR+(-KS(26,J) - KS(35,J) + KS(38,J) + KS(47,J))
     $          * ALAML
         AK(N4,6) = AK46 * CLAMW
         AK56 = BLAMV * (- KS(1,J) + KS(7,J) + KS(13,J) - KS(19,J)
     $                 + KS(28,J) - KS(31,J) - KS(40,J) + KS(43,J))
         AK56 = AK56 + (- KS(4,J) + KS(10,J) + KS(16,J) - KS(22,J)
     $            + KS(25,J) - KS(34,J) - KS(37,J) + KS(46,J))* BTY
         AK(N5,6) = AK56 * CLAMW
         AK(N4,2) = AK(N2,4)
         AK(N4,3) = AK(N3,4)
         AK(N5,1) = AK(N1,5)
         AK(N5,3) = AK(N3,5)
         AK(N5,4) = AK(N4,5)
         AK(N6,1) = AK(N1,6)
         AK(N6,2) = AK(N2,6)
         AK(N6,4) = AK(N4,6)
         AK(N6,5) = AK(N5,6)
      IF (J .EQ. NB) GO TO 280
      IF ((J+1) .EQ. NBB(J+1)  .AND.  AB(1,J+1) .LT. A2) AATX =
     $ (ELENTH + TXJ) / TWO
280   CONTINUE
C-->>                                    E
C-->>  SET UP STIFFNESS MATRIX (K )
C
         ALAMR = ALMR
         ALAML = ALML
         ARLMSQ = ARLSQ
         ALLMSQ = ALLSQ
         ATX1 = ATX
         ATX2 = ATX
      IF (KODE .EQ. 0) GO TO 590
C
C-->>  ADJUST FOR PARTIAL BLOCKS OF THE RUNNING BOND
C
      IF (AB(1,J) .GE. A2) GO TO 580
         ATX2 = (ELENTH + TXJ) / TWO
580   IF (J .EQ. NB) GO TO 590
      IF ((J+1) .NE. NBB(J+1)) GO TO 590
      IF (AB(1,J+1) .GE. A2) GO TO 590
         ATX1 = (ELENTH + TXJ) / TWO
         ATX2 = ATX
590   CONTINUE
         BK(N1,1) = ZERO
         BK(N2,2) = ZERO
         BK(N3,3) = ZERO
      DO 600 I = 1,19,6
         BK(N1,1) = BK(N1,1) + KS(I,J)
         BK(N2,2) = BK(N2,2) + KS(I+1,J)
         BK(N3,3) = BK(N3,3) + KS(I+2,J)
600   CONTINUE
         BK(N4,4) = CLAMSQ*BK(N2,2) + BLAMSQ*BK(N3,3)
         BK(N5,5) = CLAMSQ*BK(N1,1) - ATX*AATX*BK(N3,3)
         BK(N6,6) = BLAMSQ*BK(N1,1) - ATX*AATX*BK(N2,2)
         BK(N1,5) = CLAMW*( KS(1,J)+KS(7,J)-KS(13,J)-KS(19,J))
         BK(N1,6) = BLAMV*(-KS(1,J)+KS(7,J)-KS(13,J)+KS(19,J))
         BK(N2,4) = CLAMW*(-KS(2,J)-KS(8,J)+KS(14,J)+KS(20,J))
         BK(N2,6) = - ATX1 * BK(N2,2)
         BK(N3,4) = BLAMV*(KS(3,J)-KS(9,J)+KS(15,J)-KS(21,J))
         BK(N3,5) = ATX1 * BK(N3,3)
         BK(N4,5) = ATX1 * BK(N3,4)
         BK(N4,6) = - ATX1 * BK(N2,4)
         BK(N5,6) = CLAMW*BLAMV* (- KS(1,J) + KS(7,J) + KS(13,J)
     $                          - KS(19,J))
         BK(N4,2) = BK(N2,4)
         BK(N4,3) = BK(N3,4)
         BK(N5,1) = BK(N1,5)
         BK(N5,3) = - ATX2 * BK(N3,3)
```

```
          BK(N5,4) = - ATX2 * BK(N3,4)
          BK(N6,1) = BK(N1,6)
          BK(N6,2) =   ATX2 * BK(N2,2)
          BK(N6,4) =   ATX2 * BK(N2,4)
          BK(N6,5) = BK(N5,6)
C-->>                           C
C-->>   SET UP STIFFNESS MATRIX (K  ) FOR THE HORIZONTAL STACK
C
       IF (KODE .NE. C) GO TO 777
          CK(N1,1) = KS(4,J)  + KS(16,J) + KS(25,J) + KS(37,J)
          CK(N2,2) = KS(5,J)  + KS(17,J) + KS(26,J) + KS(38,J)
          CK(N3,3) = KS(6,J)  + KS(18,J) + KS(27,J) + KS(39,J)
          CK(N4,4) = CLAMSQ*CK(N2,2) - BTSQ * CK(N3,3)
          CK(N5,5) = CLAMSQ*CK(N1,1) + ARLMSQ *(KS(6,J)+KS(18,J))
     $                + ALLMSQ * (KS(27,J) + KS(39,J))
          CK(N6,6) = ARLMSQ * (KS(5,J)+KS(17,J)) + ALLMSQ *
     $               (KS(26,J)+KS(38,J)) - BTSQ * CK(N1,1)
          CK(N1,5) = CLAMW*(KS(4,J)-KS(16,J)-KS(25,J)+KS(37,J))
          CK(N1,6) = BTY * CK(N1,1)
          CK(N2,4) = CLAMW*(-KS(5,J)+KS(17,J)+KS(26,J)-KS(38,J))
          CK(N2,6) = ALAMR*(KS(5,J)+KS(17,J))
     $                - ALAML * (KS(26,J) + KS(38,J))
          CK(N3,4) = - BTY * CK(N3,3)
          CK(N3,5) = - ALAMR*(KS(6,J)+KS(18,J))
     $                + ALAML * (KS(27,J) + KS(39,J))
          CK(N4,5) = BTY * CK(N3,5)
          CK(N4,6) = CLAMW*ALAMR*(-KS(5,J) + KS(17,J))
     $                + CLAMW*ALAML*(-KS(26,J) + KS(38,J))
          CK(N5,6) = BTY * CK(N1,5)
          CK(N4,2) =   CK(N2,4)
          CK(N4,3) = - CK(N3,4)
          CK(N5,1) =   CK(N1,5)
          CK(N5,3) =   CK(N3,5)
          CK(N5,4) = - CK(N4,5)
          CK(N6,1) = - CK(N1,6)
          CK(N6,2) =   CK(N2,6)
          CK(N6,4) =   CK(N4,6)
          CK(N6,5) = - CK(N5,6)
       GO TO 790
 777   CONTINUE
C-->>                           C
C-->>   SET UP STIFFNESS MATRIX (K  ) FOR THE RUNNING BOND
C
          ALAMR = ALMR
          ALAML = ALML
          ARLMSQ = ARLSQ
          ALLMSQ = ALLSQ
       IF (J .EQ. NLB(J)  .AND.  AB(1,J) .LT. A2) ARLMSQ = ZERO
       IF (J .EQ. NPB(J)  .AND.  AB(1,J) .GE. A2) ARLMSQ = ZERO
          CK(N1,1) = KS(4,J)  + KS(16,J)
          CK(N2,2) = KS(5,J)  + KS(17,J)
          CK(N3,3) = KS(6,J)  + KS(18,J)
          CK(N4,4) = CLAMSQ * CK(N2,2) - BTSQ * CK(N3,3)
          CK(N5,5) = CLAMSQ * CK(N1,1) - ARLMSQ * CK(N3,3)
          CK(N6,6) = - BTSQ * CK(N1,1) - ARLMSQ * CK(N2,2)
          CK(N1,5) = CLAMW * (KS(4,J) - KS(16,J))
          CK(N1,6) =    BTY * CK(N1,1)
          CK(N2,4) = - CLAMW * (KS(5,J) - KS(17,J))
          CK(N2,6) = - ALAMR * CK(N2,2)
          CK(N3,4) = - BTY * CK(N3,3)
          CK(N3,5) = ALAMR * CK(N3,3)
          CK(N4,5) = BTY * CK(N3,5)
          CK(N4,6) = - ALAMR * CK(N2,4)
          CK(N5,6) = BTY * CK(N1,5)
          CK(N4,2) =   CK(N2,4)
          CK(N4,3) = - CK(N3,4)
          CK(N5,1) =   CK(N1,5)
          CK(N5,3) = - CK(N3,5)
          CK(N5,4) =   CK(N4,5)
          CK(N6,1) = - CK(N1,6)
          CK(N6,2) = - CK(N2,6)
          CK(N6,4) = - CK(N4,6)
          CK(N6,5) = - CK(N5,6)
       IF (AB(1,J) .LT. A2) GO TO 780
       IF (J .NE. NPB(J)) GO TO 785
          CK(N2,6) = ZERO
          CK(N3,5) = ZERO
          CK(N4,5) = ZERO
          CK(N4,6) = ZERO
       GO TO 785
 780      CK(N5,3) = ZERO
          CK(N5,4) = ZERO
          CK(N6,2) = ZERO
          CK(N6,4) = ZERO
 785   CONTINUE
C-->>                           E
C-->>   SET UP STIFFNESS MATRIX (K  ) FOR THE RUNNING BOND
C
          ALAMR = ALMR
          ALAML = ALML
          ARLMSQ = ARLSQ
          ALLMSQ = ALLSQ
       IF (J .EQ. NLB(J)  .AND.  AB(1,J) .GE. A2) ALLMSQ = ZERO
       IF (J .EQ. NPB(J)  .AND.  AB(1,J) .LT. A2) ALLMSQ = ZERO
          EK(N1,1) = KS(25,J) + KS(37,J)
          EK(N2,2) = KS(26,J) + KS(38,J)
          EK(N3,3) = KS(27,J) + KS(39,J)
          EK(N4,4) = CLAMSQ * EK(N2,2) - BTSQ * EK(N3,3)
          EK(N5,5) = CLAMSQ * EK(N1,1) - ALLMSQ * EK(N3,3)
          EK(N6,6) = - BTSQ * EK(N1,1) - ALLMSC * EK(N2,2)
```

```
            EK(N1,5) = CLAMW * ( - KS(25,J) + KS(37,J))
            EK(N1,6) =    BTY * EK(N1,1)
            EK(N2,4) = CLAMW * (KS(26,J) - KS(38,J))
