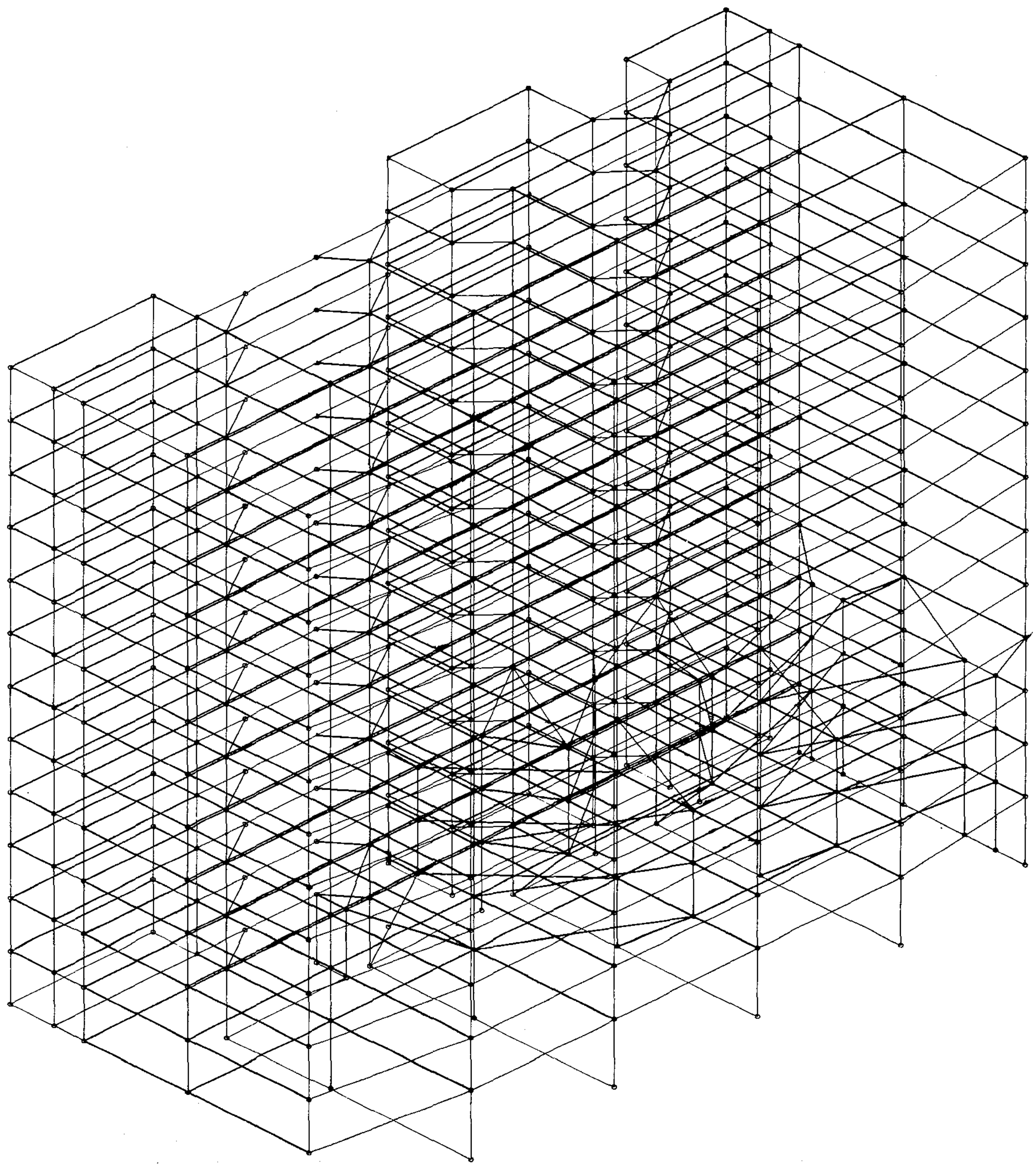
EARTHQUAKE RESPONSE AND DAMAGE PREDICTION OF REINFORCED CONCRETE MASONRY MULTISTORY BUILDINGS



MODIFICATION OF NONSAP FOR KINEMATIC INPUT

University of California, San Diego

BIBLIOGRAPHIC DATA SHEET	1. Report No. NSF-RA-E-75-218	2.	3. Recipient's Accession No.

4. Title and Subtitle
Earthquake Response and Damage Prediction of Reinforced Concrete Masonry Multistory Buildings: Modification of NONSAP for Kinematic Input

5. Report Date
August 1975

6.

7. Author(s)
G. Krishnamoorthy, C-H. Ho, R. Nunn

8. Performing Organization Rept. No. AMES-NSF TR-75-3

9. Performing Organization Name and Address
University of California, San Diego
Department of Applied Mechanics and Engineering Sciences
La Jolla, CA 92093

10. Project/Task/Work Unit No.

11. Contract/Grant No.
ATA7414818

12. Sponsoring Organization Name and Address
Research Applied to National Needs (RANN)
National Science Foundation
Washington, D.C. 20550

13. Type of Report & Period Covered

14.

15. Supplementary Notes

16. Abstracts

A numerical procedure is developed to obtain the response of a finite element-decomposed structure subjected to a time-dependent kinematic boundary history. The procedure is incorporated into the nonlinear finite element program NONSAP. The validity of the procedure is exemplified by numerical examples. The program NONSAP forms the foundation for current development work concerning the simulated response of reinforced concrete masonry assemblies and multistory structures. The input information necessary to use this computer program is included in the Appendix.

17. Key Words and Document Analysis. 17a. Descriptors

Computer programs
Kinematics
Concrete structures
Masonry
Construction
Earthquake resistant structures
Earthquakes
Reinforced concrete
Buildings
Durability

17b. Identifiers/Open-Ended Terms

NONSAP

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
NTIS	UNCLASSIFIED	33
	20. Security Class (This Page) UNCLASSIFIED	22. Price PC A03-A01

UNIVERSITY OF CALIFORNIA, SAN DIEGO
Department of Applied Mechanics and Engineering Sciences
La Jolla, California 92093

MODIFICATION OF NONSAP FOR KINEMATIC INPUT

by

G. Krishnamoorthy, C-H. Ho, and R. Nunn

August 1975
Report No. AMES-NSF TR-75-3

Prepared for the

NATIONAL SCIENCE FOUNDATION
Grant NSF-ATA 74-14818

1a

MODIFICATION OF NONSAP FOR KINEMATIC INPUT[†]

by

G. Krishnamoorthy[1], C-H. Ho[2], and R. Nunn[3]

ABSTRACT

A theoretical and numerical procedure for incorporating kinematic boundary conditions into nonlinear finite element analysis is described. The nonlinear finite element code NONSAP is modified to include the proposed method. Numerical examples are given in an effort to substantiate the developed procedures. The input information necessary to use the resulting computer program is included in the Appendix.

---

† Research was sponsored by the National Science Foundation under Grant NSF-ATA 74-14818.

1 Professor of Civil Engineering, San Diego State University, San Diego, California.

2 Postgraduate Research Engineer, Department of Applied Mechanics and Engineering Sciences, University of California, San Diego, La Jolla, California.

