

ON THE DEVELOPMENT OF COMPUTER PROGRAMS
FOR THE STRESS ANALYSIS OF SHEAR WALLS
USING HERMITIAN INTERPOLATION

by

KI-JANG LEE

B.S., Han Yang University, 1981
Seoul, Korea

A MASTER'S THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Civil Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1986

Approved by:



Major Professor

LD
2668
T4
1986
L43
c. 2

TABLE OF CONTENTS

	Page
List of Figure	iii
List of Tables	v
List of Symbols	vi
Chapter	
[1] Introduction	1
Objectives	2
[2] Review of Past Work	3
Shear Connection Technique	3
Elastic Analysis	3
Plastic Analysis	5
Frame Method	7
Elastic Analysis	7
Plastic Analysis	8
Finite Element Method	8
Dynamic Analysis and Simulated seismic Analysis	8
[3] Method of Analysis	11
Shape functions and Degrees of Freedom	12
Strain Components	19
Calculation of Generalized Nodal Forces	27
Numerical Integration and the Use of Gaussian Quadrature	28
Stiffness of the Reinforced Steel Bar	29
[4] Numerical Experiment	31
Modulus of Elasticity	31
Results of Numerical Experiment	37

[5] The Computer Program Developed	55
Element Stiffness Matrix:Main	55
Assembling Element Stiffness Matrix:Assem	55
Reduced Stiffness Matrix:Reduc	57
Boundary Conditions of Zero Displacements :Modif	62
Boundary Conditions of Zero Stresses:Enboun	63
Solution Vector:Skyline	64
Rearranged Parameters:Reparam	65
Representation of Complete Displacement Vector:Comdiap	65
3-D Representation of Stress:Stress	65
2-D Representation of Displacement:Displace	66
3-D Representation of Strain:Strain	66
Trajectories of Stresses:Trajectory	66
[6] Conclusions	68
Suggestions for Further Research	69
Reference	70
Acknowledgments	73
Appendix List of the Programs Developed	74
Program Name: MAIN	75
ASSEM	87
REDUC	89
MODIF	93
ENBOUN	95
SKYLINE	99
REPARAM	102
CONDISP	104
STRESS	105
DISPLACE	114
STRAIN	116
TRAJECTORY	125
ENERGY	130
HERMITIAN	135

LIST OF FIGURES

	Page
1. Strain energy of a rectangular element due to nodal displacements	14
1a. X-direction movement due to nodal displacements	15
1b. Y-direction movement due to nodal displacements	16
2. The local ordering system and the shape function	18
3. First order Hermitian shape function using nodal w of node 1	20
4. First order Hermitian shape function using nodal w of node 2	20
5. First order Hermitian shape function using nodal w of node 3	21
6. First order Hermitian shape function using nodal w of node 4	21
7. First order Hermitian shape function using dw/dx of node 1	22
8. First order Hermitian shape function using dw/dx of node 2	22
9. First order Hermitian shape function using dw/dx of node 3	23
10. First order Hermitian shape function using dw/dx of node 4	23
11. First order Hermitian shape function using dw/dy of node 1	24
12. First order Hermitian shape function using dw/dy of node 2	24
13. First order Hermitian shape function using dw/dy of node 3	25
14. First order Hermitian shape function using dw/dy of node 4	25
15. Gaussian points	29
16. A steel bar of an element	29

17. Overall dimensions of model structure and elevation	32
18. Finite element representation	33
19. Shear wall type 1 and type 2	34
20. Shear wall type 3 with diagonal reinforcement	35
21. Stress-strain curve of concrete	36
22. Trajectories of stresses	40
23. Displacement of shear wall	41
24. Maximum principal stress (shear wall type 1)	42
25. Minimum principal stress (shear wall type 1)	43
26. Normal stress in x-direction (shear wall type 1)	44
27. Normal stress in y-direction (shear wall type 1)	45
28. Shearing stress with respect to x,y axes (shear wall type 1)	46
29. Stress of horizontal steel bar (shear wall type 1)	47
30. Stress of vertical steel bar (shear wall type 1)	48
31. Stress of diagonal bar (shear wall type 3)	49
32. Stress of diagonal bar (shear wall type 3)	50
33. Program flow chart	54
34. Local ordering system for a typical element	55
35. Partitioned matrix	55
36. Global stiffness matrix	56
37. Lower triangular matrix	59
38. Mohr's circle for principal stresses	66

LIST OF TABLES

	Page
1. Properties of model shear walls	31
2. Maximum strains of the model structures	51
3. Comparison of the allowable shear force between the estimated value and experimental data	52
4. Comparison of the ultimate shear force between the estimated value and experimental data	53

LIST OF SYMBOLS

- A = One dimensional array of the global stiffness matrix
 $[B]$ = Element strain-displacement matrix
 $[C]$ = Connection matrix
 $[D]$ = Elasticity matrix (matrix relating stresses and strains)
 E_C = Modulus of elasticity of concrete
 E_s = Modulus of elasticity of steel
 $\{F^e\}$ = Nodal force of an element
 f'_C = Ultimate stress of concrete
 $[K]$ = Stiffness matrix
 $[N]$ = Matrix of shape (nodal interpolation) function
 $\{u^e\}$ = Displacement vector of an element
 $\{u_i\}$ = Displacement vectors of i -th element
 $\{u_r\}$ = Reduced displacement vector
 W^e = Work done by external loads
 W_i = Gaussian weighting in x direction
 W_j = Gaussian weighting in y direction
 ϵ = Strain in element
 $\{\sigma\}$ = Element stress vector
 μ = Poisson's ratio
 τ_{xy} = Shearing stress
 γ_{xy} = Shearing strain
 Γ = Strain energy density

CHAPTER. 1

INTRODUCTION

In high rise buildings, it is important to provide the lateral resistance of the frames to the loads due to winds, earthquakes or blasts. These forces can develop high stresses, and yield lateral movements, thereby causing discomfort to the occupants. In addition to stiffening the frames due to lateral loads, the shear walls support vertical loads. Although reinforced concrete walls were often used to provide the necessary resistance to horizontal forces, modern concept of shear walls designed as vertical slender cantilevers was first used in 1948 in housing projects in New York City and in Chicago in buildings designed for wind forces.

In nonearthquake areas reinforced concrete buildings with more than 15 to 20 stories are usually designed with shear walls to improve their stiffness. Economically, the inclusion of shear walls seems to be the least expensive way to increase the overall rigidity of building structures. Shear walls serve the functions as follow: a) to support gravity loads b) to provide lateral resistance, and c) to function as a wall.

Usually most of the recent high rise reinforced concrete buildings were built without any additional cost for making the design stresses in the range of allowable stresses when lateral loads are considered by the use of the shear walls.

The high lateral rigidity was also achieved as one of the results of the shear wall-frame interaction.

OBJECTIVES

The merits of shear walls for earthquake resistance, and examinations of their performance in earthquakes were made by many researchers.

In this study, the main objectives are as follows:

- a) to suggest the use of the finite element method, a convenient numerical analysis, by which an accurate estimate may be made.
- b) to obtain experience in developing programs using the finite element method.
- c) to learn the stress distributions of the shear walls under the application of design loads.
- d) to improve the technology in shear wall design.
- e) how to reduce the data to be manageable by the micro-computer.

CHAPTER 2

REVIEW OF PAST WORK

The design of high rise buildings has developed from the bearing wall types to rigid steel frames, thus causing curtain wall buildings.

The design of shear walls has been the subject of considerable discussion since 1950's. The simplest way of designing shear walls is to consider the shear wall as an independent member and design it as a vertical cantilever. While this method has the advantage of simplicity, it is not universally valid, depending on the nature of the column and girder frame provided. The design of the shear wall to resist all lateral loads as a cantilever is very conservative with respect to the shear wall, while it may be unconservative with respect to the frame.

Gould(1) suggested a method of proportioning the external load between the shear walls and the frames so as to satisfy the requirements of deformation compatibility with a minimum of mathematical complication.

2.1 SHEAR CONNECTION TECHNIQUE

2.1.1 ELASTIC ANALYSIS

Coull and Choudhury(2) developed the curves to enable design calculations to be performed rapidly by making use of the basic assumptions of the shear connection technique. This method assumes both beams and walls in coupled shear walls deflect equally with a point of contraflexure at midspan of the connecting beams, and the discrete system of connecting

beams is replaced by a continuous medium of equivalent stiffness. The behavior of coupled shear wall is expressed as a second order differential equation.

Schwaighofer(3) modified Rosman's theory. Rosman's theory is based on the following assumptions. a) the floor-to-floor height is constant, b) the modulus of elasticity of the walls and the beams is constant, c) the cross-sectional dimensions of the shear walls and beams are constant, and d) the points of deflection of all beams are located at midspan. Replacing the discrete system of shear walls and beams by an equivalent continuous system and using the shear force along the center of the lamina as the redundant. Rosman's approach can be used but appears to be too lengthy for practical purposes. Schwaighofer therefore introduced a simplification in Rosman's theory. The characteristic curve is replaced by a straight line. This results in considerable savings of time in the analysis.