            EK(N2,6) =    ALAML * EK(N2,2)
            EK(N3,4) = - BTY * EK(N3,3)
            EK(N3,5) = - ALAML * EK(N3,3)
            EK(N4,5) = BTY * EK(N3,5)
            EK(N4,6) = ALAML * EK(N2,4)
            EK(N5,5) = BTY * EK(N1,5)
            EK(N4,2) =    EK(N2,4)
            EK(N4,3) = - EK(N3,4)
            EK(N5,1) =    EK(N1,5)
            EK(N5,3) = - EK(N3,5)
            EK(N5,4) =    EK(N4,5)
            EK(N6,1) = - EK(N1,6)
            EK(N6,2) = - EK(N2,6)
            EK(N6,4) = - EK(N4,6)
            EK(N6,5) = - EK(N5,6)
         IF (AB(1,J) .LT. A2) GO TO 786
         IF (J .NE. NLB(J)) GO TO 787
            EK(N2,6) = ZERO
            EK(N3,5) = ZERO
            EK(N4,5) = ZERO
            EK(N4,6) = ZERO
         GO TO 787
  786       EK(N5,3) = ZERO
            EK(N5,4) = ZERO
            EK(N6,2) = ZERO
            EK(N6,4) = ZERO
  787    CONTINUE
  780    CONTINUE
C-->>                             D
C-->>    SET UP STIFFNESS MATRIX (K  )
C
            ALAMP = ALMP
            ALAML = ALML
            ARLMSQ = ARLSQ
            ALLMSQ = ALLSQ
            ATX1 = ATX
            ATX2 = ATX
         IF (KODE .EQ. 0) GO TO 810
         IF (J .EQ. 1) GO TO 810
C
C-->>    ADJUST FOR PARTIAL BLOCKS OF THE RUNNING BOND
C
         IF (AB(1,J) .GE. A2) GO TO 800
            ATX2 = (ELENTH + TXJ) / TWO
  800    IF ((J-1) .NE. NLB(J-1)) GO TO 810
         IF (AB(1,J-1) .GE. A2) GO TO 810
            ATX1 = (ELENTH + TXJ) / TWO
            ATX2 = ATX
            AATX = (ELENTH + TXJ) / TWO
  810    CONTINUE
         IF (J .EQ. NB) GO TO 820
         IF ((J+1) .EQ. NBB(J+1)) AATX = ATX
  820    CONTINUE
            DK(N1,1) = KS(28,J) + KS(31,J) + KS(40,J) + KS(43,J)
            DK(N2,2) = KS(29,J) + KS(32,J) + KS(41,J) + KS(44,J)
            DK(N3,3) = KS(30,J) + KS(33,J) + KS(42,J) + KS(45,J)
            DK(N4,4) = CLAMSQ * DK(2,2) + BLAMSQ * DK(N3,3)
            DK(N5,5) = CLAMSQ * DK(N1,1) - ATX*AATX*DK(N3,3)
            DK(N6,6) = BLAMSQ * DK(N1,1) - ATX*AATX*DK(N2,2)
            DK(N1,5) = CLAMW*(-KS(28,J)- KS(31,J) +KS(40,J)+KS(43,J))
            DK(N1,6) = BLAMV*(-KS(28,J)+ KS(31,J) -KS(40,J)-KS(43,J))
            DK(N2,4) = CLAMW*(KS(29,J)+ KS(32,J) - KS(41,J)-KS(44,J))
            DK(N2,6) = ATX1* DK(N2,2)
            DK(N3,4) = BLAMV*(KS(30,J)- KS(33,J) + KS(42,J)-KS(45,J))
            DK(N3,5) = - ATX1* DK(N3,3)
            DK(N4,5) = - ATX1* DK(N3,4)
            DK(N4,6) = ATX1* DK(N2,4)
            DK(N5,6) = CLAMW * BLAMV * (KS(28,J) - KS(31,J) -KS(40,J)
     $                        + KS(43,J))
            DK(N4,2) = DK(N2,4)
            DK(N4,3) = DK(N3,4)
            DK(N5,1) = DK(N1,5)
            DK(N5,3) =    ATX2 * DK(N3,3)
            DK(N5,4) =    ATX2 * DK(N3,4)
            DK(N6,1) = DK(N1,6)
            DK(N6,2) = - ATX2 * DK(N2,2)
            DK(N6,4) = - ATX2 * DK(N2,4)
            DK(N6,5) = DK(N5,6)
C-->>                             E
C-->>    SET UP STIFFNESS MATRIX (K  ) FOR THE HORIZONTAL STACK
C
          IF (KODE .NE. 0) GO TO 950
C-->>    ASSIGN VALUES TO (KE) FO HORIZONTAL STACK
            EK(N1,1) = ZERO
            EK(N2,2) = ZERO
            EK(N3,3) = ZERO
          DO 900 I = 10,46,12
            EK(N1,1) = EK(N1,1) + KS(I,J)
            EK(N2,2) = EK(N2,2) + KS(I+1,J)
            EK(N3,3) = EK(N3,3) + KS(I+2,J)
  900     CONTINUE
            EK(N4,4) = CLAMSQ * EK(N2,2) - BTSQ * EK(N3,3)
            EK(N5,5) = CLAMSQ * EK(N1,1) + ARLMSQ*(KS(12,J)+KS(24,J))
     $                 + ALLMSQ * (KS(36,J) + KS(48,J))
            EK(N6,6) = ARLMSQ * (KS(11,J)+KS(23,J)) + ALLMSQ *
     $                 (KS(35,J)+KS(47,J)) - BTSQ*EK(N1,1)
            EK(N1,5) = CLAMW*(KS(10,J)- KS(22,J) - KS(34,J)+KS(46,J))
```

```
          EK(N1,6) = - BTY * EK(N1,1)
          EK(N2,4) = - CLAMR*(KS(11,J)- KS(23,J)-KS(35,J)+KS(47,J))
          EK(N2,6) =   ALAMR*(KS(11,J) + KS(23,J))
     $                - ALAML * (KS(35,J)+KS(47,J))
          EK(N3,4) = BTY * EK(N3,3)
          EK(N3,5) = - ALAMR*(KS(12,J) +KS(24,J))
     $                +ALAML*(KS(36,J)+KS(48,J))
          EK(N4,5) = - BTY * EK(N3,5)
          EK(N4,6) = CLAMW*ALAMR*(-KS(11,J) + KS(23,J))
     $              + CLAMW*ALAML*(-KS(35,J)+KS(47,J))
          EK(N5,6) = - BTY * EK(N1,5)
          EK(N4,2) =   EK(N2,4)
          EK(N4,3) = - EK(N3,4)
          EK(N5,1) =   EK(N1,5)
          EK(N5,3) =   EK(N3,5)
          EK(N5,4) = - EK(N4,5)
          EK(N6,1) = - EK(N1,6)
          EK(N6,2) =   EK(N2,6)
          EK(N6,4) =   EK(N4,6)
          EK(N6,5) = - EK(N5,6)
       GO TO 1000
  950  CONTINUE
C-->>                               F
C-->>  SET UP STIFFNESS MATRIX (K  ) FOR THE RUNNING BOND
C
          ALAMR = ALMR
          ALAML = ALML
          ARLMSQ = ARLSQ
          ALLMSQ = ALLSQ
       IF (J .EQ. NLB(J)  .AND.  AB(1,J) .GE. A2) ALLMSQ = ZERO
       IF (J .EQ. NRB(J)  .AND.  AB(1,J) .LT. A2) ALLMSQ = ZERO
          FK(N1,1) = KS(34,J) + KS(46,J)
          FK(N2,2) = KS(35,J) + KS(47,J)
          FK(N3,3) = KS(36,J) + KS(48,J)
          FK(N4,4) = CLAMSQ * FK(N2,2) - BTSQ * FK(N3,3)
          FK(N5,5) = CLAMSQ * FK(N1,1) - ALLMSQ * FK(N3,3)
          FK(N6,6) = - BTSQ * FK(N1,1) - ALLMSQ * FK(N2,2)
          FK(N1,5) = - CLAMW * (KS(34,J) - KS(46,J))
          FK(N1,6) = - BTY * FK(N1,1)
          FK(N2,4) = CLAMW * (KS(35,J) - KS(47,J))
          FK(N2,6) =   ALAML * FK(N2,2)
          FK(N3,4) =   BTY * FK(N3,3)
          FK(N3,5) = - ALAML * FK(N3,3)
          FK(N4,5) = - BTY * FK(N3,5)
          FK(N4,6) = ALAML * FK(N2,4)
          FK(N5,6) = - BTY * FK(N1,5)
          FK(N4,2) =   FK(N2,4)
          FK(N4,3) = - FK(N3,4)
          FK(N5,1) =   FK(N1,5)
          FK(N5,3) = - FK(N3,5)
```

```
          FK(N5,4) =   FK(N4,5)
          FK(N6,2) = - FK(N2,6)
          FK(N6,1) = - FK(N1,6)
          FK(N6,4) = - FK(N4,6)
          FK(N6,5) = - FK(N5,6)
       IF (AB(1,J) .LT. A2) GO TO 960
       IF (J .NE. NLB(J)) GO TO 970
          FK(N2,6) = ZERO
          FK(N3,5) = ZERO
          FK(N4,5) = ZERO
          FK(N4,6) = ZERO
       GO TO 970
  960     FK(N5,3) = ZERO
          FK(N5,4) = ZERO
          FK(N6,2) = ZERO
          FK(N6,4) = ZERO
  970  CONTINUE
C-->>                               H
C-->>  SET UP STIFFNESS MATRIX (K  ) FOR THE RUNNING BOND
C
          ALAMR = ALMR
          ALAML = ALML
          ARLMSQ = ARLSQ
          ALLMSQ = ALLSQ
       IF (J .EQ. NLB(J)  .AND.  AB(1,J) .LT. A2) ARLMSQ = ZERO