3 Research Assistant, Department of Applied Mechanics and Engineering Sciences, University of California, San Diego, La Jolla, California.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Most numerical codes based upon the finite element approach provide only for stress boundary conditions. In many important structural problems, however, it is necessary to specify kinematic, or mixed stress and kinematic boundary conditions.

In this report a numerical procedure is developed to obtain the response of a finite element-decomposed structure subjected to a time-dependent kinematic boundary history. The procedure is incorporated into the nonlinear finite element program NONSAP [1]. The validity of the procedure is exemplified by numerical examples.

The program NONSAP noted above forms the foundation for current development work concerning the simulated response of reinforced concrete masonry assemblies and multistory structures. The program modification described herein is only one of many currently being made to NONSAP.

2. THEORETICAL FORMULATION AND NUMERICAL TIME INTEGRATION

The incremental nodal point equilibrium equations for an assemblage of nonlinear finite elements, including velocity-dependent damping forces are [1, 2]:

$$M\ddot{u}_{t+\Delta t} + C\dot{u}_{t+\Delta t} + K_t u = R_{t+\Delta t} - F_t \quad , \tag{1}$$

in which

M	=	mass matrix
C	=	damping matrix
K_t	=	tangent stiffness matrix at time t
$R_{t+\Delta t}$	=	vector of externally applied forces at time $t + \Delta t$
F_t	=	vector of nodal point forces equivalent to the stresses of the elements at time t
$\ddot{u}_{t+\Delta t}$, $\dot{u}_{t+\Delta t}$		vector of nodal point acceleration and velocities at time $t + \Delta t$
u	=	vector of nodal point displacement increments from time t to time $t + \Delta t$; i.e., $\Delta u = u_{t+\Delta t} - u_t$.

It will be assumed that the mass matrix is constant at all times, except that if a mass is known to break off from the structure, its effect could be removed. The value of the structural stiffness matrix is assumed to be the tangent at the beginning of the time-step. If the system includes viscous dampers, then effects can be included in the damping matrix.

In general, the solution of Eq. (2) yields approximate displacement increments u, as the stiffness matrix K and vector of nodal point forces F are known only at time t. The solution of u is improved by equilibrium iteration during which operation the nodal point force vector F_t is updated to $F_{t+\tau}$ by including the nonlinear effects. However, the matrix K is still the same; i.e., K_t. The accuracy of this step-by-step solution depends naturally upon the recursive scheme used to solve the system of equations. Many different schemes are currently used in practice which were used for linear systems before. The properties of the operators used in the system have been strictly established only for linear analysis and they have been successfully used in obtaining solutions in the nonlinear analysis.

It should be realized that a great deal of experimentation is required for the design of reliable time-integration operators based on consistent finite formulations in which stable constitutive relationships can be used. It is the interaction between the numerical analysis, continuum mechanics, and the correlation of experimental results that makes the development of a nonlinear analysis program for this project a rewarding challenge.

a) Wilson θ-method

In the θ-method, a linear variation of acceleration is assumed over the time increment $\tau = \theta \Delta t$ (where $\theta \geq 1.37$), and the equilibrium equations, Eq. (1), are considered at time $t + \tau$,

$$M_{t+\tau} \ddot{u} + C_{t+\tau} \dot{u} + K_t u = R_{t+\tau} - F_t \quad , \tag{2}$$

where

$$R_{t+\tau} - R_t = \theta \left(R_{t+\Delta t} - R_t \right) \quad ,$$

$$u = u_{t+\tau} - u_t \quad .$$

In order to obtain the solution at time $t + \Delta t$, the following quantities are computed

$$\dot{u}_{t+\tau} = \dot{u}_t + \frac{\tau}{2} \left(\ddot{u}_{t+\tau} + \ddot{u}_t \right) \quad , \tag{3a}$$

$$u_{t+\tau} = u_t + \tau \dot{u}_t + \frac{\tau^2}{6}\left(\ddot{u}_{t+\tau} + 2\ddot{u}_t\right) \quad , \tag{3b}$$

where u_t, $\dot{u}_t$, $\ddot{u}_t$ are known values.

The above set of equations can be solved for $\ddot{u}_{t+\tau}$ and $\dot{u}_{t+\tau}$, explicitly as shown below:

$$\ddot{u}_{t+\tau} = \frac{6}{\tau^2}\left(u_{t+\tau} - u_t\right) - \frac{6}{\tau}\dot{u}_t - 2\ddot{u}_t \quad , \tag{4a}$$

$$\dot{u}_{t+\tau} = \frac{3}{\tau}\left(u_{t+\tau} - u_t\right) - 2\dot{u}_t - \frac{\tau}{2}\ddot{u}_t \quad . \tag{4b}$$

Substituting these expressions into the dynamic incremental equilibrium equation (2), and using the expression $u = u_{t+\tau} - u_t$ and transferring terms not containing u to the right-hand side, yields

$$\hat{K}_t u = \hat{R}_{t+\tau} \quad , \tag{5}$$

where

$$\begin{aligned} \hat{K}_t &= K_t + \frac{3}{\tau}C + \frac{6}{\tau^2}M \quad , \\ \hat{R}_{t+\tau} &= R_t + \theta\left(R_{t+\Delta t} - R_t\right) + M\left[2\ddot{u}_t + \frac{6}{\tau}\dot{u}_t + \frac{6}{\tau^2}u_t\right] \\ &\quad + C\left[\frac{\tau}{2}\ddot{u}_t + 2\dot{u}_t + \frac{3}{\tau}u_t\right] - F_t \quad . \end{aligned} \tag{6}$$

The solution of Eq. (5) gives the incremental displacement vector u. The values of u are improved by the iteration procedure described in [1]. Finally, the acceleration, velocity, and displacement vectors at the desired time $t + \Delta t$ are calculated using the following interpolation technique:

$$\ddot{u}_{t+\Delta t} = \frac{6}{\theta^3 \Delta t^2} u - \frac{6}{\theta^2 \Delta t} \dot{u}_t + \left(1 - \frac{3}{\theta}\right) \ddot{u}_t \quad , \tag{7a}$$

$$\dot{u}_{t+\Delta t} = \dot{u}_t + \frac{\Delta t}{2} \left(\ddot{u}_{t+\Delta t} + \ddot{u}_t \right) \quad , \tag{7b}$$

$$u_{t+\Delta t} = u_t + \Delta t \dot{u}_t + \frac{\Delta t^2}{6} \left(\ddot{u}_{t+\Delta t} + 2 \ddot{u}_t \right) \quad . \tag{7c}$$

b. Newmark Method

In the Newmark method [3], the displacement, velocity, and acceleration vectors at time $t + \Delta t (\theta = 1)$ are expressed in the following form

$$u_{t+\Delta t} = u_t + u \quad , \tag{8a}$$

$$\ddot{u}_{t+\Delta t} = \frac{u}{\alpha \Delta t^2} - \frac{\dot{u}_t}{\alpha \Delta t} - \left(\frac{1}{2\alpha} - 1 \right) \ddot{u}_t \quad , \tag{8b}$$

$$\dot{u}_{t+\Delta t} = \dot{u}_t + (1 - \delta) \Delta t \ddot{u}_t + \delta \Delta t \ddot{u}_{t+\Delta t} \quad . \tag{8c}$$