Coull(4) presented the method for the analysis of coupled shear walls supported on elastic foundations. He indicated that the general case of a foundation conditions allowing both vertical settlement and rotation to occur could be obtained by superposition of solution worked out.

Because the solutions allowing for vertical settlement of the foundation alone and for rotation of the foundation alone are only of limited application to foundations, Tso and Chan(5) presented the problem of lateral loading of coupled shear walls resting on flexible foundation.

Arvidsson(6) presented a method which took into consideration the effect of door openings near the edge of the wall. He included corrective terms in mathematical model, thus removing the limitations of applicabilities. Also he(7) presented the method for the analysis of shear walls with two rows of opening. He reduced from the high degree of static indeterminacy of the system to a fourth order differential equation.

Arvidsson(8) analysed the interaction between coupled shear walls and frames. Secondary design effects, such as column shortening in the frame and elastic foundation conditions, were also taken into account.

Coull and Low(9) presented for the analysis of non-plane coupled shear walls subjected to lateral loads which produce combined bending and torsional deformations.

Tso and Biswas(10, 11) presented the analysis of nonplanar coupled shear walls using continuum method. Both bending and torsional deformation were taken into account. The shear walls were treated as thin-walled beams of open section and the connecting beams between shear walls were replaced by an elastic medium of known stiffness properties. The total stiffness of the coupled shear wall system was then equal to the sum of the stiffness of the shear walls and the elastic medium.

2.1.2 PLASTIC ANALYSIS

Winokur and Gluck(12) presented an ultimate strength method to analyze coupled shear walls. The collapse mechanism

was assumed to have plastic hinges at the points of contact of the connecting beams and the shear wall and at the bottom of the latter.

Paulay(13) emphasized that cracking might have considerable effect on the distribution of internal actions. The elasto-plastic behavior was assessed by an approximate analysis, which also yielded the plastic rotations of the coupling beams. He emphasized ductility in the shear wall component. On shear walls, which carry a low intensity of gravity loading, the lateral load may be large enough to cause cracking. This is certainly the case when the structure is exposed to a moderate earthquake. In building frames, cracking under service conditions can be expected to cause a considerable reduction of stiffness. However, relatively little change in relative stiffness occurs. Hence, cracking need not significantly affect the bending moment pattern for a frame. In shear walls on the other hand the structural components react in markedly different ways to cracking. By means of a step by step procedure the postelastic performance of structure can be traced.

Coull and Choo(14) presented an approximate technique, based on the continuum representation of the system of connecting beams, for the analysis of the elasto-plastic behavior of laterally loaded coupled shear walls. The assumption is made that plastic hinges form at the ends of a connecting beam, and the substituted continuous medium, when the end moments reach a limiting value.

Gluck(15) presented an elasto-plastic procedure of lateral load analysis of coupled shear walls based on the continuum approach. Due to limitations on the rotational ductility factor, full plastification with height of the lamina system is very rarely possible. The ultimate state of the structure being associated with an incomplete collapse mechanism, a plastification on part of the height was to be considered.

2.2 FRAME METHOD

2.2.1 ELASTIC ANALYSIS

MacLeod(16) showed how three-dimensional behavior of a shear wall structure could be considered in analysis without using a full space frame approach. A plane frame analysis involving line elements with rigid ends was used to idealize the various wall conditions.

MacLeod(17) developed the complete stiffness matrix, showing how the shear, bending and axial deformation of the flexible part, rotational springs and the rigid parts are taken into account. He also described a method for carrying out the back substitution for member actions.

Heidebrecht and Swift(18) presented a method for analysing asymmetrical coupled shear walls which takes into consideration the warping of the shear wall piers and the coupling of the piers by floor beams and slabs.

MacLeod and Hosny(19) presented frame analysis of shear wall cores. He dealt with any type of rectangular layout, e.g., unsymmetrical cores, multicell cores, variation of

properties with height, etc.. He took into consideration, torsional, lateral, and vertical load. Supports can be fixed or elastically restrained.

2.2.2 PLASTIC ANALYSIS

Nayar and Coull(20) presented a linear stepwise elastic-plastic analysis based on the wide-column frame analogy. Incremental loads are applied until a plastic hinge forms at the most heavily stressed section. The assemblage of the elastoplastic stiffness matrix is organized.

2.3 FINITE ELEMENT METHOD

MacLeod(21) , Girijavallebhan(22) , Al-Mahaidi and Nilson(23) presented the finite element method in determining stress distributions in shear walls with openings and in predicting more accurate values for the bending moment, and axial and shear forces which act upon the beam when the shear walls are subjected to boundary loads. MacLeod used the polynomials of second order as displacement functions which were chosen by a trial and error process on the basis of the boundary compatibility criterion. Researchers mentioned above experimented only the homogeneous materials and didn't analyse the shear walls with reinforcement.

2.4 DYNAMIC ANALYSIS AND SIMULATED SEISMIC ANALYSIS

Paulay(24) presented the behavior of coupling beams of reinforced concrete shear walls. For a coupled shear wall structure to develop its ultimate strength with modest overall ductility, the coupling beams need to possess a

considerable ability to deform plastically. To ensure maximum ductility, diagonal tension failure in the beams must be avoided. When diagonal tension failure in the beams was prevented by adequate stirrup reinforcement, the ultimate flexural strength of the coupling beams could be predicted. To ensure the utilization of the maximum available ductility, the shear strength must not be permitted to govern the design. The loss of stiffness, after the cracking of the test beams, was considerably more than the loss encountered with normally proportioned reinforced concrete flexural members.

Paulay(25) presented the analysis of spandrel beam under the simulated seismic loading. In deep spandrel beam (span to depth ratio is less than 2), after cyclic loading of high intensity, enough to cause the onset of yield, only 85 % of the flexural strength, as predicted by a conventional ultimate load analysis, can be expected to develop. To ensure the maximum ductility obtainable, a shear failure, by separation along main diagonal, must be suppressed.

Paulay(26) presented the ductile behavior of coupled shear walls. In coupling beams, it was evident that the energy absorbed by the diagonal reinforcement was considerably larger particularly at low loads. He showed that the extra energy absorbed at the instrumented lengths of the diagonal reinforcement was approximately 50 % of the corresponding total energy absorbed in usually reinforced beams. The use of ductile coupling beams enabled a considerable portion of the total energy to be dissipated in

the coupling system. This dissipation was dispersed over the full height of the structure and it resulted in a relief of the plastic hinges at the base of the walls.

Paulay, Priestly, and Syngge(27) analysed the ductility in earthquake resisting squat shear walls. They experimented squat shear walls with a height to length ratio of less than 2. They examined the possibilities of achieving an acceptable level of energy dissipation in squat shear walls, mainly by flexural yielding. They showed sliding shear along the base is the most significant cause of loss of stiffness and strength and consequent reduction in ability to dissipate energy. Crossed diagonal reinforcement in two directions, effective in both tension and compression, was found to considerably improve seismic response of squat shear walls. They presented how to determine a desirable amount of diagonal shear reinforcement to control sliding and necessary horizontal shear reinforcement to prevent diagonal tension failure.

CHAPTER 3

METHOD OF ANALYSIS

The use of finite element method as a tool of stress analysis is a computer based technique for establishing and solving the governing equations for complex problem in a very effective way. In this method, the actual continuum or structure is represented as an assemblage of finite elements. These elements are interconnected at nodes. Since the functions of displacements, strains and stresses are unknowns, the displacement components within an element are approximately characterized by the unknown parameters of the corresponding displacement components and/or their partial derivatives at the selected nodal points of the element by the use of interpolation functions, the shape functions(28, 29,30,31). Using these expressions, one can calculate the strain components and use the obtained results to find the desired stresses and the corresponding strain energy by the method of numerical integrations.

By the use of the principle of virtual work, or the stationary conditions of the total energy in each element of the structural system, the relationships between the generalized nodal displacement components and the generalized force components are obtained. When the elements are assembled together to form the entire structure of interest, a matrix equation for finding the nodal displacement parameters is obtained. Also appropriate boundary conditions should be used in the establishment of the matrix equation to

ensure the existence of the unique solution of the problem. A list of the standard steps of the finite element analysis is :

- i) Discretization of the structure
- ii) Selection of proper interpolation functions
- iii) Calculation of element stiffness matrices and load vectors by the use of numerical integration
- iv) Assemblage of element stiffness matrices to obtain overall global stiffness matrix and loading vector
- v) Enforcement of appropriate boundary condition
- vi) Solution for the unknown nodal displacements
- vii) Computation of strains and stresses at the points of interest

Because of the limitation of the memory space of Zenith 100 micro-computer, the matrix equation was further reduced by the use of antisymmetric property of the displacement of the particular problem of interest. The detailed steps of these operations will be reported later.