       IF (J .EQ. NRB(J)  .AND.  AB(1,J) .GE. A2) ARLMSQ = ZERO
          HK(N1,1) = KS(10,J) + KS(22,J)
          HK(N2,2) = KS(11,J) + KS(23,J)
          HK(N3,3) = KS(12,J) + KS(24,J)
          HK(N4,4) = CLAMSQ * HK(N2,2) - BTSQ * HK(N3,3)
          HK(N5,5) = CLAMSQ * HK(N1,1) - ARLMSQ * HK(N3,3)
          HK(N6,6) = - BTSQ * HK(N1,1) - ARLMSQ * HK(N2,2)
          HK(N1,5) = CLAMW * (KS(10,J) - KS(22,J))
          HK(N1,6) = - BTY * HK(N1,1)
          HK(N2,4) = CLAMW * ( - KS(11,J) + KS(23,J))
          HK(N2,6) = - ALAMR * HK(N2,2)
          HK(N3,4) =   BTY * HK(N3,3)
          HK(N3,5) = ALAMR * HK(N3,3)
          HK(N4,5) = - BTY * HK(N3,5)
          HK(N4,6) = - ALAMR * HK(N2,4)
          HK(N5,6) = - BTY * HK(N1,5)
          HK(N4,2) =   HK(N2,4)
          HK(N4,3) = - HK(N3,4)
          HK(N5,1) =   HK(N1,5)
          HK(N5,3) = - HK(N3,5)
          HK(N5,4) =   HK(N4,5)
          HK(N6,2) = - HK(N2,6)
          HK(N6,1) = - HK(N1,6)
          HK(N6,4) = - HK(N4,6)
          HK(N6,5) = - HK(N5,6)
```

```
      IF (AB(1,J) .LT. A2) GO TO 980
      IF (J .NE. NRB(J)) GO TO 990
          HK(N2,6) = ZERO
          HK(N3,5) = ZERO
          HK(N4,5) = ZERO
          HK(N4,6) = ZERO
      GO TO 990
 980      HK(N5,3) = ZERO
          HK(N5,4) = ZERO
          HK(N6,2) = ZERO
          HK(N6,4) = ZERO
 990  CONTINUE
1000  CONTINUE
      IF (TIME .NE. ZERO) GO TO 1050
      IF (AKK.EQ. ZERO) GO TO 1050
          UC(2,J) = P(2,J) / AKK
      IF (J .LE. NBXX) GO TO 450
          UC(2,J) = UC(2,J) + UC(2,J-NBXX)
      GO TO 450
1050  CONTINUE
 450  CONTINUE
   RETURN
   END
```

```
      SUBROUTINE FORCES
C
C     ****************************************************************
C     *                                                              *
C     * FUNCTION: CALCULATE THE NODAL (SPRING) FORCES FOR EACH BLOCK *
C     *                                                              *
C     ****************************************************************
C
      IMPLICIT REAL * 8 (A-H , O-Z)
      COMMON /BLOK  / A, B, C, WJ, TI, TE, EH, ELENTH
      COMMON /BORDER/ NLB(125), NRB(125)
      COMMON /CURVE / NSTAGE(48,125), STRESS( 5), STRAIN( 5), E(5), NSEG
      COMMON /GLAMKS/ KS(48,125), G(48,6), LAMR, LAML, LAMV, LAMW, AKK
      COMMON /MASLOD/ P(6,125), MASINV(6,125), BLOKWT(125)
      COMMON /NOBLOK/ AB(2,125), NB, NBX, NBY, NEX, NOB, NBXX, NNBX
      COMMON /RESULT/ FORCE(48,125), UC(6,125), POS(6,125), TIME
      COMMON /WALL  / L, L1, L2, H, H2, ILS, IRS, IUS, IBS, KODE
      DIMENSION  SIGN(48), NOEF(48), NOCH(48), LAM(2)
      DIMENSION DISP(48,125), STFORC(4,125), UN(48,125)
      DOUBLE PRECISION LAM, L, L1, L2
      DOUBLE PRECISION KS, MASINV, LAMR, LAML, LAMV, LAMW
      DATA  ZERO, ONE, TWO /0.0D00, 1.0D00, 2.0D00/
      DATA  THREE, FOUR/3.0D00, 4.0D00/
      DATA  NOEF / 24*0,25,26,27,6*0,34,35,36,37,38,39,6*0,46,47,48 /
      DATA  NOCH /3*0,4,5,6,3*0,10,11,12,3*0,16,17,18,3*0,22,23,24,24*0/
C
C-->>   DETERMINE NODAL DISPLACEMENTS FROM CENTROIDAL DISPLACEMENTS;
C-->>   THROUGH USE OF GEOMETRY (TRANSFORMATION) MATRIX : G
C
              A2 = TWO * A
              NCALL = 0
C
C-->>   CALL SUBROUTINE "GEOMET" TO FORMULATE THE (G) MATRIX
C-->>   FOR THE HORIZONTAL STACK
C
          IF (KODE .EQ. 0) CALL GEOMET(A, LAMR, LAML)
              KR = - 1
              KM = 2
              KN = 1
          DO 100 JB = 1,NB
          IF (KODE .EQ. 0) GO TO 105
C
C-->>   CALL SUBROUTINE "GEOMET" TO FORMULATE (G) MATRIX
C-->>   FOR THE RUNNING BOND
C
          IF (JB .NE. NLB(JB)) GO TO 101
              KR = -KR
              KM = KM - KR
              KN = KN + KR
              LAM(KM) = LAMR
```

```
               LAM(KN) = LAML
  101     CONTINUE
          IF (AB(1,JB) .GE. A2) GO TO 102
               EL2 = ELENTH / TWO
      CALL GEOMET (EL2, LAM(1), LAM(2))
               NCALL = 0
          GO TO 105
  102     IF (NCALL .EQ. 1) GO TO 105
      CALL GEOMET (A, LAM(1), LAM(2))
               NCALL = 1
  105     CONTINUE
          IF (TIME .GT. ZERO) GO TO 108
C
C-->>    INITIALIZE STATIC FORCES TO ZERO
C
          DO 107 I = 1,4
  107          STFORC(I,JB) = ZERO
          GO TO 109
  108          UC(2,JB) = ZERO
  109     CONTINUE
C
C-->>    CALCULATE NODAL DISPLACEMENTS: (UN)
C
          DO 110 IN = 1,48
               SUM        = ZERO
          DO 120 K6 = 1,6
               SUM     = SUM + G(IN,K6) * UC(K6,JB)
  120     CONTINUE
               UN(IN,JB) = SUM
          IF (JB .NE. NLB(JB)) GO TO 115
          IF (AB(1,JB) .LT. A2 .AND. IN .EQ. NCEF(IN)) UN(IN,JB) = ZERO
          GO TO 110
  115     IF (JB .NE. NRB(JB)) GO TO 110
          IF (AB(1,JB) .LT. A2 .AND. IN .EQ. NCCH(IN)) UN(IN,JB) = ZERO
  110     CONTINUE
  100     CONTINUE
C
C-->>    DETERMINE CHANGE IN SPRING LENGTHS (DIFF. BETWEEN NODAL
C-->>    DISPLACEMENTS).  STORE RESULTS IN ARRAY "DISP"
C
              NBX1 = NBX + 1
         DO 300 JNB = 1,NB
C
C-->>    DETERMINE BLOCK NUMBERS OF BLOCKS SURROUNDING A GIVEN BLOCK
C-->>    FOR USE IN CALCULATING NODAL DISPLACEMENTS "DISP"
C
              JNBX       = JNB + NBX
              JNB1       = JNB + 1
              JN1        = JNB - 1
              JNX        = JNB - NBX
          IF (KODE .EQ. 0) GO TO 170
          GO TO (145, 145, 150, 150), KODE
  145          JCX   = JNB + NBX1
               JFX   = JNX - 1
          GO TO 180
  150     IF (JNB .NE. NLB(JNB)) GO TO 165
          IF (AB(1,JNB) .GT. A2) GO TO 160
               NP = 1
               MP = 2
          GO TO 165
  160          NP = 2
               MP = 1
  165          JCX   = JNBX + NP
               JFX   = JNX - MP
               JNBX  = JCX - 1
               JNX   = JFX + 1
          GO TO 180
  170          JCX    = JNBX
               JFX    = JNX
  180     CONTINUE
          DO 200 M = 1,24,12
               P =     ONE
               S =     ONE
               Q = -   ONE
          DO 200 K = 1,3
C
C-->>    DETERMINE NODE NUMBERS FOR USE IN CALCULATING "DISP"
C
               I            = K + M - 1
          IF (KODE .EQ. 0) GO TO 205
               M1   = 46-M+K
               M2   = 37-M+K
               M3   = 22-M+K
               M4   = 16-M+K
          GO TO 207
  205          M1   = I+9
               M2   = I+3
               M3   = I+33
               M4   = I+24
  207     CONTINUE
C
C-->>    CALCULATE NODAL DISPLACEMENTS "DISP" (NET CHANGE IN SPRING
C-->>    LENGTH) AND SIGN ACCORDING TO ESTABLISHED BEAM SIGN CONVENTION