Similar to the Wilson θ-method, the incremental equilibrium equation (2) can be then written as indicated in equation (5) by substituting equation (8) in which

$$\hat{K}_t = K_t + \frac{1}{\alpha \Delta t^2} M + \frac{\delta}{\alpha \Delta t} C \quad , \tag{9a}$$

$$\hat{R}_{t+\Delta t} = R_{t+\Delta t} + M \left[\frac{1}{\alpha \Delta t} \dot{u}_t + \left(\frac{1}{2\alpha} - 1 \right) \ddot{u}_t \right] + C \left[\left(\frac{\delta}{\alpha} - 1 \right) \dot{u}_t + \left(\frac{8}{2\alpha} - 1 \right) \Delta t \ddot{u}_t \right] - F_t \quad . \tag{9b}$$

It should be noted that the incremental displacement u, solved by this system of equations is at time $t + \Delta t$. The velocity, and acceleration at time $t + \Delta t$ are obtained by using the interpolation technique given in equation (8).

3. KINEMATIC DATA INPUT

In the case of kinematic time-history input, the values of u have to be computed according to the interpolation techniques used in the respective analyses by either the Wilson θ-method or the Newmark method. This is required in order to obtain the correct values at time $t + \Delta t$. Table 1, shows the procedures for each interpolation scheme.

4. MODIFICATION OF THE EFFECTIVE STIFFNESS MATRIX

In all the schemes, the structural stiffness matrix is assembled by a direct superposition using equilibrium equations. If a degree-of-freedom is eliminated, the corresponding equation is not retained in this structural stiffness matrix. However, if a degree-of-freedom 'p' has a specific value at time equal to t, the corresponding equation is retained. The equations can be represented in the following matrix form:

$$
\begin{array}{c}
\\ \\ \\ p^{th}\text{ row} \\ \\ \\ \\ \\
\end{array}
\overset{\qquad\qquad\qquad p^{th}\text{ col.}}{\begin{bmatrix}
X & X & 0 & 0 & X & 0 & 0 & 0 \\
 & X & X & X & X & X & 0 & 0 \\
 & & X & X & X & 0 & X & 0 \\
- & - & - & X & X & X & X & 0 \\
 & & \text{Sym} & & X & X & X & X \\
 & & & & & X & X & X \\
 & & & & & & X & X \\
 & & & & & & & X
\end{bmatrix}}
\begin{Bmatrix} u_1 \\ \vdots \\ \vdots \\ \bar{u}_p \\ \vdots \\ \vdots \\ \vdots \\ u_n \end{Bmatrix}
=
\begin{Bmatrix} \hat{R}_1 \\ \vdots \\ \vdots \\ \hat{R}_p \\ \vdots \\ \vdots \\ \vdots \\ \hat{R}_n \end{Bmatrix}, \quad (10)
$$

Table 1

Calculation of effective incremental displacement u.

Method	Type of input	Solve $\ddot{u}_{t+\Delta t}$ from:	Calculate u by substituting $\ddot{u}_{t+\Delta t}$ into:
Wilson θ-method $u = u_{t+\tau} - u_t$	Displacement $u_{t+\Delta t}$	Eq. (7c)	Eq. (7a)
	Velocity $\dot{u}_{t+\Delta t}$	Eq. (7b)	Eq. (7a)
	Acceleration $\ddot{u}_{t+\Delta t}$		Eq. (7a)
Newmark Method $u = u_{t+\Delta t} - u_t$	Displacement $u_{t+\Delta t}$		†Eq. (8a)
	Velocity $\dot{u}_{t+\Delta t}$	Eq. (8c)	Eq. (8b)
	Acceleration $\ddot{u}_{t+\Delta t}$		Eq. (8b)

† u is not dependent on $\ddot{u}_{t+\Delta t}$.

where $\bar{u}_p$ is the known value of the degree-of-freedom p at time t, $\hat{R}_p$ is the unknown value of the force at time t. The rest of the u vector components are unknown, and the rest of the R vector components are known.

In order to keep the number of equations the same, the above equation is rewritten in the following form. This results in decoupling the matrix for a prescribed degree-of-freedom such that the calculation of the unknown value of u_p will equal the prescribed value of $\bar{u}_p$ during the solution phase.

$$\begin{bmatrix} X & X & 0 & 0 & X & 0 & 0 & 0 \\ & X & X & 0 & X & X & 0 & 0 \\ & & X & 0 & X & 0 & X & 0 \\ & & & 1 & 0 & 0 & 0 & 0 \\ & & & & X & X & X & X \\ & & \text{Sym} & & & X & X & X \\ & & & & & & X & X \\ & & & & & & & X \end{bmatrix} \begin{Bmatrix} u_1 \\ \vdots \\ \vdots \\ u_p \\ \vdots \\ \vdots \\ \vdots \\ u_n \end{Bmatrix} = \begin{Bmatrix} \hat{R}_1 \\ \vdots \\ \vdots \\ \bar{u}_p \\ \vdots \\ \vdots \\ \vdots \\ \hat{R}_n \end{Bmatrix} - \begin{Bmatrix} 0 \\ X \\ X \\ 0 \\ X \\ X \\ X \\ 0 \end{Bmatrix} \bar{u}_p , \quad (11)$$

This scheme for modifying the effective load vector $\hat{R}$ for the specified degree-of-freedom p is

$$\begin{aligned} R_i^* &= \hat{R}_i - k_{ip}\,\bar{u}_p \qquad (\text{for } i \neq p) \\ R_p^* &= \bar{u}_p \quad . \end{aligned} \qquad (12)$$

Then the elements in the p^{th} row and the column in $\hat{K}$ are replaced by zeros except the diagonal element k_{pp}, which is set to equal unity. This procedure is repeated for all specified degrees-of-freedom. Finally, this set of equations can be expressed in a compact form

$$K^* u = R^* \tag{13}$$

where

K^* = modified effective stiffness matrix,

R^* = modified effective force vector.

5. MODIFICATION OF NONSAP FOR KINEMATIC INPUT

The existing nonlinear finite element computer program NONSAP suggests the use of pseudo high stiffness spring to approach the specified displacements. This technique may result in a poorly conditioned numerical scheme. To avoid this difficulty, the above modification is introduced into the NONSAP program. The required modification in the NONSAP program is given as follows.

1. Input phase, after reading the load data, the kinematic data are read next. The load vector and kinematic vector are in the same array and are stored on tape 3.
2. In the calculation of effective load vector, the increment of node displacement is computed and stored in the nodal increase force array in corresponding specified nodal degrees-of-freedom position.
3. In the solution subroutine COLSOL, if the reformed effective stiffness matrix is to be solved, the column vectors of the effective stiffness matrix corresponding to the specified nodal degrees-of-freedom are stored on tape 33, and then the effective stiffness matrix and effective force vector are modified. If the

solution step does not involve the re-formation of the stiffness matrix, the existing modified stiffness matrix is already triangularized. Hence the force vector can be modified by using the corresponding column vector stored on a tape 33 in the previous step.