SHAPE FUNCTIONS AND DEGREES OF FREEDOM

The shear wall of interest was treated as a plane stress problem since the stress component in the direction of thickness is equal to zero almost everywhere. (30,31). There are many shape functions for solving 2-dimensional elasticity problems available such as constant strain element, element with quadratic interpolations, Hermitian interpolation and n-th order Hermitian interpolating functions, etc.. Through the numerical experiments during the investigation of the

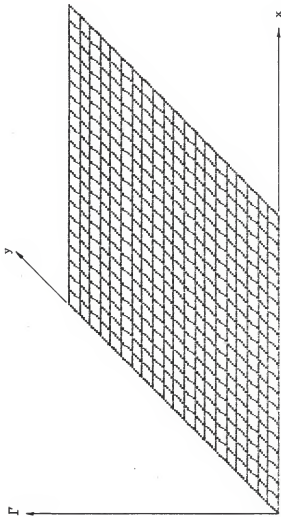
properties of the Hermitian interpolating function, the author discovered that the strain energy density function obtained by the use of Hermitian interpolation functions to represent the displacement function of elements under the application of rigid body displacement is zero only when the domain of element is a rectangle (See Fig.1-Fig.1b and program "ENERGY" in APPENDIX. Note that this plot shows that strain energy value is zero everywhere when the given nodal displacements are specified according to a rigid body displacement). Otherwise, for example an arbitrary quadrilateral element, the obtained strain energy density functions for rigid body displacements are not zero.

In order to reduce the number of unknowns, it is not suitable to use constant strain element. The use of the first order Hermitian element can provide quadratic stress expressions over each element and the obtained stress expression over the entire continuum will be continuous. It is natural to use rectangular shaped domain for the analysis of shear wall-typed structures which will produce consistent strain distributions according to theoretical expressions as well as numerical experiment performed by the use of the Zenith 100.(See Fig. 2)

The Hermitian shape function for this particular case can be constructed according to the products of the following functions:

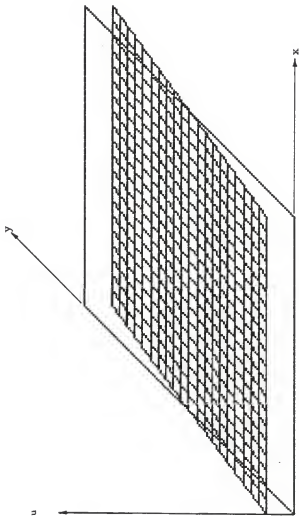
$$F_1(X) = \frac{L-X}{L} + \frac{X(L-X)}{L^3} (L-2X) \quad (1)$$

Fig.1 Strain energy of a rectangular element due to nodal displacements



STRAIN ENERGY DUE TO $U_1, 2 = .1$ $U_3, 4 = -.1$, $V_1, 4 = -.1$ $V_2, 3 = .1$, and $\Theta = .00125$

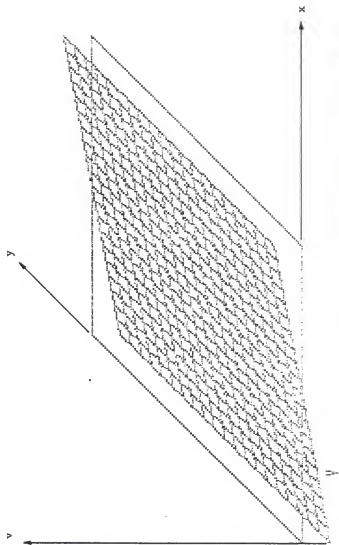
Fig. 1-a X-direction movement due to nodal displacements



U

DUE TO U1,2=.1 U3,4=-.1,V1,4=-.1 V2,3=.1 & THETA=.00125

Fig. 1-b Y-direction movement due to nodal displacements



DOE 10 01, 2=, 1 U3, 4=-, 1, V1, 4= . 1 V2, 3=, 1 A 111.111=, 60125

$$F_2(X) = \frac{X}{L} + \frac{X(L-X)}{L^3}(2X-L) \quad (2)$$

$$F_3(X) = \frac{X(L-X)^2}{L^2} \quad (3)$$

$$F_4(X) = -\frac{(L-X)X^2}{L^2} \quad (4)$$

Note that $F_1(X)$ and $F_2(X)$ are exactly the influence lines of R_a and R_b under the application of a moving unit load on a fixed-fixed beam. The $F_3(X)$ and $F_4(X)$ are the influence lines of the fixed end moments $M_a(X)$ and $M_b(X)$ as shown in Fig. 2.

Let the nodal displacement components at the i -th node be arranged according to the following vectors,

$$(U_i) = \begin{Bmatrix} u_i \\ v_i \\ (\partial u / \partial x)_i \\ (\partial v / \partial x)_i \\ (\partial u / \partial y)_i \\ (\partial v / \partial y)_i \end{Bmatrix} = \begin{Bmatrix} u_i \\ v_i \\ (u_x)_i \\ (v_x)_i \\ (u_y)_i \\ (v_y)_i \end{Bmatrix} \quad (5)$$

and, locally (with respect to each element) the order of the 4 nodes of the e -th element be ordered according to Fig. 2, the components of the nodal displacement parameters of this element expressed in a row vector form reads

$$(u^e) = \underline{u_1, v_1, u_{x1}, v_{x1}, u_{y1}, v_{y1}, u_2, v_2, u_{x2}, v_{x2}, u_{y2}, v_{y2}, u_3, v_3, u_{x3}, v_{x3}, u_{y3}, v_{y3}, u_4, v_4, u_{x4}, v_{x4}, u_{y4}, v_{y4}} \quad (6)$$

According to this arrangement, the 24 interpolating functions corresponding to these nodal parameters are constructed as

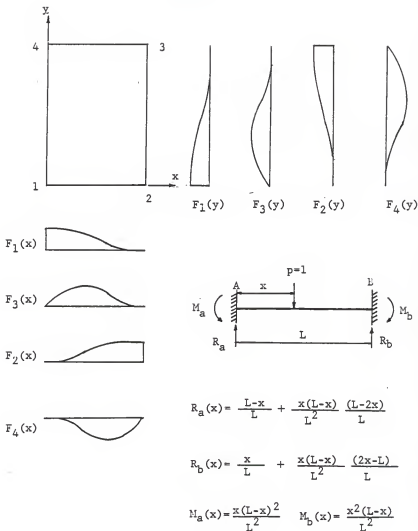


Fig. 2 The local ordering system and the shape function

follows.

$$\begin{array}{ll}
 N_1 = F_1(X) \cdot F_1(Y) & N_3 = F_3(X) \cdot F_1(Y) \\
 N_2 = N_1 & N_4 = N_3 \\
 N_5 = F_1(X) \cdot F_3(Y) & N_7 = F_2(X) \cdot F_1(Y) \\
 N_6 = N_5 & N_8 = N_7 \\
 N_9 = F_4(X) \cdot F_1(X) & N_{11} = F_2(X) \cdot F_3(Y) \\
 N_{10} = N_9 & N_{12} = N_{11} \\
 N_{13} = F_2(X) \cdot F_2(Y) & N_{15} = F_4(X) \cdot F_2(Y) \quad (7) \\
 N_{14} = N_{13} & N_{16} = N_{15} \\
 N_{17} = F_2(X) \cdot F_4(Y) & N_{19} = F_1(X) \cdot F_2(Y) \\
 N_{18} = N_{17} & N_{20} = N_{19} \\
 N_{21} = F_3(X) \cdot F_2(Y) & N_{23} = F_1(X) \cdot F_4(Y) \\
 N_{22} = N_{21} & N_{24} = N_{23}
 \end{array}$$

These interpolating functions are used in the 3-dimensional plot program (see appendix) and plotted in Fig. 3 - 14 for a typical rectangular shaped element.

The equation which describes the displacement of the element is

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & N_3 & 0 & N_5 & \dots & N_{21} & 0 & N_{23} & 0 \\ 0 & N_2 & 0 & N_4 & 0 & \dots & 0 & N_{22} & 0 & N_{24} \end{bmatrix} (u^e) \quad (8)$$

For simplicity, the displacement components of the e-th element can be rewritten as

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = [N] (u^e) \quad (9)$$

The definition of [N] and (u^e) can be redefined by comparing Eq.(5) and Eq.(7).