C
          IF (JNB .EQ. NRB(JNB)) GO TO 210
               DISP(I,JNB)     =    R * (UN(I,JNB) - UN(40-M+K,JNB1))
               DISP(I+6,JNB) =    R * (UN(I+6,JNB) - UN(43-M+K,JNB1))
          GO TO 220
  210          DISP(I,JNB)     =    R * UN(I,JNB)
               DISP(I+6,JNB) =    R * UN(I+6,JNB)
```

```
      IF (IRS .NE. 3) GO TO 220
         RR = - R
      IF (I .EQ. 1  .OR.  I .EQ. 13) RR = R
         DISP(I,JNB) = DISP(I,JNB) + RR * UN(I,JNB)
         DISP(I+6,JNB) = DISP(I+6,JNB) + RR * UN(I+6,JNB)
220      SIGN(I)        = - R
         SIGN(I+6)      = - R
      IF (JNB .GT. MNBX) GO TO 230
      IF (KODE .EQ. 0) GO TO 225
      IF (JNB .NE. NRB(JNB)) GO TO 225
      IF (AB(1,JNB) .GT. A2) GO TO 225
         DISP(I+3,JNB) = ZERO
      GO TO 240
225      DISP(I+3,JNB) =    Q * (UN(I+3,JNB) - UN(M1,JCX))
      GO TO 240
230      DISP(I+3,JNB) =    Q * UN(I+3,JNB)
      IF (IUS .NE. 3) GO TO 240
         QQ = - Q
      IF (I .EQ. 2  .OR.  I .EQ. 14) QQ = Q
         DISP(I+3,JNB) = DISP(I+3,JNB) + QQ * UN(I+3,JNB)
240      SIGN(I+3)      = - Q
      IF(JNB .LE. NBXX) GO TO 250
      IF (KODE .EQ. 0) GO TO 245
      IF (JNB .NE. NRB(JNB)) GO TO 245
      IF (AB(1,JNB) .GT. A2) GO TO 245
         DISP(I+9,JNB) = ZERO
      GO TO 260
245      DISP(I+9,JNB) = S*R * (UN(I+9,JNB) - UN(M2,JNX))
      GO TO 260
250      DISP(I+9,JNB) = S*R * UN(I+9,JNB)
260      SIGN(I+9)      = - S*R
         J = I + 24
      IF (JNB .GT. MNBX) GO TO 270
      IF (KODE .EQ. 0) GO TO 265
      IF (JNB .NE. NLB(JNB)) GO TO 265
      IF (AB(1,JNB) .GT. A2) GO TO 265
         DISP(J,JNB)    = ZERO
      GO TO 280
265      DISP(J,JNB)    =    Q * (UN(J,JNB) - UN(M3,JNBX))
      GO TO 280
270      DISP(J,JNB)    =    Q * UN(J,JNB)
      IF (IUS .NE. 3) GO TO 280
         QQ = - Q
      IF (J .EQ. 26  .OR.  J .EQ. 38) QQ = Q
         DISP(J,JNB) = DISP(J,JNB) + QQ * UN(J,JNB)
280      SIGN(J)        = - Q
      IF (JNB .EQ. NLB(JNB)) GO TO 290
         DISP(J+3,JNB) = S*Q * (UN(J+3,JNB) - UN(13-M+K,JN1))
         DISP(J+6,JNB) = S*Q * (UN(J+6,JNB) - UN(19-M+K,JN1))
      GO TO 310
290      DISP(J+3,JNB) = S*Q * UN(J+3,JNB)
         DISP(J+6,JNB) = S*Q * UN(J+6,JNB)
310      SIGN(J+3)      = - S*Q
         SIGN(J+6)      = - S*Q
      IF(JNB .LE. NBXX) GO TO 320
      IF (KODE .EQ. 0) GO TO 315
      IF (JNB .NE. NLB(JNB)) GO TO 315
      IF (AB(1,JNB) .GT. A2) GO TO 315
         DISP(J+9,JNB) = ZERO
      GO TO 330
315      DISP(J+9,JNB) = S*R * (UN(J+9,JNB) - UN(M4,JFX))
      GO TO 330
320      DISP(J+9,JNB) = S*R * UN(J+9,JNB)
330      SIGN(J+9)      = - S*R
      GO TO (340, 350, 200), K
340      R =  - ONE
         Q =    ONE
      GO TO 200
350      R =    ONE
         S =  - ONE
200   CONTINUE
C
C-->>   CALCULATE NODAL FORCES ON EACH BLOCK
C
      DO 450 IFC = 1,48
         FORCE(IFC,JNB) = SIGN(IFC) * KS(IFC,JNB) * DISP(IFC,JNB)
      IF (TIME .NE. ZERO) GO TO 450
C
C-->>   CALCULATE THE STATIC FORCES ON EACH BLOCK
C
         II = 6 * ((IFC+1)/6) - 1
      IF (IFC .EQ. 26  .OR.  IFC .EQ. 38) II = II + 3
      IF (IFC .EQ. 29  .OR.  IFC .EQ. 41) II = 0
         AK = ZERO
      IF (IFC .EQ. II) AK = AKK / FOUR
      IF (JNB .LE. MNBX) GO TO 400
      IF (IFC .EQ. 5  .OR.  IFC .EQ. 17) AK = ZERO
      IF (IFC .EQ. 26  .OR.  IFC .EQ. 38) AK = ZERO
400   CONTINUE
         FORCE(IFC,JNB) = SIGN(IFC) * AK  * DISP(IFC,JNB)
450   CONTINUE
      IF (TIME .NE. ZERO) GO TO 500
      IF (IUS .NE. 3) GO TO 460
      IF (JNB .LE. MNBX) GO TO 460
         FORCE(17,JNB) = BLOKWT(JNB) / FOUR - FORCE(11,JNB)
         FORCE( 5,JNB) = FORCE(17,JNB)
         FORCE(26,JNB) = FORCE(17,JNB)
         FORCE(38,JNB) = FORCE(17,JNB)
460      STFORC(1,JNB) = FORCE(17,JNB)
         STFORC(2,JNB) = FORCE(26,JNB)
```

```
              STFORC(3,JNB) = FORCE(11,JNB)
              STFORC(4,JNB) = FORCE(35,JNB)
          GO TO 300
  500     CONTINUE
C
C-->    THE FOLLOWING STATEMENTS ARE DESIGNED TO REASSIGN THE STATIC
C-->    LOAD CARRIED BY A GIVEN BLOCK TO THE REMAINING BOTTOM SPRINGS
C-->    OF THE BLOCK, SHOULD ANY OF THE ORIGINAL FOUR AXIAL BOTTOM
C-->    SPRINGS FAILS
C
C-->>   DETERMINE WHICH SPRINGS HAD FAILED
C
              I5  = NSTAGE (5,JNB) / 8
              I17 = NSTAGE(17,JNB) / 8
              I26 = NSTAGE(26,JNB) / 8
              I38 = NSTAGE(38,JNB) / 8
              I11 = NSTAGE(11,JNB) / 8
              I23 = NSTAGE(23,JNB) / 8
              I35 = NSTAGE(35,JNB) / 8
              I47 = NSTAGE(47,JNB) / 8
              SMT = I5 + I17 + I26 + I38
              SMB = I11 + I23 + I35 + I47
          IF (SMT .EQ. ONE) SMT = FOUR / THREE
          IF (SMT .EQ. ZERO) SMT = ONE
          IF (SMB .EQ. ONE) SMB = FOUR / THREE
          IF (SMB .EQ. ZERO) SMB = ONE
              FORCE( 5,JNB) = FORCE( 5,JNB) + (1-I5) *SMT*STFORC(1,JNB)
              FORCE(17,JNB) = FORCE(17,JNB) + (1-I17)*SMT*STFORC(1,JNB)
              FORCE(26,JNB) = FORCE(26,JNB) + (1-I26)*SMT*STFORC(2,JNB)
              FORCE(38,JNB) = FORCE(38,JNB) + (1-I38)*SMT*STFORC(2,JNB)
              FORCE(11,JNB) = FORCE(11,JNB) + (1-I11)*SMB*STFORC(3,JNB)
              FORCE(23,JNB) = FORCE(23,JNB) + (1-I23)*SMB*STFORC(3,JNB)
              FORCE(35,JNB) = FORCE(35,JNB) + (1-I35)*SMB*STFORC(4,JNB)
              FORCE(47,JNB) = FORCE(47,JNB) + (1-I47)*SMB*STFORC(4,JNB)
  300     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GEOMET (AA, LMR, LML)
C
C     ******************************************************************
C     *                                                                *
C     * FUNCTION: CALCULATE THE GEOMETRY MATRIX (G)                    *
C     *                                                                *
C     ******************************************************************
C
      IMPLICIT REAL * 8 (A-H , O-Z)
      COMMON /BLOK  / A, B, C, WJ, TI, TE, EH, ELENTH
      COMMON /JOINTS/ TXJ, TYJ, TLJ, TRJ, TUJ, TWJ
      COMMON /GLAMKS/ KS(48,125), G(48,6), LAMR, LAML, LAMV, LAMW, AKK
      DOUBLE PRECISION LAMR, LAML, LAMV, LAMW, LMR, LML, KS
      DATA  ZERO, ONE, TWO /0.0D00, 1.0D00, 2.0D00/
C
              ALAMR      = A  * (ONE - LMR)
              ALAML      = A  * (ONE - LML)
          IF (AA .LT. A) ALAMR = ZERO
          IF (AA .LT. A) ALAML = ZERO
              BLAMV      = B * (ONE - LAMV)
              CLAMW      = C * (ONE - LAMW)
              ATX        = AA + TXJ/TWO
              BTY        = B + TYJ / TWO
C
C-->>   FORMULATE THE G (GEOMETRY) MATRIX . .