4. In the iteration subroutine EQUIT, the corresponding location of incremental load array is set equal to zero.

6. NUMERICAL EXAMPLE AND CONCLUSIONS

In order to illustrate the use of the developed procedure for kinematic boundary conditions, a square plate as shown in Figure 1, consisting of plane-stress elements was analyzed. This plate was subjected to inplane time-dependent forces at corner nodes as shown in Figure 2. Displacement, velocity and acceleration were obtained for all nodes at every time-step.

The results obtained above were used as kinematic input at corner nodes where the forces were applied. Independent analyses were made for displacement, velocity, and acceleration inputs, respectively, as shown in Figure 2 in order to assess the reliability of the procedure.

The results are given in Tables 2 and 3, and it may be seen from these tables that the results are identical for linear range (Step 1, $t = 0.2$), and are very close within the tolerance limit prescribed for iteration for nonlinear range (Steps 2 and 3, $t = 0.4$ and 0.6). It should be noted that the Newmark method and Wilson θ-method may yield significantly different results depending on the time-step used for integration.

The analyses demonstrate that the calculation procedures are dependable for the specified kinematic boundary conditions. It must be pointed out that the nodal equilibrium equations (2) are solved at time $t + \Delta t$ in the Newmark method and at time $t + \theta \Delta t$ ($\theta \geq 1.37$) in the Wilson θ-method for the displacement increments. The Wilson θ-method interpolates backwards for the accelerations, velocities, and displacements at time $t + \Delta t$. In some nonlinear cases, the structural response at time $t + \theta \Delta t$ and $t + \Delta t$ may be significantly different. Hence, it is suggested that the Newmark method be used unless the time-step considered is extremely small.

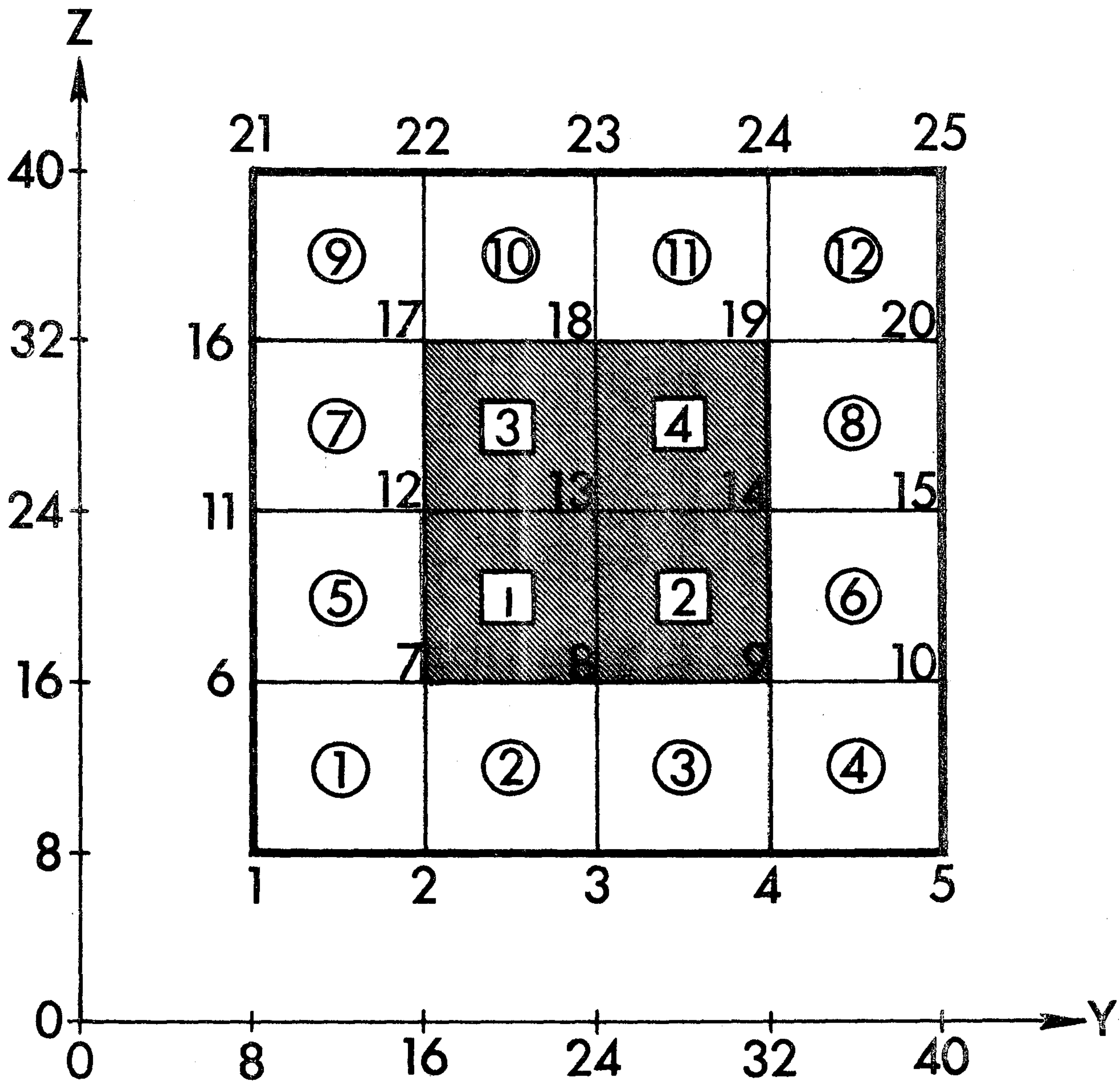


Fig. 1. The finite element grid of the rectangular plate.