STRAIN COMPONENTS

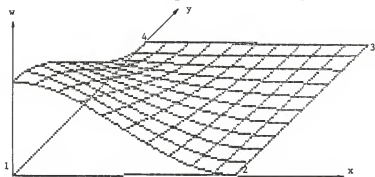


Fig.3 First order Hermitian shape function using nodal w of node 1

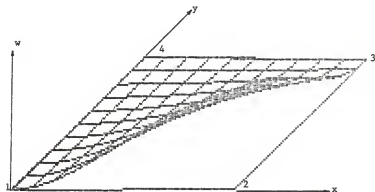


Fig. 4 First order Hermitian shape function using nodal w of node 2

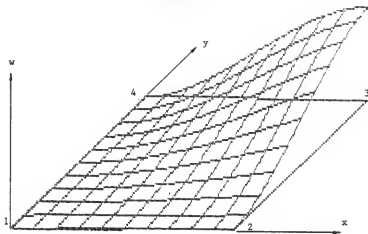


Fig. 5 First order Hermitian shape function using nodal w of node 3

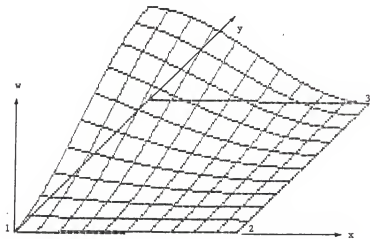


Fig. 6 First order Hermitian shape function using nodal w of node 4

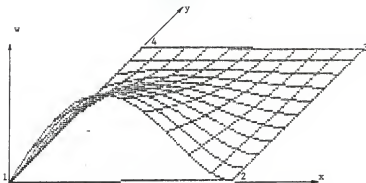


Fig. 7 First order Hermitian shape function using dw/dx of node 1

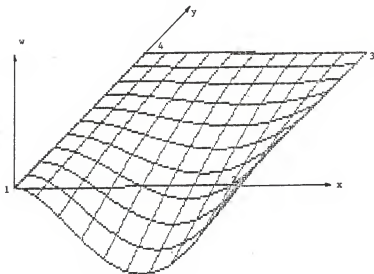


Fig. 8 First order Hermitian shape function using dw/dx of node 2

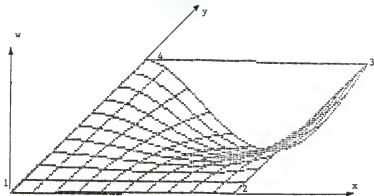


Fig. 9 First order Hermitian shape function using dw/dx of node 3

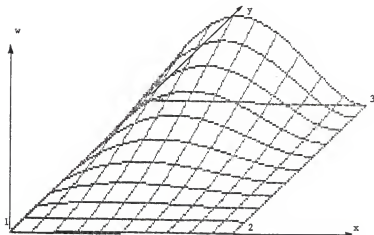


Fig. 10 First order Hermitian shape function using dw/dx of node 4

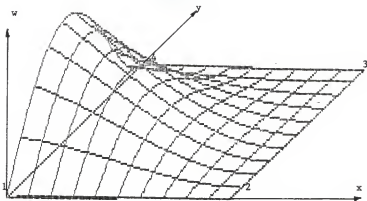


Fig. 11 First order Hermitian shape function using dw/dy of node 1

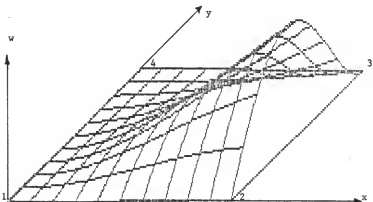


Fig. 12 First order Hermitian shape function using dw/dy of node 2

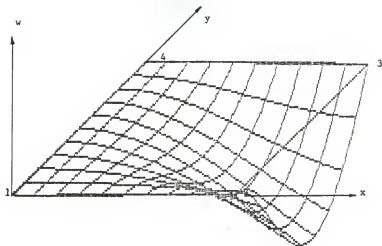


Fig. 13 First order Hermitian shape function using dw/dy of node 3

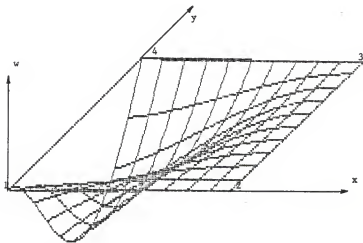


Fig. 14 First order Hermitian shape function using dw/dy of node 4

For the stress analysis of shear wall, it is considered as a plane stress problem. The strain components are calculated by

$$\{ \epsilon \} = \{ B \} \{ u^e \} \quad (10)$$

where

$$\{ \epsilon \} = \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} \quad (11); \quad [B] = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \quad \left[N \right] \quad (12)$$

The detailed expression of [B] reads

$$[B] = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 & \dots & \frac{\partial N_{21}}{\partial x} & 0 & \frac{\partial N_{23}}{\partial x} & 0 \\ 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_4}{\partial y} & \dots & 0 & \frac{\partial N_{22}}{\partial y} & 0 & \frac{\partial N_{24}}{\partial y} \\ \frac{\partial N_2}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_4}{\partial y} & \frac{\partial N_3}{\partial x} & \dots & \frac{\partial N_{22}}{\partial y} & \frac{\partial N_{21}}{\partial x} & \frac{\partial N_{24}}{\partial y} & \frac{\partial N_{23}}{\partial x} \end{bmatrix} \quad (13)$$

In case of linear elastic isotropic plane stress problem, the stress-strain relations are given by Hooke's law .

$$\{ \sigma \} = [D] \{ \epsilon \} \quad (14)$$

where

$$[D] = \frac{E}{1 - \mu^2} \begin{bmatrix} 1 & \mu & 0 \\ \mu & 1 & 0 \\ 0 & 0 & \frac{1 - \mu}{2} \end{bmatrix} \quad (15)$$

and

$$\{ \sigma \} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} \quad (16)$$

The strain energy of the element can be expressed as follows.

$$U = \iint \Gamma(x, y) t \, dx \, dy \quad (17)$$

where the Γ is the strain energy density function

$$\Gamma = \frac{1}{2} \{ \epsilon \}^T [D] \{ \epsilon \} \quad (18)$$

and t is the function of thickness of a wall.

By the substitution of Eq. (10), one has the strain energy

$$U = \frac{1}{2} \{ u^e \}^T \left[\iint t [B]^T [D] [B] \, dA \right] \{ u^e \} \quad (19)$$

Note that by the use of the definition of stiffness matrix, the nodal force $\{ F^e \} = [K] \{ u^e \}$. Therefore the work done by the nodal force

$$W = \frac{1}{2} \{ u^e \}^T [K] \{ u^e \} \quad (20)$$

The expression of stiffness can be yielded by comparison of U and W .

$$[K] = \iint [B]^T [D] [B] \, dA \quad (21)$$

CALCULATION OF GENERALIZED NODAL FORCES

In terms of nodal forces and nodal displacement components, the work done by the e -th element is

$$W = \frac{1}{2} \{ u^e \}^T \{ F^e \} \quad (22)$$

The work done due to the external loading intensity of $f_x(s)$ and $f_y(s)$, which are the x and y components, respectively, at s over the i -th loaded side s_i can be calculated by

$$W_i = \frac{1}{2} \int_{s_i} \{ u, v \} \begin{Bmatrix} f_x(s) \\ f_y(s) \end{Bmatrix} ds \quad (23)$$

The work done by the external load applied on the N^e boundaries of the e -th element is the summation of W_i .

$$W^e = \sum_{i=1}^{N^e} \frac{1}{2} \int_{s_i} \{ u, v \} (f) ds \quad (24)$$

By the substitution of Eq.(8) into Eq.(24), one gets

$$W^e = \sum_{i=1}^{N^e} \frac{1}{2} \int_{s_i} \{ N^e \}^T (f) ds \quad (25)$$

By the use of the principle of virtual work, one can show that the i -th component of the (F^e) vectors is

$$F_i^e = \frac{\partial W^e}{\partial u_i} \quad (26)$$

Thus

$$(F_i)^e = \sum_{i=1}^{N^e} \int_{s_i} \{ N \}^T (f_s) ds_i \quad (27)$$

The vector of generalized nodal force components is obtained.

NUMERICAL INTEGRATION AND THE USE OF GAUSSIAN QUADRATURE

There are several methods available for the numerical evaluation of definite integrals. The use of Gaussian quadrature has been proved to be most useful for finite element work because the integration of a n -point Gaussian quadrature can provide exact integration for $(2n-1)$ -th polynomials. In two-dimensional (rectangular) regions, the result can be obtained by using Gaussian quadrature with respect to x and y direction iteratively.

$$I = \sum_{i=1}^n \sum_{j=1}^n W_i W_j f(x_i, y_j) a \cdot b \quad (28)$$

where W_i and W_j are the weighting factors at the i -th and j -th sample point and x_i and y_j are the x and y values of the i -th and j -th sample points. (30) The meaning of a and b are shown in Fig. 15.

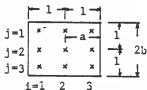


Fig. 15 Gaussian points

In the computer program developed, 9 Gaussian points over the rectangular domain were used. They are shown in Fig. 15.