C
          DO 100 I = 1,48
          DO 100 J = 1,6
              G(I,J)     = ZERO
  100     CONTINUE
          DO 250 I = 1,3
          DO 200 K = I,48,3
              G(K,I)     = ONE
  200     CONTINUE
  250     CONTINUE
              N          = 1
              K          = 2
          DO 300 J = 4,5
          DO 350 I = K,11,3
              G(I,J)     = - CLAMW * N
              G(I+12,J) =   CLAMW * N
              G(I+24,J) =   CLAMW * N
              G(I+36,J) = - CLAMW * N
  350     CONTINUE
              N          = -1
              K          = 1
  300     CONTINUE
              N          = 1
              K          = 1
          DO 400 J = 4,6,2
```

```
      DO 450 M = 1,2
         G(K+2,J)  = BLAMV * N
         G(K+5,J)  = BTY  * N
         G(K+8,J)  = - BLAMV * N
         G(K+11,J) = -BTY  * N
         G(K+26,J) =  BTY  * N
         G(K+29,J) =  BLAMV * N
         G(K+32,J) = -BLAMV * N
         G(K+35,J) = - BTY  *N
         K         = K + 12
450   CONTINUE
         N         = -1
         K         = -1
400   CONTINUE
         N         = 1
         K         = 2
      DO 500 J = 5,6
      DO 520 I = 1,19,6
         M         = K + I
         G(M,J)    = - ATX * N
         G(M+3,J)      = -ALAMR * N
520   CONTINUE
      DO 540 I = 25,37,12
         M         = K + I
         G(M,J)    = ALAML * N
         G(M+3,J)  = ATX * N
         G(M+6,J)  = ATX * N
         G(M+9,J)  = ALAML * N
540   CONTINUE
         N         = -1
         K         = 1
500   CONTINUE
    RETURN
     END
```

APPENDIX D

PROGRAM "WALBLAST": GUIDE FOR DATA INPUT

Cards No. 1 and 2: IDENTIFICATION OF RUN

Two alphameric cards.

20 A4
1 – 80

20 A4
1 – 80

Card No. 3: IDENTIFICATION OF PROBLEM

One alphameric card for each problem.

Program terminates execution, if NPROB (Problem Name) is blank.

NPROB	Description of Problem
A4	19 A4
1 – 4	11 – 80

Card No. 4: WALL DIMENSIONS

KODE	LF	LI	HF	HI
I1	I4	F10.0	I5	F10.0
1	5	15	20	30

where

KODE = Pattern specification. See Figure 12.
LF = Length of wall (portion in feet)
LI = Length of wall (portion in inches)
HF = Height of wall (portion in feet)
HI = Height of wall (portion in inches)

Card No. 5: SUPPORT CONDITIONS

ILS	IRS	IUS	LBS
I5	I5	I5	I5
1 – 5	10	15	20

where

ILS = Condition at left support

IRS = Condition at right support

IUS = Condition at upper support

IBS = Condition at bottom support

Specifications for all supports:

Free	- 0
Simple	- 1
Fixed	- 2
Symmetric	- 3

Card No. 6 : MASONRY UNIT INFORMATION

IBLOK	A2	B2	C2	WT	WJ	TI	TE	EH
I5	F10.0	F10.0	F10.0	F10.0	F5.0	F5.0	F5.0	F5.0
1 5	15	25	35	45	50	55	60	65

where

IBLOK = Block type:
- solid bricks - 1
- 2 - core concrete blocks - 2
- 3 - core concrete blocks - 3

A2 = Length of unit (inches)
B2 = Height of unit (inches)
C2 = Depth of unit (inches)
WT = Weight of unit (pounds)

The following items are for concrete blocks only:

WJ = Thickness of face shell (inches)
TI = Thickness of interior web (inches)
TE = Thickness of exterior web (inches)
EH = Thickness of end heel (inches)

Card No. 7: SPRING LOCATIONS

LAMR	LAML	LAMV	LAMW
F10.0	F10.0	F10.0	F10.0
1 — 10	20	30	40

where

LAMR = λ_{ur}

LAML = $\lambda_{u\ell}$

LAMV = λ_{v}

LAMW = λ_{w}

[See Equation (2.1) and Figure 3 for definitions]

Card No. 8: NUMBER OF SEGMENTS IN THE STRESS-STRAIN CURVE FOR MORTAR

NSEG
I5
1 — 5

STRESS-STRAIN CURVE COORDINATES FOR MORTAR

As many cards as the number of segments (NSEG) are needed. Each card must be in the following form:

Stress(I)		STRAIN(I)
D10.3		D10.3
1 — 10		16 — 25

Where

STRESS(I) = Stress value at the end of segment I (I=1,2,..,NSEG)

STRAIN(I) = Corresponding strain value

Subsequent Cards

Three cards follow the last card of the Stress-Strain specifications. They contain:

1. MORTAR PROPERTIES

TSBOND	SSBOND	CSMORT	MORTWT
F10.0	F10.0	F10.0	F10.0
1 10	20	30	40

where

TSBOND = Tensile bond strength for mortar (psi)
SSBOND = Shear bond strength for mortar (psi)
CSMORT = Compressive strength for mortar (psi)
MORTWT = Unit weight for mortar (pcf)

2. JOINT WIDTHS

TXJ	TYJ	TLJ	TRJ	TUJ	TBJ
F10.0	F10.0	F10.0	F10.0	F10.0	F10.0
1 10	20	30	40	50	60

where

TXJ = Thickness of interior head (vertical) joint
TYJ = Thickness of interior bed (horizontal) joint
TLJ = Thickness of left boundary joint
TRJ = Thickness of upper boundary joint
TBJ = Thickness of lower boundary joint

3. LOADING INFORMATION

TYPE	ILOAD	DPEAK	SPEAK	RTIME	CTIME	DTIME
A4	I1	F10.0	F10.0	F10.0	F10.0	F10.0
1 4	9	10 20	30	40	50	60

where

TYPE = Type of loading distribution:

Sinosoidal distribution - SINE
Uniform distribution - UNIF
Sinosoidal and uniform - COMB

ILOAD = Pressure-time history code. See Figure 18
DPEAK = Peak pressure for uniform distribution
SPEAK = Peak pressure for sinosoidal distribution
RTIME = Rise time of pressure to peak value
CTIME = Duration of constant peak pressure
DTIME = Decay time of pressure from peak value to zero

END OF RUN

A blank card must be added as the last card in the data cards.