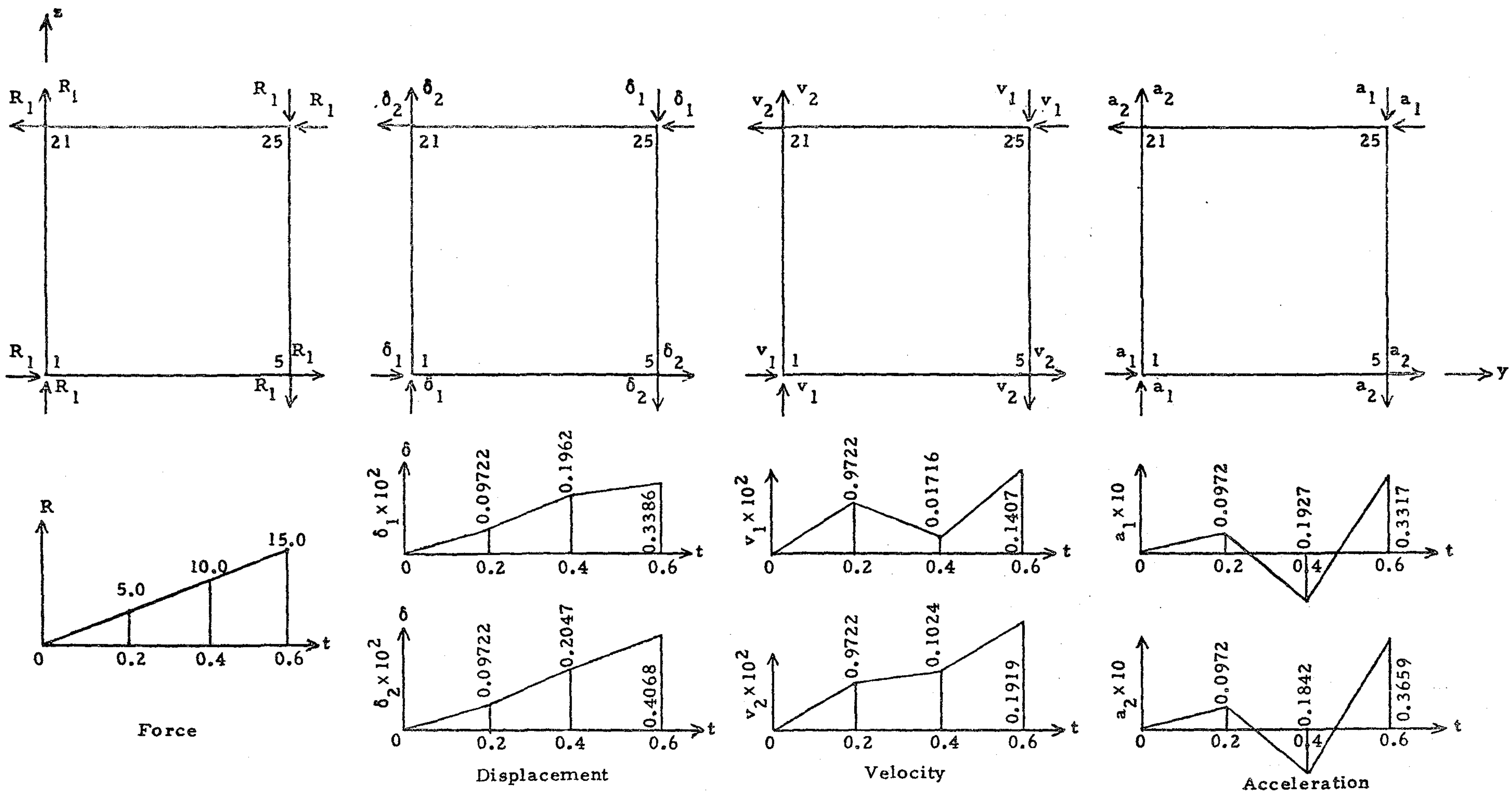


Fig. 2a. Input for Newmark method.

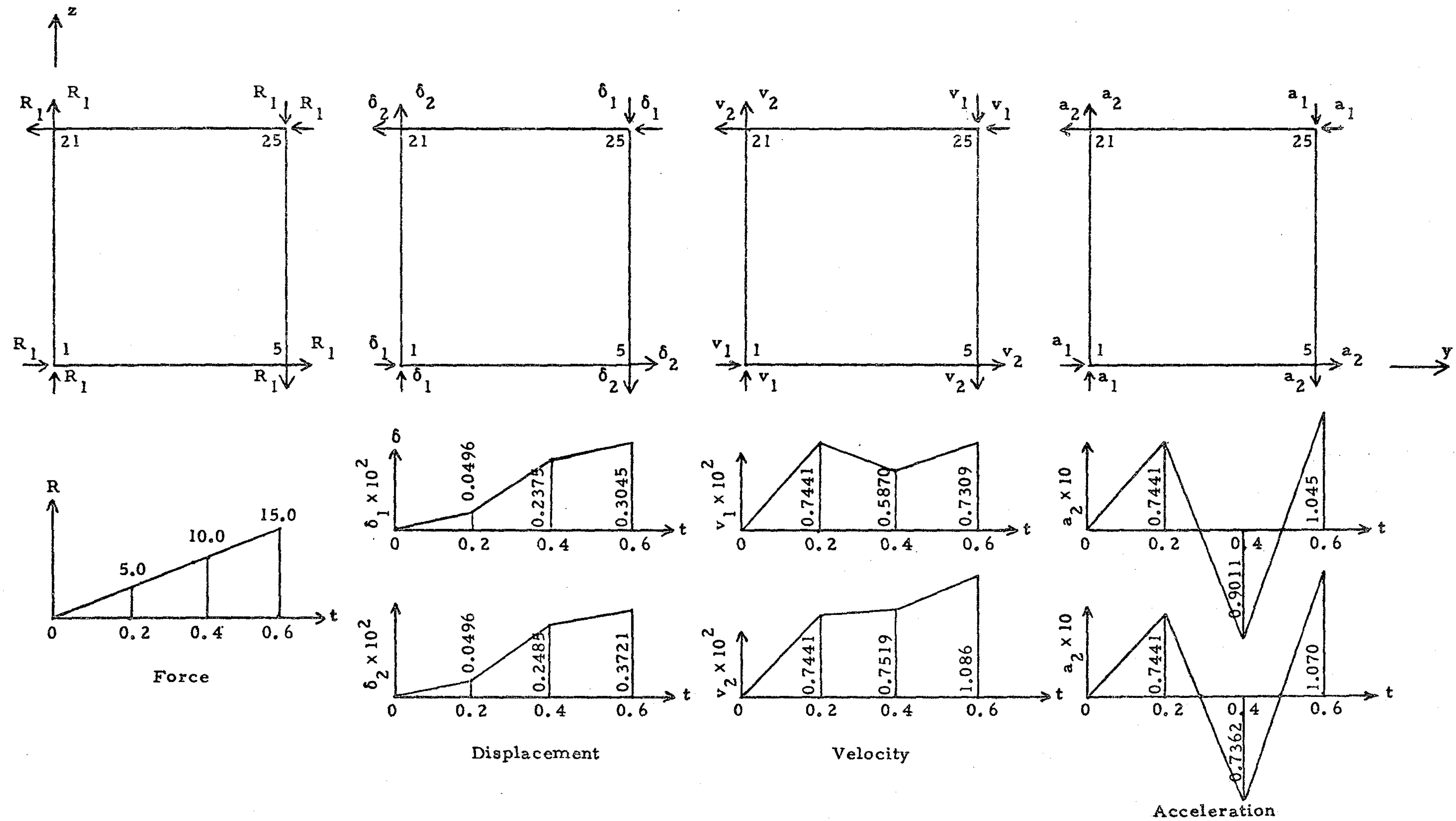


Fig. 2b. Input for Wilson θ-method.

Table 2. Displacements by the Newmark method.