STIFFNESS OF THE REINFORCED STEEL BAR

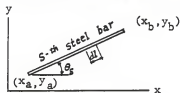


Fig.16 A steel bar of an element

The contribution of the reinforced steel shown in Fig.16 to the stiffness matrix of an element is considered as follows. Since the reinforced steel bar is embedded in the

concrete, it is suitable to assume that the displacement of the reinforced steel bar at any point is characterized by the same displacement functions of the element which contains the steel bar of interest, i.e.,

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} N_u \\ N_v \end{bmatrix} \{ u^e \} \quad (29)$$

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{Bmatrix} = [B] \{ u^e \} \quad (30)$$

The strain of interest for a steel bar is only the strain in the axis of the bar. Therefore

$$\begin{aligned} (\epsilon_{\theta})_s &= \begin{bmatrix} \cos^2\theta & \sin^2\theta & \frac{1}{2}\sin 2\theta \end{bmatrix} \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} \\ &= \begin{bmatrix} T_s \end{bmatrix} (\epsilon) \end{aligned} \quad (31)$$

i.e.,

$$(\epsilon_{\theta})_s = \begin{bmatrix} T_s \end{bmatrix} [B] (u^e) \quad (32)$$

To avoid addition of concrete again, which is replaced by steel, the Young's modulus of steel E_s is replaced by

$$E^* = E_s - E_c \quad (33)$$

where E_c is the Young's modulus of concrete. Therefore

$$\epsilon_s = (E_s - E_c) (\epsilon_{\theta})_s \quad (34)$$

The additional strain energy of the reinforced steel bar of area A_s will be

$$U_s = \frac{A_s}{2} \int_a^b \begin{bmatrix} u^e \end{bmatrix} [B]^T (T_s) E^* \begin{bmatrix} T_s \end{bmatrix} [B] (u^e) dl \quad (35)$$

Thus the additional stiffness due to the s-th steel bar will be

$$K_s = A_s \int_{l_s} [B]^T (T_s) E^* \begin{bmatrix} T_s \end{bmatrix} [B] dl \quad (36)$$

By adding the summation of the contribution of all bars to the stiffness of concrete of the e-th structural element, the combined stiffness matrix becomes

$$[K]^e = [K_{\text{concrete}}]^e + [K_{st}]^e \quad (37)$$

where $[K_{st}]$ denotes stiffness of steel bars and the expression is

$$[K_{st}]^e = \sum_{s=1}^{n^e} [K_s] = \sum_{s=1}^{n^e} A_s \int_{l_s} [B]^T (T_s) E^* \begin{bmatrix} T_s \end{bmatrix} [B] dl \quad (38)$$

CHAPTER 4

NUMERICAL EXPERIMENT

The dimension of model shear walls analysed are the same as that employed in a test unit experimented by Paulay (27). The overall dimension of the model shear walls is shown in Fig. 17.

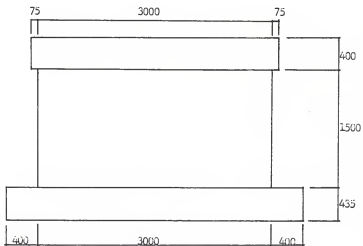
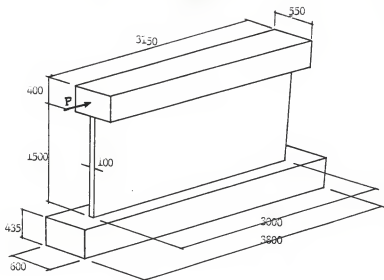
The model shear wall is divided into 52 rectangular elements with 71 nodal points as shown Fig. 18. The 3 types of shear walls with different array of reinforcement as shown in Fig. 19 and 20 are analysed. Properties of each shear walls are presented in Table. 1.

WALL TYPE	SECTION	f'_c MPa	REINFORCEMENT		
			HORIZONTAL	VERTICAL	DIAGONAL
TYPE 1	rectangular	27.2	D14-120	D12-140	-
TYPE 2	rectangular	27.2	D12-120	D12-140	-
TYPE 3	rectangular	27.2	-	4-D12 for each edge	D12-200

The reinforcement used in type 1 is exactly the same as that used by Paulay (27)

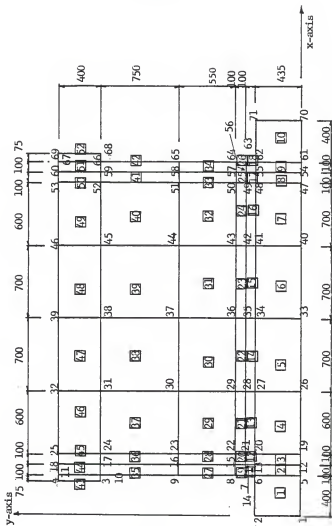
MODULUS OF ELASTICITY

For the purpose of analysis of the shearwalls, three different kinds of modulus of elasticity of concrete were used as shown in Stress-strain curve of Fig. 21. The first value used is the modulus of elasticity which is calculated according to the formula (8-5-1) given in the ACI code (34). For clarity, first value will be called as initial modulus



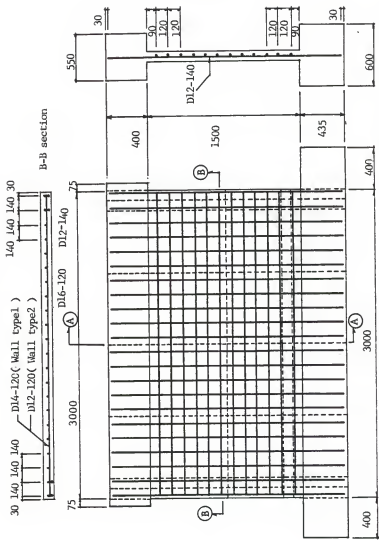
(unit:mm)

Fig. 17 Over all dimensions of model structure and elevation



(unit:mm)

Fig. 18 Finite element representation



A-A section

(unit:mm)

Fig. 19 Shear wall type1 and type2

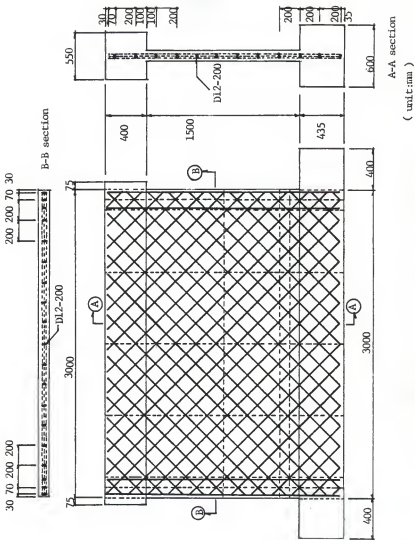
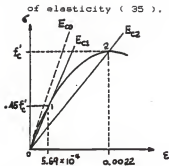


Fig. 20 Shear wall type3 with diagonal reinforcement



$$E_{c0} = 57000 \sqrt{f'_c}$$

$$= 3579700 \text{ psi (24700 MPa)}$$

The maximum strain ϵ_2 , at point 2 shown in Fig.24, of this concrete is estimated as follows.

$$\epsilon_2 = \frac{2f'_c}{E} = 2.203 \times 10^{-3}$$

Fig.21 Stress-strain curve of concrete The stress-strain curve can be interpolated by the use of parabolic curve.

$$\sigma(\epsilon) = A_1 \cdot \epsilon + A_2 \cdot \epsilon^2 \quad (39)$$

or

$$A_2 = \frac{\sigma(\epsilon) - A_1 \cdot \epsilon}{\epsilon^2} \quad (40)$$

Differentiation of Eq. (39) with respect to ϵ yields

$$\frac{d\sigma(\epsilon)}{d\epsilon} = A_1 + 2 A_2 \epsilon \quad (41)$$

Initial Modulus of elasticity can be found by substituting $\epsilon=0$ into Eq. (41).

$$A_1 + 2A_2 \cdot 0 = 3579700$$

Therefore,

$$A_1 = 3579700 \text{ psi}$$

The coefficient A_2 can be obtained by substituting the value of ϵ_2 and A_1 into Eq. (40).

$$A_2 = \frac{3944 - 3579700 \cdot 0.0022}{(0.0022)^2}$$

$$= - 812260000$$

Therefore the function of stress-strain curve is

$$\sigma(\epsilon) = 3579700\epsilon - 812260000\epsilon^2 \quad (42)$$

When the stress $\sigma(\epsilon) = 0.45 f'_c$, the strain can be obtained by substituting $0.45 f'_c = 1774.8$ psi into Eq. (42).

Thus,

$$1774.8 = 3579700\epsilon_1 - 812260000\epsilon_1^2$$

ϵ can be obtained by solving the above equation. Therefore,

$$\epsilon_1 = 5.6935 \times 10^{-4}$$

Secant modulus of elasticity E_{c1} can be obtained by substituting the above ϵ_1 into the following equation.

$$\begin{aligned} E_{c1} &= \frac{0.45 f'_c}{\epsilon_1} \\ &= \frac{1774.8}{5.6935 \times 10^{-4}} = 3117000 \text{ psi (21500 MPa)} \end{aligned}$$

Another secant modulus of elasticity corresponding to $f_c = f'_c$ can be obtained by the following.

$$\begin{aligned} E_{c2} &= \frac{f'_c}{\epsilon_{\max}} \\ &= \frac{3944}{0.0022} = 1793000 \text{ psi (12400 MPa)} \end{aligned}$$

Modulus of elasticity of reinforcement used was 29×10^6 psi as described in (8-5-2) of ACI code.