APPENDIX E

COMPUTER PRINTOUTS: INPUT DATA AND RESULTS

DATA, MODULI OF ELASTICITY, AND SPRING STIFFNESSES FOR
PROBLEM PRESENTED IN SECTION 6.2

PROBLEM BEM1: 8-UNIT MASONRY BEAM; SIMPLY SUPPORTED; SOLID BRICKS

****
*   PROGRAM "WALBLAST"   *
*   FOR THE   *
* DYNAMIC ANALYSIS OF MASONRY WALLS   *
*   *
* PROGRAMMER: T.M. AL-ASWAD   *
*   *
*   SPRING, 1978   *
*   *
*   SCHOOL OF CIVIL ENGINEERING   *
*   OKLAHOMA STATE UNIVERSITY   *
*   STILLWATER, OKLAHOMA   *
****

******************************** D A T A ********************************

------------------------------ WALL SYSTEM ------------------------------

CLEAR LENGTH(FT--IN)		CLEAR HEIGHT(FT--IN)		PATTERN	TYPE VERSION
5	4.375	0	8.000	H. STACK	NOT APPLICABLE

------------------ SUPPORT TYPE ------------------

LEFT	RIGHT	UPPER	LOWER
SMPL	SYMM	FREE	FREE

------------------------------ BLOCKS ------------------------------

LENGTH	HEIGHT	DEPTH	I.WEB	E.WEB	FACE	HEEL	WEIGHT	TYPE
(I	N	C	H	E	S)		(LBS)	
15.625	8.000	8.000	NOT APPLICABLE				67.708	SOLID BRICK

-------------- JOINT THICKNESSES (INCHES) --------------

------INTERIOR ----

HEAD	BED
0.375	0.0

-------------BOUNDARY ------------

LEFT	RIGHT	UPPER	LOWER
0.375	0.375	0.0	0.0

-------------------- MORTAR PROPERTIES --------------------

BOND STRENGTH(PSI)		COMPR. STRENGTH	UNIT WEIGHT
TENSILE	SHEAR	(PSI)	(PCF)
90.000	110.000	3000.000	110.000

------------------------------ SPRINGS ------------------------------

- - - - - - - - - - LOCATIONS - - - - - - - - - -

LAMBDA R	LAMBDA L	LAMBDA V	LAMBDA W
0.50000	0.50000	0.50000	0.42265

STRESS-STRAIN CURVE COORDINATES:

STRESS(PSI) 0.25000+04

STRAIN(IN/IN) 0.10000-02

------------------------------ BLAST LOADING ------------------------------

TYPE	TIME CODE	UNIF. PEAK (PSI)	SINE PEAK (PSI)	RISE TIME (S	PEAK DURATION E C O N D	DECAY TIME S)
SINE	4	0.0	-0.500	0.000200	0.035000	0.0

MODULI OF ELASTICITY:

SEGMENT NO. 1   0.25000000000000D+07

******************** S P R I N G   S T I F F N E S S E S ********************

	INTERIOR JOINTS			
	FULL BLOCKS		PORTION BLOCKS	
	HEAD	BED	HEAD	BED
A X I A L	0.2500000000D+07	0.0	NOT APPLICABLE	
IN-PLANE SHEAR	0.1086956522D+07	0.0	NOT APPLICABLE	
Z-DIREC. SHEAR	0.7246376812D+06	0.0	NOT APPLICABLE	

	BOUNDARY JOINTS					
	FULL BLOCKS				PORTION BLOCKS	
	LEFT	RIGHT	UPPER	LOWER	LEFT	RIGHT
A X I A L	0.0	0.2500000000D+07	0.0	0.0	NOT APPLICABLE	
IN-PLANE SHEAR	0.1086956522D+07	0.1086956522D+07	0.0	0.0	NOT APPLICABLE	
Z-DIREC. SHEAR	0.1416085850D+07	0.7246376812D+06	0.0	0.0	NOT APPLICABLE	

DATA, MODULI OF ELASTICITY, AND SPRING STIFFNESSES FOR
PROBLEM PRESENTED IN SECTION 6.2

PROBLEM CBM1: 8-UNIT MASONRY BEAM; SIMPLY SUPPORTED; CONCRETE BLOCKS

*********************************** D A T A ***********************************

****
PROGRAM "WALBLAST"
FOR THE
DYNAMIC ANALYSIS OF MASONRY WALLS

PROGRAMMER: T.M. AL-ASWAD

SPRING, 1978

SCHOOL OF CIVIL ENGINEERING
OKLAHOMA STATE UNIVERSITY
STILLWATER, OKLAHOMA
****

------------------------------ WALL SYSTEM ------------------------------

CLEAR LENGTH(FT--IN)		CLEAR HEIGHT(FT--IN)		PATTERN	TYPE	VERSION
5	4.375	0	8.000	H. STACK	NOT APPLICABLE	

----------------- SUPPORT TYPE -----------------

LEFT	RIGHT	UPPER	LOWER
SMPL	SYMM	FREE	FREE

---------------------------------- BLOCKS ----------------------------------

LENGTH	HEIGHT	DEPTH	I.WEB	E.WEB	FACE	HEEL	WEIGHT	TYPE
	(I	N	C H	E	S)		(LBS)	
15.625	8.000	8.000	1.250	1.500	1.500	0.0	45.732	2-CORE CONC BLK

-------------- JOINT THICKNESSES (INCHES) ----------------

------INTERIOR ----		------------BOUNDARY ------------			
HEAD	BED	LEFT	RIGHT	UPPER	LOWER
0.375	0.0	0.375	0.375	0.0	0.0

-------------------- MORTAR PROPERTIES ------------------------------

BOND STRENGTH(PSI)		COMPR. STRENGTH	UNIT WEIGHT
TENSILE	SHEAR.	(PSI)	(PCF)
90.000	110.000	3000.000	110.000

------------------------------------ SPRINGS ----------------------------------

- - - - - - - - - LOCATIONS - - - - - - - - - -

LAMBDA R	LAMBDA L	LAMBDA V	LAMBDA W
0.50000	0.50000	0.50000	0.42265

STRESS-STRAIN CURVE COORDINATES:

STRESS(PSI) 0.2500D+04

STRAIN(IN/IN) 0.1000D-02

---------------------------------- BLAST LOADING --------------------------------

TYPE	TIME CODE	UNIF. PEAK (PSI)	SINE PEAK (PSI)	RISE TIME	PEAK DURATION	DECAY TIME
				(S E	C O N D	S)
SINE	4	0.0	-0.500	0.000200	0.035000	0.0

MODULI OF ELASTICITY:

SEGMENT NO. 1   0.25000000000000D+07

** S P R I N G   S T I F F N E S S E S **

	INTERIOR JOINTS			
	FULL BLOCKS		PORTION BLOCKS	
	HEAD	BED	HEAD	BED
A X I A L	0.2500000000D+07	0.0	NOT APPLICABLE	
IN-PLANE SHEAR	0.1086956522D+07	0.0	NOT APPLICABLE	
Z-DIREC. SHEAR	0.7246376812D+06	0.0	NOT APPLICABLE	

	BOUNDARY JOINTS					
	FULL BLOCKS				PORTION BLOCKS	
	LEFT	RIGHT	UPPER	LOWER	LEFT	RIGHT
A X I A L	0.0	0.2500000000D+07	0.0	0.0	NOT APPLICABLE	
IN-PLANE SHEAR	0.1086956522D+07	0.1086956522D+07	0.0	0.0	NOT APPLICABLE	
Z-DIREC. SHEAR	0.1416085850D+07	0.7246376812D+06	0.0	0.0	NOT APPLICABLE	

DATA, MODULI OF ELASTICITY, AND SPRING STIFFNESSES FOR PROBLEM PRESENTED IN SECTION 6.3

PROBLEM HSW1: HORIZONTAL STACK WALL; SIMPLY SUPPORTED ON ALL SIDES

PROGRAM "WALBLAST"
FOR THE
DYNAMIC ANALYSIS OF MASONRY WALLS
PROGRAMMER: T.M. AL-ASAAD
SPRING, 1978
SCHOOL OF CIVIL ENGINEERING
OKLAHOMA STATE UNIVERSITY
STILLWATER, OKLAHOMA

D A T A

WALL SYSTEM

CLEAR LENGTH(FT--IN)		CLEAR HEIGHT(FT--IN)		PATTERN	TYPE	VERSION
4	0.375	2	8.375	H. STACK	NOT APPLICABLE	

SUPPORT TYPE

LEFT	RIGHT	UPPER	LOWER
SMPL	SYMM	SYMM	SMPL

BLOCKS

LENGTH	HEIGHT	DEPTH	I.WEB	E.WEB	FACE	HEEL	WEIGHT (LBS)	TYPE
(I	N	C	H	E	S)			
15.625	7.625	4.000	NOT APPLICABLE				32.267	SOLID BRICK

JOINT THICKNESSES (INCHES)

INTERIOR HEAD	INTERIOR BED	BOUNDARY LEFT	BOUNDARY RIGHT	BOUNDARY UPPER	BOUNDARY LOWER
0.375	0.375	0.375	0.375	0.375	0.375

MORTAR PROPERTIES

BOND STRENGTH(PSI) TENSILE	BOND STRENGTH(PSI) SHEAR.	COMPR. STRENGTH (PSI)	UNIT WEIGHT (PCF)
115.000	140.000	5437.500	110.000

SPRINGS

LOCATIONS

LAMBDA R	LAMBDA L	LAMBDA V	LAMBDA W
0.33333	0.33333	0.33333	0.42265

STRESS-STRAIN CURVE COORDINATES:

STRESS(PSI)	0.3989D+04	0.5199D+04	0.5438D+04
STRAIN(IN/IN)	0.2070D-02	0.3263D-02	0.4088D-02