Node	Time	Force Input		Displacement Input		Velocity Input		Acceleration Input	
		Y	Z	Y	Z	Y	Z	Y	Z
1	0.2	0.972229×10^{-3}	0.972229×10^{-3}	← SAME AS FORCE INPUT →					
	0.4	0.196161×10^{-2}	0.196161×10^{-2}						
	0.6	0.338597×10^{-2}	0.338597×10^{-2}						
2	0.2	0.293433×10^{-3}	0.290779×10^{-3}	0.293433×10^{-3}	0.290779×10^{-3}	0.293433×10^{-3}	0.290779×10^{-3}	0.293433×10^{-3}	0.290779×10^{-3}
	0.4	0.605136×10^{-3}	0.598197×10^{-3}	0.605107×10^{-3}	0.598169×10^{-3}	0.605108×10^{-3}	0.598170×10^{-3}	0.605109×10^{-3}	0.598171×10^{-3}
	0.6	0.121995×10^{-2}	0.121474×10^{-2}	0.122300×10^{-2}	0.121824×10^{-2}	0.122300×10^{-2}	0.121824×10^{-2}	0.122301×10^{-2}	0.121825×10^{-2}
3	0.2	0.164665×10^{-3}	0.718830×10^{-12}	0.164665×10^{-3}	0.164931×16^{-16}	0.164665×10^{-3}	0.164931×10^{-16}	0.164665×10^{-3}	0.164931×10^{-16}
	0.4	0.351836×10^{-3}	-0.111277×10^{-4}	0.351800×10^{-3}	-0.111136×10^{-3}	0.351800×10^{-3}	-0.111136×10^{-3}	0.351800×10^{-3}	-0.111128×10^{-4}
	0.6	0.834390×10^{-2}	-0.449841×10^{-3}	0.838100×10^{-3}	-0.460206×10^{-3}	0.838100×10^{-3}	-0.460207×10^{-3}	0.838100×10^{-3}	-0.460205×10^{-3}
4	0.2	0.293435×10^{-3}	-0.290779×10^{-3}	0.293433×10^{-3}	-0.290779×10^{-3}	0.293433×10^{-3}	-0.290779×10^{-3}	0.293433×10^{-3}	$-0.2902779 \times 10^{-}$
	0.4	0.629162×10^{-3}	-0.661244×10^{-3}	0.629120×10^{-3}	-0.661097×10^{-3}	0.629120×10^{-3}	-0.661098×10^{-3}	0.629119×10^{-3}	-0.661096×10^{-3}
	0.6	0.169921×10^{-2}	-0.198137×10^{-2}	0.171056×10^{-2}	-0.199530×10^{-2}	0.171056×10^{-2}	-0.199530×10^{-2}	0.171056×10^{-2}	-0.199530×10^{-2}
5	0.2	0.972229×10^{-3}	-0.972229×10^{-3}	← SAME AS FORCE INPUT →					
	0.4	0.204689×10^{-2}	-0.204689×10^{-2}						
	0.6	0.406830×10^{-2}	-0.406830×10^{-2}						
7	0.2	0.317452×10^{-3}	0.317452×10^{-3}	0.317453×10^{-3}	0.317453×10^{-3}	0.317452×10^{-3}	0.317453×10^{-3}	0.317453×10^{-3}	0.317453×10^{-3}
	0.4	0.651651×10^{-3}	0.651651×10^{-3}	0.651628×10^{-3}	0.651628×10^{-3}	0.651629×10^{-3}	0.651629×10^{-3}	0.651630×10^{-3}	0.651630×10^{-3}
	0.6	0.150374×10^{-2}	0.150374×10^{-2}	0.151267×10^{-2}	0.151267×10^{-2}	0.151267×10^{-2}	0.151267×10^{-2}	0.151268×10^{-2}	0.151268×10^{-2}
8	0.2	0.230405×10^{-3}	0.718830×10^{-12}	0.230405×10^{-3}	0.166462×10^{-16}	0.230405×10^{-3}	0.166462×10^{-16}	0.230405×10^{-3}	0.166462×10^{-16}
	0.4	0.495065×10^{-3}	0.480953×10^{-5}	0.495009×10^{-3}	0.478205×10^{-5}	0.495000×10^{-3}	0.478221×10^{-5}	0.495009×10^{-3}	0.478253×10^{-5}
	0.6	0.180376×10^{-2}	-0.632363×10^{-3}	0.182541×10^{-2}	-0.650204×10^{-3}	0.182541×10^{-3}	-0.650205×10^{-3}	0.182541×10^{-3}	-0.650203×10^{-3}
9	0.2	0.317452×10^{-3}	-0.317452×10^{-3}	0.317453×10^{-3}	-0.317453×10^{-3}	0.317453×10^{-3}	-0.317453×10^{-3}	0.317453×10^{-3}	-0.317453×10^{-3}
	0.4	0.768678×10^{-3}	-0.768678×10^{-3}	0.768379×10^{-3}	-0.768375×10^{-3}	0.768379×10^{-3}	-0.768379×10^{-3}	0.768377×10^{-3}	-0.768377×10^{-3}
	0.6	0.226183×10^{-2}	-0.226183×10^{-2}	0.227316×10^{-2}	-0.227316×10^{-2}	0.227316×10^{-2}	-0.227316×10^{-2}	0.227315×10^{-2}	-0.227315×10^{-2}

Table 3. Displacements by the Wilson θ-method.