RESULTS OF NUMERICAL EXPERIMENT

Three model structures with different reinforcement were studied using three different moduli of elasticity of concrete for each model. Lateral load of 200 kN was applied on the upper left top of the model as shown in Fig. 17. The

flow chart of computer program as shown in Fig. 33 was followed and the programs explained in Chapter 5 were used.

Nodal displacements of each node were found and two dimensional plotting of displacement shape as shown in Fig. 23 was drawn by the computer program.

By the use of the known displacements, the trajectories of stresses as shown in Fig. 22 can be found. The steel reinforcements inside the shear walls can be placed by following the trajectories of the stresses. For simplicity, the diagonal reinforcement was used in wall type 3.

By the use of the known displacements of each node, strain and stress of the points of interest were calculated. The maximum strains of model structure were found and presented in Table. 2. Also the three dimensional displays of normal stresses, shearing stresses and stresses of reinforcement are shown in Fig. 24 - 32. For simplicity, the results of shear wall type 1 (Fig. 24-30) and shear wall type 3 (Fig. 31-32) using the initial modulus E_{c0} are presented.

The comparison of allowable shear force between the estimated value and the experiment data done by Paulay (27) are presented in Table. 3. The estimated allowable shear force is 2.4 times higher than that of the experimental data. The estimated cracking load is smaller than that of the experiment data.

A comparison of the ultimate shear force between the estimated and the experiment data is presented in Table. 4.

The estimated ultimate shear force is 2 times higher than that of the experiment data.

If the initial modulus of elasticity at zero stress level is used in the finite element analysis, the estimated value of shear force that could start crack of concrete is calculated as follows.

$$\begin{aligned}
 P_{\text{crack}} &= \frac{(e_c'/10) \cdot P_{\text{applied}}}{\text{strain}} \\
 &= \frac{(0.0022/10) \cdot 200}{1.130213 \cdot 10^{-4}} = 389 \text{ kN}
 \end{aligned}$$

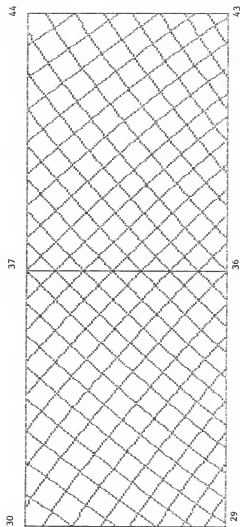
When the highest compressive stress of shear wall reaches $0.45f_c$, the stress of rest parts of shear wall is less than $0.45f_c$. Therefore, Young's modulus of whole structure is between the value calculated according to initial modulus at zero stress level and that calculated at the secant modulus of $0.45f_c$. This suggests the average value of allowable P predicted by these two models should be more accurate in comparison with the experimental data. According to this, the average allowable P is equal to

$$\frac{389 + 342^{**}}{2} = 365.5 \text{ kN}$$

The error is less than 1 % difference between the value of $0.45 P_{\text{exPerimented}}$ and this estimation.

** This value $P=342 \text{ kN}$ is reported on PP.52 which is an estimated P value that could start the hair crack of concrete in the wall.

Fig. 22 Trajectories of the stresses (element 30 and 31)



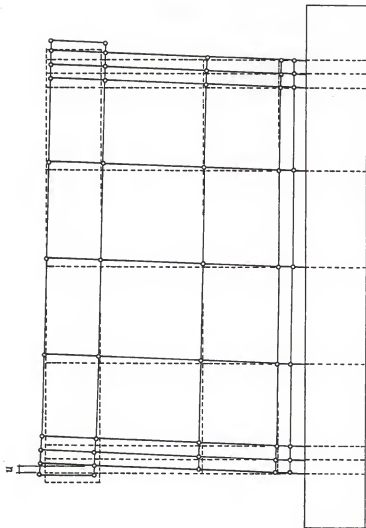
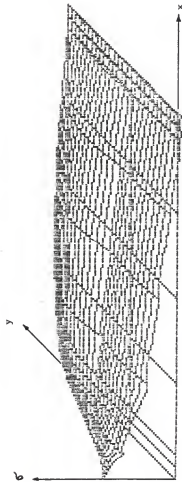


Fig. 23 Displacement of shear wall

Fig. 24 Maxium principal stress - shear wall type 1



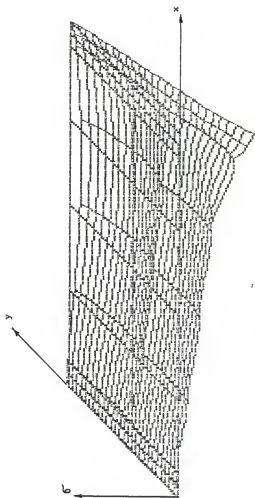
MAXIMUM PRINCIPAL STRESS

MAX 3.019127 x coor. 40 y coor. 63.5 MIN-.1100787 x coor. 325 y coor.

63.5

unit: MPa

Fig. 25 Minimum principal stress - shear wall type 1

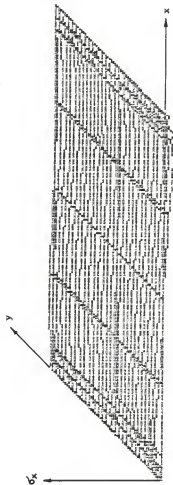


MINIMUM PRINCIPAL STRESS

MAX .1100787 x coor. 55 y coor. 63.5 MIN-3.019127 x coor. 340 y coor.
63.5

unit: MPa

Fig. 26 Normal stress in x-direction - shear wall type 1

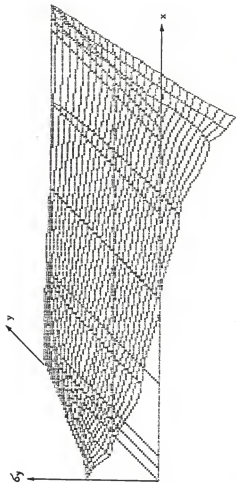


NORMAL STRESS IN X-DIRECTION

MAX .3315765 x coor. 55 y coor. 63.5 MIN-.3315767 x coor. 325 y coor.
63.5

unit: MPa

Fig. 27 Normal stress in y-direction - shear wall type 1

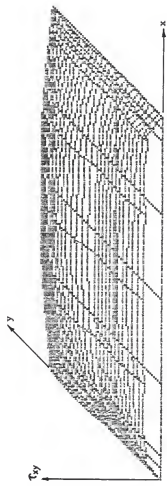


NORMAL STRESS IN Y-DIRECTION

MAX 3.019127 x COOR. 40 y COOR. 63.5 MIN-3.019127 x COOR. 346 y COOR.
63.5

unit: MPa

Fig. 28 Shearing stress with respect to x,y axes-shear wall type 1



SHEARING STRESS WITH RESPECT TO X,Y AXES
MAX 1.216703 x coor. 190 y coor. 148.5 MIN-2.930604E-02 x coor. 40
y coor. 103.5
unit: MPa

Fig. 29 Stress of horizontal steel bar-shear wall type 1

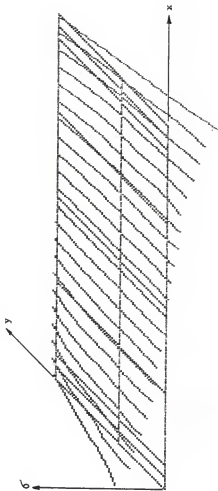


unit: MPa

STRESS OF STEEL BAR(horizontal or greater than 90 deg)

MAX 3.599114 x coor. 337 y coor. 64.5 MIN-3.599113 x coor. 43 y coor.
64.5

Fig. 30 Stress of vertical steel bar - shear wall type 1

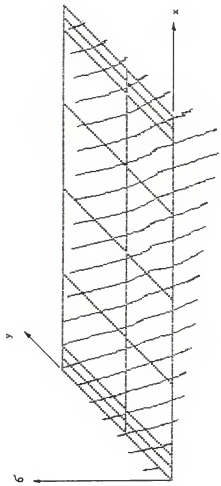


unit: MPa

STRESS OF STEEL BAR(vertical or less than 90 deg)

MAX 20.78271 x coor. 43 y coor. 63.5 MIN-20.78271 x coor. 337 y coor.
63.5

Fig. 31 Stress of diagonal bar - shear wall type 3

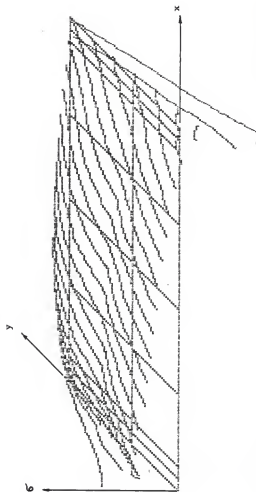


unit: MPa

STRESS OF STEEL BAR(horizontal or greater than 90 deg)

MAX 6.433375 x coor, 43 y coor. 70.5 MIN-12.28311 x coor, 327.5 y coor.