BLAST LOADING

TYPE	TIME CODE	UNIF. PEAK (PSI)	SINE PEAK (PSI)	RISE TIME	PEAK DURATION	DECAY TIME
				(S	E C O N D	S)
SINE	4	0.0	-1.000	0.000500	0.020000	0.0

MODULI OF ELASTICITY:

SEGMENT NO. 1	0.1926711686906D+07
SEGMENT NO. 2	0.1014451206223D+07
SEGMENT NO. 3	0.2894149607063D+06

S P R I N G   S T I F F N E S S E S

	INTERIOR JOINTS			
	FULL BLOCKS		PORTION BLOCKS	
	HEAD	BED	HEAD	BED
A X I A L	0.9181985383D+06	0.3763108763D+07	NOT APPLICABLE	
IN-PLANE SHEAR	0.3992167558D+06	0.1636134245D+07	NOT APPLICABLE	
Z-DIREC. SHEAR	0.6653612596D+05	0.6817226021D+05	NOT APPLICABLE	

	BOUNDARY JOINTS					
	FULL BLOCKS				PORTION BLOCKS	
	LEFT	RIGHT	UPPER	LOWER	LEFT	RIGHT
A X I A L	0.0	0.9181985383D+06	0.3763108763D+07	0.0	NOT APPLICABLE	
IN-PLANE SHEAR	0.3992167558D+06	0.3992167558D+06	0.1636134245D+07	0.1636134245D+07	NOT APPLICABLE	
Z-DIREC. SHEAR	0.1300247958D+06	0.6653612596D+05	0.6817226021D+05	0.1302395419D+06	NOT APPLICABLE	

DATA, MODULI OF ELASTICITY, AND SPRING STIFFNESSES FOR
PROBLEM PRESENTED IN SECTION 6.4

PROBLEM RBW1: RUNNING BOND WALLS SIMPLY SUPPORTED ON ALL SIDES

************************************ PROGRAM "WALELAST" FOR THE DYNAMIC ANALYSIS OF MASONRY WALLS ************************************

PROGRAMMER: T.M. AL-ASWAD

SPRING, 1978

SCHOOL OF CIVIL ENGINEERING
OKLAHOMA STATE UNIVERSITY
STILLWATER, OKLAHOMA

****

*********************************** D A T A ***********************************

---------- WALL SYSTEM ----------

CLEAR LENGTH (FT--IN)		CLEAR HEIGHT (FT--IN)		PATTERN	TYPE	VERSION
8	0.375	5	4.375	R. BOND	I	1

---------- SUPPORT TYPE ----------

LEFT	RIGHT	UPPER	LOWER
SMPL	SMPL	SMPL	SMPL

---------- BLOCKS ----------

LENGTH	HEIGHT	DEPTH	I.WEB	E.WEB	FACE	HEEL	WEIGHT (LBS)	TYPE
(I	N	C	H	E	S)			
15.625	7.625	4.000	NOT APPLICABLE				32.267	SOLID BRICK

---------- JOINT THICKNESSES (INCHES) ----------

INTERIOR: HEAD	INTERIOR: BED	BOUNDARY: LEFT	BOUNDARY: RIGHT	BOUNDARY: UPPER	BOUNDARY: LOWER
0.375	0.375	0.375	0.375	0.375	0.375

---------- MORTAR PROPERTIES ----------

BOND STRENGTH (PSI): TENSILE	BOND STRENGTH (PSI): SHEAR.	COMPR. STRENGTH (PSI)	UNIT WEIGHT (PCF)
115.000	140.000	5437.500	110.000

---------- SPRINGS ----------

- - - - - LOCATIONS - - - - -

LAMBDA R	LAMBDA L	LAMBDA V	LAMBDA W
0.49900	0.48800	0.33333	0.42265

STRESS-STRAIN CURVE COORDINATES:

STRESS(PSI)	0.3933D+04	0.5199D+04	0.5438D+04
STRAIN(IN/IN)	0.2070D-02	0.3263D-02	0.4088D-02

---------- BLAST LOADING ----------

TYPE	TIME CODE	UNIF. PEAK (PSI)	SINE PEAK (PSI)	RISE TIME (SECONDS)	PEAK DURATION (SECONDS)	DECAY TIME (SECONDS)
SINE	4	0.0	-1.000	0.000500	0.020000	0.0

MODULI OF ELASTICITY:

SEGMENT NO. 1   0.1926711686906D+07

SEGMENT NO. 2   0.1014451206223D+07

SEGMENT NO. 3   0.2894149607063D+06

** S P R I N G   S T I F F N E S S E S **

	INTERIOR JOINTS			
	FULL BLOCKS		PORTION BLOCKS	
	HEAD	BED	HEAD	BED
A X I A L	0.9181985383D+06	0.3763108763D+07	0.9181985383D+06	0.3763108763D+07
IN-PLANE SHEAR	0.3992167558D+06	0.1636134245D+07	0.3992167558D+06	0.1636134245D+07
Z-DIREC. SHEAR	0.6653612596D+05	0.4320506748D+05	0.8871483462D+05	0.6097512320D+05

	BOUNDARY JOINTS					
	FULL BLOCKS				PORTION BLOCKS	
	LEFT	RIGHT	UPPER	LOWER	LEFT	RIGHT
A X I A L	0.0	0.0	0.0	0.0	0.0	0.0
IN-PLANE SHEAR	0.3992167558D+06	0.3992167558D+06	0.1636134245D+07	0.1636134245D+07	0.3992167558D+06	0.3992167558D+06
Z-DIREC. SHEAR	0.1300247958D+06	0.1300247958D+06	0.1302395419D+06	0.1302395419D+06	0.2542275858D+06	0.2542275858D+06

Sample Results for the Horizontal Stack Wall (Section 6.3)

TIME = 0.0075000 SECONDS

	DISPLACEMENTS						VELOCITIES					
BLOCK	U	V	W	THETA	BETA	PHI	U	V	W	THETA	BETA	PHI
1	0.6950D-20	-0.1557D-20	-0.1666D-02	0.0	0.0	0.0	0.0000	-0.0000	-0.0755	0.0	0.0	0.0
2	0.1096D-19	-0.9701D-20	-0.2343D-02	-0.4298D-03	0.0	0.0	0.0000	-0.0000	-0.1092	-0.0204	0.0	0.0
3	0.1444D-19	-0.1668D-19	-0.2901D-02	-0.6263D-03	0.0	0.0	0.0000	-0.0000	-0.1320	-0.0296	0.0	0.0
4	0.1167D-19	-0.9440D-21	-0.2506D-02	0.0	0.2577D-03	0.0	0.0000	-0.0000	-0.1221	0.0	0.0158	0.0
5	0.2558D-19	-0.7155D-20	-0.6494D-02	-0.3674D-03	0.1940D-03	-0.8409D-21	0.0000	-0.0000	-0.2911	-0.0174	0.0056	-0.0000
6	0.3483D-19	-0.1355D-19	-0.8852D-02	-0.5437D-03	0.7266D-04	-0.9968D-21	0.0000	-0.0000	-0.3958	-0.0258	0.0016	-0.0000
7	0.1556D-19	-0.5670D-21	-0.3401D-02	0.0	0.4081D-03	0.0	0.0000	-0.0000	-0.1549	0.0	0.0204	0.0
8	0.3943D-19	-0.5656D-20	-0.9701D-02	-0.2473D-03	0.3229D-03	-0.8113D-21	0.0000	-0.0000	-0.4397	-0.0120	0.0145	-0.0000
9	0.5457D-19	-0.1156D-19	-0.1351D-01	-0.3686D-03	0.1217D-03	-0.9412D-21	0.0000	-0.0000	-0.6144	-0.0175	0.0040	-0.0000
10	0.1774D-19	-0.4754D-21	-0.3941D-02	0.0	0.4826D-03	0.0	0.0000	-0.0000	-0.1828	0.0	0.0219	0.0
11	0.4668D-19	-0.5079D-20	-0.1145D-01	-0.8720D-04	0.3905D-03	-0.5356D-21	0.0000	-0.0000	-0.5226	-0.0043	0.0203	-0.0000
12	0.6509D-19	-0.1064D-19	-0.1606D-01	-0.1303D-03	0.1471D-03	-0.5603D-21	0.0000	-0.0000	-0.7293	-0.0061	0.0091	-0.0000

Sample Results for the Running Bond Wall (Section 6.4)

TIME = 0.0075000 SECONDS

	DISPLACEMENTS						VELOCITIES					
BLOCK	U	V	W	THETA	BETA	PHI	U	V	W	THETA	BETA	PHI
1	0.8088D-20	0.3510D-21	-0.1200D-02	0.0	0.0	0.0	-0.0000	0.0000	0.0097	0.0	0.0	0.0
2	0.1038D-19	-0.6499D-20	-0.1904D-02	-0.3365D-03	0.0	0.0	-0.0000	-0.0000	0.0058	0.0009	0.0	0.0
3	0.1355D-19	-0.1278D-19	-0.2414D-02	-0.5002D-03	0.0	0.0	-0.0000	-0.0000	-0.0007	0.0000	0.0	0.0
4	0.1399D-19	-0.1547D-19	-0.2372D-02	-0.4961D-03	0.0	0.0	-0.0000	-0.0000	-0.0037	0.0000	0.0	0.0
5	0.1188D-19	-0.1339D-19	-0.1735D-02	-0.3275D-03	0.0	0.0	-0.0000	-0.0000	0.0059	0.0009	0.0	0.0
6	0.8790D-20	-0.7688D-20	-0.6692D-03	0.0	0.0	0.0	-0.0000	-0.0000	0.0020	0.0	0.0	0.0
7	0.1154D-19	0.4990D-21	-0.1089D-02	0.0	0.2065D-03	0.0	-0.0000	0.0000	0.0041	0.0	-0.0004	0.0
8	0.2055D-19	-0.1934D-20	-0.3942D-02	-0.1686D-03	0.2222D-03	-0.8670D-21	-0.0000	-0.0000	0.0107	0.0001	-0.0001	-0.0000
9	0.3038D-19	-0.7757D-20	-0.7043D-02	-0.3738D-03	0.1382D-03	-0.1068D-20	-0.0000	-0.0000	-0.0141	-0.0004	0.0039	-0.0000
10	0.3443D-19	-0.1178D-19	-0.8187D-02	-0.4470D-03	-0.4420D-05	-0.9564D-21	-0.0000	-0.0000	-0.0340	-0.0009	0.0004	-0.0000