Node	Time	Force Input		Displacement Input		Velocity Input		Acceleration Input	
		Y	Z	Y	Z	Y	Z	Y	Z
1	0.2	0.496035×10^{-3}	0.496035×10^{-3}	← SAME AS FORCE INPUT →					
	0.4	0.237546×10^{-2}	0.237546×10^{-2}						
	0.6	0.304463×10^{-2}	0.304463×10^{-2}						
2	0.2	0.149711×10^{-3}	0.148357×10^{-3}	0.149711×10^{-3}	0.148357×10^{-3}	0.149711×10^{-3}	0.148357×10^{-3}	0.149710×10^{-3}	0.148357×10^{-3}
	0.4	0.742424×10^{-3}	0.734438×10^{-3}	0.743065×10^{-3}	0.735069×10^{-3}	0.743065×10^{-3}	0.735069×10^{-3}	0.743065×10^{-3}	0.735969×10^{-3}
	0.6	0.110589×10^{-2}	0.109863×10^{-2}	0.110776×10^{-2}	0.110044×10^{-2}	0.110776×10^{-2}	0.110044×10^{-2}	0.110776×10^{-2}	0.110044×10^{-2}
3	0.2	0.840128×10^{-4}	0.203512×10^{-12}	0.840128×10^{-4}	0.113161×10^{-16}	0.840128×10^{-4}	0.104709×10^{-16}	0.840128×10^{-4}	0.967077×10^{-17}
	0.4	0.438694×10^{-3}	-0.595623×10^{-4}	0.439537×10^{-3}	-0.618080×10^{-4}	0.439536×10^{-3}	-0.618076×10^{-4}	0.439536×10^{-3}	-0.618076×10^{-4}
	0.6	0.765680×10^{-3}	-0.424063×10^{-3}	0.768167×10^{-3}	-0.430779×10^{-3}	0.768165×10^{-3}	-0.430777×10^{-3}	0.768166×10^{-3}	-0.430779×10^{-3}
4	0.2	0.149711×10^{-3}	-0.148357×10^{-3}	0.149711×10^{-3}	-0.148357×10^{-3}	0.149711×10^{-3}	-0.148357×10^{-3}	0.149710×10^{-3}	-0.148357×10^{-3}
	0.4	0.809368×10^{-3}	-0.848573×10^{-3}	0.811816×10^{-3}	-0.851918×10^{-3}	0.811815×10^{-3}	-0.851917×10^{-3}	0.811815×10^{-3}	-0.851917×10^{-3}
	0.6	0.156362×10^{-2}	-0.184281×10^{-2}	0.157089×10^{-2}	-0.185273×10^{-2}	0.157089×10^{-2}	-0.185272×10^{-2}	0.157089×10^{-2}	-0.185273×10^{-2}
5	0.2	0.496035×10^{-3}	-0.496035×10^{-3}	0.496035×10^{-3}	-0.496035×10^{-3}	0.496035×10^{-3}	-0.496035×10^{-3}	0.496035×10^{-3}	-0.496035×10^{-3}
	0.4	0.248540×10^{-2}	-0.248540×10^{-2}	0.248540×10^{-2}	-0.248540×10^{-2}	0.248540×10^{-2}	-0.248540×10^{-2}	0.248540×10^{-2}	-0.248540×10^{-2}
	0.6	0.372121×10^{-2}	-0.372121×10^{-2}	0.372121×10^{-2}	-0.372121×10^{-2}	0.372120×10^{-2}	-0.372120×10^{-2}	0.372121×10^{-2}	-0.372121×10^{-2}
7	0.2	0.161966×10^{-3}	0.161966×10^{-3}	0.161966×10^{-3}	0.161966×10^{-3}	0.161966×10^{-3}	0.161966×10^{-3}	0.161966×10^{-3}	0.161966×10^{-3}
	0.4	0.822125×10^{-3}	0.822125×10^{-3}	0.823749×10^{-3}	0.823749×10^{-3}	0.823748×10^{-3}	0.823748×10^{-3}	0.823748×10^{-3}	0.823748×10^{-3}
	0.6	0.136201×10^{-2}	0.136201×10^{-2}	0.136677×10^{-2}	0.136677×10^{-2}	0.136677×10^{-2}	0.136677×10^{-2}	0.136677×10^{-2}	0.136677×10^{-2}
8	0.2	0.117554×10^{-3}	0.203511×10^{-12}	0.117554×10^{-3}	0.107121×10^{-16}	0.117554×10^{-3}	0.933580×10^{-17}	0.117553×10^{-3}	0.916379×10^{-17}
	0.4	0.688788×10^{-3}	-0.778706×10^{-4}	0.693295×10^{-3}	-0.815129×10^{-4}	0.693294×10^{-3}	-0.815123×10^{-4}	0.693294×10^{-3}	-0.815123×10^{-4}
	0.6	0.165835×10^{-2}	-0.586154×10^{-3}	0.167170×10^{-2}	-0.597052×10^{-3}	0.167169×10^{-2}	-0.597050×10^{-3}	0.167170×10^{-2}	-0.597052×10^{-3}
9	0.2	0.161966×10^{-3}	-0.161966×10^{-3}	0.161966×10^{-3}	-0.161966×10^{-3}	0.161966×10^{-3}	-0.161966×10^{-3}	0.161966×10^{-3}	-0.161966×10^{-3}
	0.4	0.950654×10^{-3}	-0.950654×10^{-3}	0.953970×10^{-3}	-0.953970×10^{-3}	0.953969×10^{-3}	-0.953969×10^{-3}	0.953969×10^{-3}	-0.953969×10^{-3}
	0.6	0.212457×10^{-2}	-0.212457×10^{-2}	0.213436×10^{-2}	-0.213436×10^{-2}	0.213435×10^{-2}	-0.213435×10^{-2}	0.213435×10^{-2}	-0.213435×10^{-2}

REFERENCES

1. Bathe, K. J., E. L. Wilson, and R. H. Iding, "NONSAP— A Structural Analysis Program for Static and Dynamic Response of Nonlinear Systems," Dept. of Civil Engineering, University of California, Berkeley, Report No. UCSESM 74-3, February 1974.

2. Bathe, K. J., H. Ozdemir, and E. L. Wilson, "Static and Dynamic Geometric and Material Nonlinear Analysis," Dept. of Civil Engineering, University of California, Berkeley, Report No. UCSESM 74-4, February 1974.

3. Newnark, N. M., "A Method of Computation for Structural Dynamics," Proceedings ASCE 85 (EM3), pp. 67-94, 1959.

APPENDIX

INPUT INFORMATION FOR

SECTIONS IV AND VIII

OF NONSAP

IV. APPLIED LOADS AND KINEMATICS† DATA

1. Control card (7I5)

note	columns	variable	entry
(1)	1 - 5	NLOAD	Number of cards used to prescribe loads acting at the nodes
(2)	6 - 10	NLCUR	Number of load curves (time functions)
(2)	11 - 15	NPTM	Maximum number of points used to describe any one of the load curves
(3)	16 - 20	NDEFL	Number of cards used to prescribe deflections, velocities and accelerations acting on the nodes
(4)	21 - 25	NDCUR	Number of kinematic curves (time function)
(4)	26 - 30	NDPTM	Maximum number of points used to describe any one of the kinematic curves
(5)	31 - 35	IDENTY	Flag for kinematic input • EQ. 1 displacement • EQ. 2 velocity • EQ. 3 acceleration

† Kinematics may mean either displacement, velocity, or acceleration depending on the context.

NOTES/

(1) NLOAD determining the number of cards to be read in Section IV. 3, below. The loads defined in Section IV. 3 are concentrated node forces/moments that do not change direction as the structure deforms; i. e., the applied node forces are conservative loads.

(2) Time-dependent loads are applied to the structure by means of load (or time) function [i. e., f(t)] references and function multipliers assigned with the loads. At time t the value of f(t) is found by linear interpolation in the table of f(t) vs. t; f(t) times the multiplier is the magnitude of the applied load at t. NPTM is the maximum number of [f(t), t] pairs used to describe any one of the NLCUR functions; an individual function may have fewer than NPTM [f(t), t] points as input, but no function can be input with more than NPTM points. At least two points are required per function; otherwise interpolation in time is not possible.

(3) NDEFL determines the number of cards to be read in Section IV. 5, below. The kinematic quantities defined in this section are those that do not change direction as the structure deforms.

(4) Time-dependent kinematic quantities are applied to the structure by means of deflection (or time) function [i. e., $\delta(t)$] references and function multipliers assigned with the loads. At time t the value of $\delta(t)$ is found by linear interpolation in the table of $\delta(t)$ vs. t; $\delta(t)$ times the multiplier is the magnitude of the applied deflection at t. NDPTM is the maximum number of [$\delta(t)$, t] pairs used to describe any one of the NLCUR functions; an individual function may have fewer than [$\delta(t)$, t] points as input, but no function can be input with more than NDPTM points. At least two points are required per function; otherwise interpolation in time is not possible.

(5) IDENTY determines the type of prescribed kinematic input.

If set to equal 1, the prescribed input is displacement;
If set to equal 2, the prescribed input is velocity;
If set to equal 3, the prescribed input is acceleration.