Fig. 32 Stress of diagonal bar - shear wall type 3



unit: MPa

STRESS OF STEEL BAR (vertical or less than 90 deg)

MAX 20.93987 x coor. 43 y coor. 63.5 MIN-20.93987 x coor. 337 y coor.

63.5

0:

Table. 2 Maximum strains of the model structures

1	2	3	4	5	6	7	8
WALL TYPE	E	ABSOLUTE MAXIMUM PRINCIPAL STRAIN *10 ⁻⁴	MAXIMUM NORMAL STRAIN IN X DIR. *10 ⁻⁵	MAXIMUM NORMAL STRAIN IN Y DIR. *10 ⁻⁴	MAXIMUM SHEARING STRAIN W.R.T. X, Y	MAXIMUM STRAIN OF STEEL *10 ⁻⁴	REMARKS
	E _{c0}	1.13	2.14	1.07	1.02	1.05	
TYPE 1	E _{c1}	1.29	2.07	1.22	1.18	1.19	
	E _{c2}	2.14	3.37	1.98	2.03	1.93	
	E _{c0}	1.13	2.14	1.07	1.03	1.05	
TYPE 2	E _{c1}	1.29	2.08	1.22	1.18	1.19	
	E _{c2}	2.14	3.37	1.98	2.03	1.93	
	E _{c0}	1.11	2.10	1.06	.972	1.04	
TYPE 3	E _{c1}	1.26	2.40	1.20	1.10	1.17	
	E _{c2}	2.04	4.08	1.94	1.79	1.89	

By using the data shown in this table, the allowable P and ultimate P can be calculated and compared to the experimental value.

Table. 3 Comparison of the allowable shear force between the estimated value and experimental data (wall type 1)

P allowable using E_{cl} *		Predicted P	0.45(P _{ult}) exp.	$\frac{P \text{ predicted}}{0.45(P_{ult}) \text{ exp.}}$
(f _c) ^{max} = 0.45f _c **	$\left(\frac{f_y}{1.6}\right)$ +			
885kN	1673kN	885kN	364.5kN	2.4

* Sample calculation for each column of the table.

$$1. \text{ Column 1. } \frac{1.29 * 10^{-4}}{5.69 * 10^{-4}} * 200 = 885$$

$$2. \text{ Column 2. } \frac{46}{1.6} / 29 * 10^3 * 1.185 * 10^{-4} = 1673$$

$$3. \text{ Column 4. } 0.45 * 810 = 364.5$$

** Maximum allowable compressive stress level. + Maximum allowable stress in steel.

If assumed threshold strain of concrete in tension be 0.0022/10, then P value which starts a hair crack of concrete can be estimated as follows.

$$P_{\text{crack}} = \frac{0.0022}{10} * 200 = \frac{1.286 * 10^{-4}}{1.286 * 10^{-4}} = 342\text{kN}$$

Table. 4 Comparison of the ultimate shear force between the estimated value and experimental data. (wall type 1)

P ultimate using E_c2		$(P_{ult})_{exp.}$	$\frac{P_{predicted}}{(P_{ult})_{experimented}}$
$(f'_c)^{**}$	$(f_y)^+$ Predicted P		
2060kN	1650kN	810kN	2.0

* Sample calculation for each column of this table.

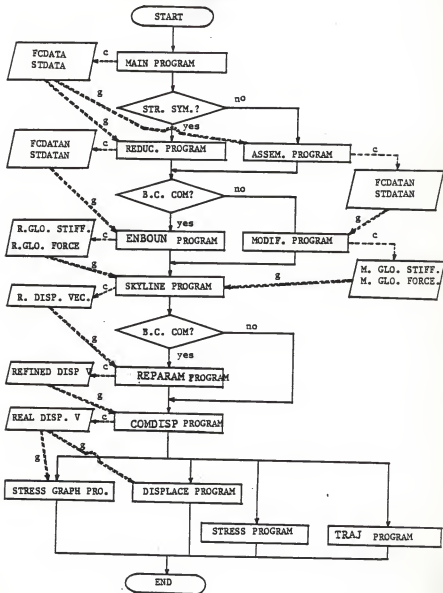
$$1. \text{ Column 1. } \frac{2.13 * 10^{-4}}{0.0022} * 200 = 2060$$

$$2. \text{ Column 2. } \frac{46}{29 * 10^3 * 1.93 * 10^{-4}} = 1650$$

** Maximum stress level in compressive concrete of steel reinforced shear wall.

+ Maximum stress level in tension steel bar.

Fig.33 PROGRAM FLOW CHART



CHAPTER 5

THE COMPUTER PROGRAMS DEVELOPED

ELEMENT STIFFNESS MATRIX :MAIN PROGRAM

This program calculates the stiffness matrices and the load vectors of various element. Input data of this program include modulus of elasticity and Poisson's ratio of concrete, coordinates and node numbers, locations of steel bars, steel area, Young's modulus of steel, Gaussian weights and locations, and external loads. Matrix [D] is calculated. Element stiffness matrix and load vector are calculated and stored in STDATA data file and FCDATA file respectively.

ASSEMBLING ELEMENT STIFFNESS MATRIX:ASSEM PROGRAM

This program gets the (24*24) stiffness matrix of each element from a data file named "STDATA" and assembles element by element to form a lower triangular global stiffness matrix of the structure. The data assembled by this program are stored row by row in a data file "STDATAN". This is done by the use of the global order of the 4 nodes of each element assigned to the nodes of a fixed local ordering system shown in Fig. 34.

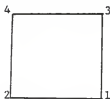


Fig. 34 Local ordering system for a typical element

$$S = \begin{bmatrix} S_{11} & S_{12} & S_{13} & S_{14} \\ S_{21} & S_{22} & S_{23} & S_{24} \\ S_{31} & S_{32} & S_{33} & S_{34} \\ S_{41} & S_{42} & S_{43} & S_{44} \end{bmatrix}$$

Fig. 35 Partitioned matrix

For example, let G_1 , G_2 , G_3 and G_4 be the order number of the nodes $l_1=1$, $l_2=2$, $l_3=3$ and $l_4=4$ as shown in Fig. 35, then the location of each submatrix of the stiffness matrix of the structural element can be associated to the required location in the global stiffness matrix explained as follows.

In this partitioned stiffness matrix, each submatrix has (6×6) elements. The first subscript refers to the local order number of the node to which the six nodal force components are of interest, and the second subscript refers to the related nodes that its six displacements are to be used as its operand. Note that the row number of the global stiffness matrix is related to the corresponding nodal force component, and the column number is related to the corresponding nodal displacement parameter when they are arranged according to the order of the nodes of the structure. For simplicity of explanation, only the procedure to add S_{13} submatrix into the global stiffness matrix $[K]$ will be explained.

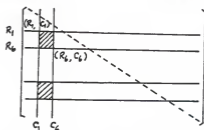


Fig.36 Global stiffness matrix

Since the global node number of l_1 and l_3 are G_1 and G_3 , therefore

$$R_1 = (G_1 - 1) * 6 + 1$$

$$R_6 = G_1 * 6$$

$$C_1 = (G_3 - 1) * 6$$

$$C_3 = G_3 * 6$$

The submatrix S_{13} should be superposed into the global matrix $[K]$ at the location of a submatrix of $K(R_1, C_1)$ and $K(R_6, C_6)$, which are located on the upper left corner and on the lower left corner respectively as shown by the shaded block in Fig. 36. The superposition of all other submatrix can be done similarly. Note that only elements belonging to the lower triangular part of the global stiffness matrix will be stored in the data file.

This program also assembles the element force vector into the global force vector. This program gets the (24×1) force vector of each element from a data file named "FCDATA" and assembles element by element to form a global force vector. The data are stored in a data file "FCDATAN".

By the use of this program, a complete system of simultaneous equation can be expressed as

$$[K] \{u\} = \{F\} \quad (43)$$

REDUCED STIFFNESS MATRIX: REDUC PROGRAM

This program reduces the dimensionality of stiffness matrices to form a reduced matrix equation. Because the structure is a symmetric shear wall, nodal displacements of all the nodes to the right of the structure can be expressed in terms of the corresponding components of the nodal parameter of the corresponding nodes on the left half of the

structure.

Let $[C]$ be a connection matrix which is constructed according to the condition of asymmetry of the displacement field over the corresponding nodes. For example, the conditions of u and v at the points (x_i, y_i) and $(-x_i, y_i)$ are $u(x_i, y_i) = u(-x_i, y_i)$, and $v(x_i, y_i) = -(v(-x_i, y_i))$. If the global order numbers for these two nodes are p and q , the $[(p-1)*6+1]$ th row of C matrix has value of zero at each column except for the value of 1 at column $(q-1)*6+1$. On the $[(p-1)*6+2]$ th row, only the $[(q-1)*6+2]$ th column has value of 1 and the value of rest elements of this row is zero.