11	0.3041D-19	-0.1232D-19	-0.6894D-02	-0.3715D-03	-0.1459D-03	-0.5566D-21	-0.0000	-0.0000	-0.0111	-0.0001	-0.0026	0.0000
12	0.2057D-19	-0.9277D-20	-0.3704D-02	-0.1698D-03	-0.2234D-03	-0.8160D-22	-0.0000	-0.0000	0.0086	0.0004	-0.0005	0.0000
13	0.1148D-19	-0.6406D-20	-0.9071D-03	0.0	-0.1952D-03	0.0	-0.0000	-0.0000	0.0013	0.0	0.0006	0.0
14	0.2143D-19	-0.2848D-21	-0.2906D-02	0.0	0.3497D-03	0.0	-0.0000	-0.0000	0.0024	0.0	0.0018	0.0
15	0.3926D-19	-0.4271D-20	-0.8579D-02	-0.1897D-03	0.3100D-03	-0.4450D-21	-0.0000	-0.0000	-0.0177	-0.0004	0.0051	-0.0000
16	0.5126D-19	-0.8447D-20	-0.1221D-01	-0.2880D-03	0.1165D-03	-0.4617D-21	-0.0000	-0.0000	-0.0709	-0.0011	0.0030	-0.0000
17	0.5063D-19	-0.1027D-19	-0.1213D-01	-0.2880D-03	-0.1252D-03	-0.3379D-21	-0.0000	-0.0000	-0.0685	-0.0009	-0.0036	0.0000
18	0.3771D-19	-0.9012D-20	-0.8411D-02	-0.1909D-03	-0.3117D-03	-0.1222D-21	-0.0000	-0.0000	-0.0092	-0.0000	-0.0043	0.0000
19	0.1998D-19	-0.5327D-20	-0.2807D-02	0.0	-0.3404D-03	0.0	-0.0000	-0.0000	0.0044	0.0	0.0005	0.0
20	0.1758D-19	-0.1574D-20	-0.1606D-02	0.0	0.3704D-03	0.0	-0.0000	-0.0000	-0.0011	0.0	0.0006	0.0
21	0.3537D-19	-0.2940D-20	-0.6848D-02	-0.3943D-04	0.4267D-03	-0.2931D-21	-0.0000	-0.0000	-0.0054	-0.0000	0.0035	-0.0000
22	0.5489D-19	-0.6267D-20	-0.1282D-01	-0.8733D-04	0.2740D-03	-0.2789D-21	-0.0000	-0.0000	-0.0665	-0.0003	0.0066	-0.0000
23	0.6211D-19	-0.7941D-20	-0.1510D-01	-0.1055D-03	-0.4316D-05	-0.1977D-21	-0.0000	-0.0000	-0.1149	-0.0004	0.0002	0.0000
24	0.5340D-19	-0.7356D-20	-0.1270D-01	-0.8824D-04	-0.2783D-03	-0.5341D-22	-0.0000	-0.0000	-0.0635	-0.0000	-0.0063	0.0000
25	0.3361D-19	-0.4839D-20	-0.6731D-02	-0.4058D-04	-0.4218D-03	0.8017D-22	-0.0000	-0.0000	0.0037	0.0001	-0.0028	0.0000
26	0.1604D-19	-0.3314D-20	-0.1576D-02	0.0	-0.3613D-03	0.0	-0.0000	-0.0000	0.0078	0.0	0.0014	0.0
27	0.2444D-19	-0.3480D-20	-0.3382D-02	0.0	0.4139D-03	0.0	-0.0000	-0.0000	0.0003	0.0	0.0025	0.0
28	0.4578D-19	-0.5676D-20	-0.1014D-01	0.6834D-04	0.3733D-03	0.2657D-22	-0.0000	-0.0000	-0.0298	0.0004	0.0049	-0.0000
29	0.6034D-19	-0.6763D-20	-0.1452D-01	0.1024D-03	0.1424D-03	0.1581D-21	-0.0000	-0.0000	-0.1003	0.0006	0.0021	-0.0000
30	0.5982D-19	-0.5954D-20	-0.1446D-01	0.1019D-03	-0.1491D-03	0.2459D-21	-0.0000	-0.0000	-0.0928	0.0007	-0.0055	0.0000
31	0.4404D-19	-0.3586D-20	-0.1002D-01	0.6754D-04	-0.3730D-03	0.2591D-21	-0.0000	-0.0000	-0.0225	0.0003	-0.0052	0.0000
32	0.2268D-19	-0.1403D-20	-0.3332D-02	0.0	-0.4054D-03	0.0	-0.0000	-0.0000	0.0023	0.0	0.0007	0.0
33	0.1561D-19	-0.4403D-20	-0.1382D-02	0.0	0.3117D-03	0.0	-0.0000	-0.0000	0.0038	0.0	-0.0007	0.0
34	0.3048D-19	-0.6142D-20	-0.5780D-02	0.1158D-03	0.3566D-03	0.1083D-21	-0.0000	+0.0000	0.0033	0.0004	0.0020	-0.0000
35	0.4653D-19	-0.6814D-20	-0.1076D-01	0.2527D-03	0.2279D-03	0.2964D-21	-0.0000	-0.0000	-0.0450	0.0011	0.0060	-0.0000
36	0.5260D-19	-0.5596D-20	-0.1266D-01	0.3016D-03	-0.3460D-05	0.4281D-21	-0.0000	-0.0000	-0.0324	0.0014	-0.0002	0.0000
37	0.4540D-19	-0.2908D-20	-0.1066D-01	0.2516D-03	-0.2319D-03	0.4197D-21	-0.0000	-0.0000	-0.0380	0.0007	-0.0055	0.0000
38	0.2835D-19	0.2262D-21	-0.5670D-02	0.1156D-03	-0.3536D-03	0.3160D-21	-0.0000	0.0000	0.0016	-0.0001	-0.0014	0.0000
39	0.1407D-19	0.3050D-21	-0.1335D-02	0.0	-0.3047D-03	0.0	-0.0000	0.0000	0.0035	0.0	0.0005	0.0
40	0.1570D-19	-0.5789D-20	-0.1994D-02	0.0	0.2209D-03	0.0	-0.0000	-0.0000	0.0074	0.0	-0.0014	0.0
41	0.2636D-19	-0.7137D-20	-0.5592D-02	0.2895D-03	0.1929D-03	0.5825D-21	-0.0000	-0.0000	0.0038	0.0001	0.0012	-0.0000
42	0.3357D-19	-0.6096D-20	-0.7863D-02	0.4290D-03	0.7268D-04	0.9008D-21	-0.0000	-0.0000	-0.0280	0.0011	0.0012	-0.0000
43	0.3303D-19	-0.3403D-20	-0.7835D-02	0.4275D-03	-0.7838D-04	0.9764D-21	-0.0000	-0.0000	-0.0327	0.0009	-0.0008	0.0000
44	0.2505D-19	-0.3163D-23	-0.5471D-02	0.2867D-03	-0.1955D-03	0.7793D-21	-0.0000	-0.0000	0.0006	-0.0001	-0.0008	0.0000
45	0.1429D-19	0.1496D-20	-0.1866D-02	0.0	-0.2185D-03	0.0	-0.0000	0.0000	0.0044	0.0	-0.0010	0.0
46	0.9728D-20	-0.5081D-20	-0.6386D-03	0.0	0.0	0.0	-0.0000	-0.0000	-0.0040	0.0	0.0	0.0
47	0.1028D-19	-0.7240D-20	-0.1312D-02	0.2224D-03	0.0	0.0	-0.0000	-0.0000	0.0037	-0.0006	0.0	0.0
48	0.1303D-19	-0.6877D-20	-0.2165D-02	0.4383D-03	0.0	0.0	-0.0000	-0.0000	0.0005	-0.0001	0.0	0.0
49	0.1404D-19	-0.4609D-20	-0.2495D-02	0.5157D-03	0.0	0.0	-0.0000	-0.0000	-0.0062	0.0004	0.0	0.0
50	0.1240D-19	-0.1203D-20	-0.2119D-02	0.4332D-03	0.0	0.0	-0.0000	-0.0000	0.0016	-0.0002	0.0	0.0
51	0.9467D-20	0.1984D-20	-0.1188D-02	0.2146D-03	0.0	0.0	-0.0000	0.0000	0.0064	-0.0006	0.0	0.0
52	0.7982D-20	0.1384D-20	-0.3106D-03	0.0	0.0	0.0	-0.0000	0.0000	-0.0077	0.0	0.0	0.0

VITA[2]

Tahseen Michael Al-Aswad

Candidate for the Degree of

Doctor of Philosophy

Thesis: RESPONSE OF MASONRY WALLS TO BLAST LOADING: A DISCRETE ELEMENT ANALYSIS

Major Field: Civil Engineering

Biographical:

Personal Data: Born in Mosul, Iraq, on October 13, 1947, the son of Michael Al-Aswad and Mary Barsoum.

Education: Graduated from Al-Nidhal Secondary School, Baghdad, Iraq, in July, 1964; received the High Diploma in Civil Engineering from the Higher Industrial Engineering Institute of the University of Baghdad in July, 1968; received the Bachelor of Science in Engineering degree from Walla Walla College, College Place, Washington, in June, 1970; received the Master of Science degree in Civil Engineering from Tennessee Technological University, Cookeville, Tennessee, in August, 1973; completed requirements for the Doctor of Philosophy degree at Oklahoma State University, Stillwater, Oklahoma, in May, 1979.

Professional Experience: Graduate assistant in research and teaching, Department of Civil Engineering, Tennessee Technological University, January, 1971, to June, 1972; surveyor with Barge, Waggoner, Sumner, and Cannon, Nashville, Tennessee, January, 1972, to June, 1972; engineer's assistant for bridge design with the Oklahoma Department of Highways, June, 1973, to August, 1973; graduate teaching assistant, School of Civil Engineering, Oklahoma State University, August, 1972, to June, 1976, and August, 1977, to June, 1978.

Membership in Professional Organizations: Iraqi Engineers Society, American Society of Civil Engineers, American Concrete Institute, Chi Epsilon.