IV. APPLIED LOADS AND DEFLECTION DATA (continued)

2. Load function data

Input NLCUR sets of the following data cards in order of increasing load function number.

a. Control data (2I5)

note	columns	variable	entry
	1 - 5	NTF	Time function number; GE. 1 and LE. NLCUR
	6 - 10	NPTS	Number of points (i. e., f(t), t pairs) used to input this time function; GE. 2 and LE. NPTM

b. [f(t), t] data (8F10. 0)

note	columns	variable	entry
(1)	1 - 10	TIMV(1)	Time at point 1, t_1
	11 - 20	RV (1)	Function value at point 1, $f(t_1)$
	21 - 30	TIMV(2)	Time at point 2, t_2
	31 - 40	RV (2)	Function value at point 2, $f(t_2)$
	...	...	...
	71 - 80	RV (4)	Function value at point 4, $f(t_4)$
	Next card (if required)		
(2)	1 - 10	TIMV(5)	Time at point 5, t_5
	11 - 20	RV (5)	Function value at point 5, $f(t_5)$
	...	...	...

NOTES/

(1) Time values at successive points must increase in magnitude (i. e., TIMV(1) < TIMV(2) < TIMV(3), etc.), and TIMV(1) must be equal to zero (i. e., TIMV(1). EQ. 0. 0). The last time value for the function [i. e., TIMV(NPTS)] must be greater than or equal to the time at the end of solution; i. e., TIMV(NPTS) ≥ TSART + NSTE*DT otherwise an error condition is declared.

(2) Input as many cards in this section as are required to define NPTS points, four points per card.

IV. APPLIED LOADS AND DEFLECTIONS DATA (continued)

3. Nodal Loads Data (3I5, F10.0)

Skip this section if NLOAD. EQ. 0; otherwise input NLOAD cards in this section.

note	columns	variable	entry
(1)	1 - 5	NOD	Node number to which this load is applied; GE. 1 and LE. NUMNP
	6 - 10	IDIRN	Degree-of-freedom number for this load component; EQ. 0; solution terminated EQ. 1; X-translation EQ. 2; Y-translation EQ. 3; Z-translation EQ. 4; X-rotation EQ. 5; Y-rotation EQ. 6; Z-rotation
	11 - 15	NCUR	Load curve number that describes the time dependence of the load; GE. 1 and LE. NLCUR
	16 - 25	FAC	Function multiplier used to scale f(t) for the load at "t";

NOTES/

(1) If the same degree-of-freedom (IDIRN) at the same node (NOD) is given a multiple number of times, the program combines the loads algebraically with no error diagnostic.

IV. APPLIED DEFLECTION DATA (continued)

4. Deflection function data

Input NDCUR sets of the following data cards in order of increasing deflection function number.

a. Control data (2I5)

note	columns	variable	entry
	1 - 5	NTF	Time function number; GE. 1 and LE. NDCUR
	6 - 10	NPTS	Number of points (i. e., $\delta(t)$, t pairs) used to input this time function; GE. 2 and LE. NOPTM

b. [δ (t), t] data (8F10.0)

note	columns	variable	entry
(1)	1 - 10	TIMV(1)	Time at point 1, t_1
	11 - 20	RV (1)	Function value at point 1, $\delta(t_1)$
	21 - 30	TIMV(2)	Time at point 2, t_2
	31 - 40	RV (2)	Function value at point 2, $\delta(t_2)$
	...	...	...
	71 - 80	RV (4)	Function value at point 4, $\delta(t_4)$
		Next card (if required)	
(2)	1 - 10	TIMV(5)	Time at point 5, t_5
	11 - 20	RV (5)	Function value at point 5, $\delta(t_5)$

NOTES/

(1) Time values at successive points must increase in magnitude (i. e., TIMV(1)<TIMV(2)<TIMV(3), etc.), and TIMV(1) must be equal to zero (i. e., TIMV(1). EQ. 0.0). The last time value for the function [i. e., TIMV(NDPTM)] must be greater than or equal to the time at the end of solution; i. e., TIMV(NDPTM) ≥TSTART + NSTE*DT otherwise an error condition is declared.

(2) Input as many cards in this section as are required to define points, four points per card.

IV. APPLIED DEFLECTIONS DATA (continued)

5. Nodal Deflection Data (3I5, F10.0)

Skip this section if NDEFL. EQ. 0; otherwise input NDEFL cards in this section.

note	columns	variable	entry
(1)	1 - 5	NOD	Node number to which this load is applied; GE. 1 and LE. NUMNP
	6 - 10	IDIRN	Degree-of-freedom number for this load component; EQ. 0; solution terminated EQ. 1; X-translation EQ. 2; Y-translation EQ. 3; Z-translation EQ. 4; X-rotation EQ. 5; Y-rotation EQ. 6; Z-rotation
	11 - 15	NDCUR	Load curve number that describes the time dependence of the load; GE. 1 and LE. NLCUR
	16 - 25	FAC	Function multiplier used to scale f(t) for the load at "t";

NOTES/

(1) If the same degree-of-freedom (IDIRN) at the same node (NOD) is given a multiple number of times, the program combines the deflections algebraically with no error diagnostic.

VIII. INITIAL CONDITIONS

Initial conditions for the element are defined in this section.

Initial conditions may be established using one (1) of three (3) methods.

METHOD 1—For MODEX. EQ. 2, this is a restart job. Refer to Appendix A for setting up a restart job. The variable "ICON" appearing on the card below is read by the program, but ignored; i. e., the control card (Section VIII. a) must still be input.

METHOD 2—For MODEX. NE. 2, and initial conditions of all zero, input ICON. EQ. 0 with no additional data; all vector components are then automatically initialized to zero at time of solution start, TSTART.

METHOD 3—For MODEX. NE. 2, and known non-zero initial conditions, input ICON. EQ. 1 and read the system vectors in compacted form from cards as described in Section VIII. b, below.

a) <u>Control card</u> (I5)

note	columns	variable	
(1)	1 - 5	ICON	Flag indicating the type of initial conditions; EQ. 0 and MODEX. NE. 2 with NDEFL. EQ. 0, zero initial conditions are generated automatically EQ. 1 and MODEX. NE. 2, non-zero initial conditions are read from data cards immediately following (see Section (b)) EQ. 0 and MODEX. NE. 2 with NDEFL. NE. 0, non-zero initial conditions are read from data cards immediately following (see Section (c))

b) <u>Card Input of System Vectors</u> (3F10. 0)

For the case MODEX. NE. 2 and ICON. EQ. 1, the program performs the following read operations:

READ (5, 1010) (DIS(K), VEL(K), ACC(K), K = 1, NEQ)
where DIS/VEL/ACC are the system initial displacement velocity/acceleration vectors, respectively. The variable NEQ is the total number of freedoms retained for evaluation; i. e., six (6) times the total nodes minus (-) all deletions provided by fixed boundary condition specifications.

The list of equation numbers can be obtained in Section III (variable PSF) and can be identified conveniently from the displacement (velocity and acceleration) print-out of a previous solution.

For the case of a static solution, the VEL/ACC system initial vectors are not read from card input. A static solution is performed if IMASS. EQ. 0 (Section II, card 2).

c) Card Input of System Vectors (3F10.0)

For the case MODEX. NE. 2 and ICON. EQ. 0 (i. e., Displacement input condition) read NDEFL CARD in the same order as in the displacement input.

READ (5, 1010) (DIS(K), VEL(K), ACC(K), K = 1, NDEFL)
where DIS/VEL/ACC are the system initial displacement where deflections are prescribed.