After the connection matrix C has been constructed, the complete vector of the global nodal displacements(parameters) $\{u\}$ can be expressed in terms of the reduced vector of unknown nodal displacements $\{u_r\}$ as follows.

$$\begin{matrix} \{u\} \\ (n=1) \end{matrix} = [C] \begin{matrix} \{u_r\} \\ (n=r) \end{matrix} \quad (r=1) \quad (44)$$

Substituting this equation into Eq.(43), one gets

$$\begin{matrix} [K] \\ (n=n) \end{matrix} [C] \begin{matrix} \{u_r\} \\ (n=r) \end{matrix} = \begin{matrix} \{F\} \\ (n=1) \end{matrix} \quad (r=1) \quad (45)$$

Premultiplication of Eq.(45) by the transposition of $[C]$ yields

$$\begin{matrix} [C] \\ (r=n) \end{matrix} [K] [C] \begin{matrix} \{u_r\} \\ (n=r) \end{matrix} = \begin{matrix} [C] \\ (r=n) \end{matrix} \begin{matrix} \{F\} \\ (n=1) \end{matrix} \quad (46)$$

$$\text{Let } [K_r] = [C] [K] [C] \quad (r=r) \quad (47)$$

$$\text{and } \{F_r\} = [C] \{F\} \quad (48)$$

Eq.(46) is reduced into the following form

$$[K_r] \{u_r\} = \{F_r\} \quad (49)$$

However, the $[C]$ and $[C]$ matrices did not actually constructed in the program "REDUC". The concept was used to accomplish the process of $[C] [K] [C]$ and $[C] \{F\}$.

Note that particular attention was given to take care the case that only the lower triangular part of matrix was saved. This introduced complicated manipulation of element location in the development of program "REDUC". The important part of the process are reported as follows.

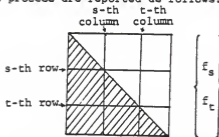


Fig. 37 Lower triangular matrix

Considering the case that the s -th row of $[C]^T$ matrix has the following elements

$$C_s = [0, 0, \dots, 0, 1, 0, \dots, a, 0, \dots, 0] \quad (50)$$

↑ ↑
s-th column t-th column

then the s -th row of $[K^*]$ is the summation of K_s and $a * [K_t]$, i.e.,

$$[K_s^*] = [K_s] + a * [K_t], \quad t > s \quad (51)$$

where

$$[K^*] = [C]^T [K] \quad (52)$$

$[K_s^*]$ = s -th row of K matrix

$[K_t^*]$ = t -th row of K matrix

Note that only the shaded part of K matrix shown in Fig. 37 is saved in the memory, and the required element (K_{ij}) for $j > i$ needs to be calculated by

$$K_{ij} = K_{ji} \quad (53)$$

In the program, K matrix is saved row-wisely from first column to the main diagonal row by row in a column matrix, i.e., (A), then the sequence number of K_{ji} in (A) is

$$n = \frac{j*(j-1)}{2} + 1 \quad (54)$$

Therefore

$$K_{ij} = A(n). \quad (55)$$

By using this method of locating desired elements, one can find K_s as follows:

For elements K_{sj} with $j \leq s$,

$$K_{sj} = A\left(\frac{s*(s-1)}{2} + j\right) + a * A\left(\frac{t*(t-1)}{2} + j\right) \quad (56)$$

For elements K_{sj} with $j > s$ and $j \leq t$,

$$K_{sj} = A\left(\frac{j*(j-1)}{2} + s\right) + a * A\left(\frac{t*(t-1)}{2} + j\right) \quad (57)$$

For K_{sj} with $j > t$,

$$K_{sj} = A\left(\frac{j*(j-1)}{2} + s\right) + a * A\left(\frac{j*(j-1)}{2} + j\right) \quad (58)$$

The superposition of $[K_s]$ and $a*[K_t]$ to complete the s-th column of

$$[C] [K] [C] = [KR] \quad (59)$$

can be done likewise. For simplicity, let

$$[R] = [KR] \quad (60)$$

then the s-th column of R can be expressed

$$[R_s] = [K_s^*] + a * [K_t^*] \quad (61)$$

Note that in the expression of [C] [K] = [K^*] the elements in the related columns have the following calculations.

For $t > i$ and $i \neq s$

$$K_{it}^* = A \left(\frac{t*(t-1)}{2} + i \right) \quad (62)$$

$$K_{st}^* = A \left(\frac{t*(t-1)}{2} + s \right) + a * A \left(\frac{t*(t-1)}{2} + t \right) \quad (63)$$

For $i > t$

$$K_{it}^* = A \left(\frac{i*(i-1)}{2} + s \right) \quad (64)$$

For $i > s$

$$K_{is}^* = A \left(\frac{i*(i-1)}{2} + s \right) \quad (65)$$

$$K_{ss}^* = A \left(\frac{s*(s-1)}{2} + s \right) + a * A \left(\frac{t*(t-1)}{2} + s \right) \quad (66)$$

Substituting these relationships into Eq. (61), one can find the s-th column of the reduced stiffness matrix. The superposition of [C] [F] can be done in a straightforward way which needs no further explanation.

Because the model structure as shown in Fig. 18 has 71 nodes and each node has 6 nodal parameters, the size of assembled global stiffness matrix is (426*426) and the total number of elements of stiffness matrix is 181,476. The Zenith 100 microcomputer cannot handle this amount of data for

lack of memory spaces and the data used should be reduced. By using the advantage of the property of symmetry in the stiffness matrix, lower (or upper) triangular part including main diagonal elements can be stored. In this case, the total elements of triangular part are

$$\begin{aligned} & \frac{6n \cdot (6n+1)}{2} \\ & = \frac{6 \cdot 71 \cdot (6 \cdot 71 + 1)}{2} \\ & = 90,951 \end{aligned}$$

where n denotes total node numbers. Total elements of stiffness matrix by the use of "REDUC" program are

$$\frac{6 \cdot 39 \cdot (6 \cdot 39 + 1)}{2} = 27,495.$$

Percentage of data space required between the "REDUC" program and the usual program not using reduction is

$$\frac{27,495}{90,951} \cdot 100 = 30.23 \%$$

This program requires only 30.23 % memory space. A lot of computer memory spaces and C.P.U. time can be saved. Force vector are also reduced by this program.

BOUNDARY CONDITIONS OF ZERO DISPLACEMENTS:MODIF PROGRAM

After assembling the element stiffness matrix and element load vector of each structural element, the matrix equation of the entire structure can be written as

$$\begin{matrix} [K] & \{ u \} & = & \{ F \} \\ n \times n & n \times 1 & & n \times 1 \end{matrix} \quad (67)$$

Without enforcement of the boundary conditions, this equation

cannot be solved for $\{u\}$ since the matrix $[K]$ will be singular and its inverse does not exist. The physical significance of this is that the structure is free to undergo unlimited rigid body motion unless some support constraints are imposed to keep the structure in equilibrium under the loads. Hence some boundary conditions have to be applied before solving $\{u\}$. The rows and columns corresponding to the zero nodal parameters have to be eliminated in order to make the stiffness matrix non-singular. The lower triangular part of the stiffness matrix is stored as a single dimensional array. Except for the main diagonal elements which are unit, the rest of the eliminating part are made to be zero. Data created in this program is the same as one created in "ASSEM" program except for eliminating rows and columns.

BOUNDARY CONDITIONS OF ZERO STRESSES:ENBOUN PROGRAM

In addition to the procedure of "MODIF" program, the zero stress components of all the boundary nodes are considered in this program.

In case of $\epsilon_x = 0$ on a certain boundary node, then

$$\epsilon_x + \mu \epsilon_y = 0 \quad (68)$$

where μ denotes the Poisson's ratio. Then,

$$\epsilon_x = -\mu \epsilon_y \quad (69)$$

Coefficients of the row and column related to nodal parameter $\partial u/\partial y$ are superimposed by the coefficients, which are multiplied by $-1/\mu$, of the row and column related to nodal parameter $\partial u/\partial x$. The coefficients related to $\partial u/\partial x$ are

eliminated .

In case of $\epsilon_y = 0$ on a certain boundary node, then

$$\epsilon_y + \mu \epsilon_x = 0 \quad (70)$$

or

$$\epsilon_y = -\mu \epsilon_x \quad (71)$$

Superposition of coefficients are same as the case of $\epsilon_x = 0$.

In case of $\gamma_{xy} = 0$ on a certain boundary node,

$$\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = 0 \quad (72)$$

or

$$\frac{\partial u}{\partial y} = - \frac{\partial v}{\partial x} \quad (73)$$

The coefficient of the row and column related to nodal parameter $\partial u/\partial y$ are superimposed by the coefficients, which are multiplied by -1, of the row and column related to $\partial u/\partial x$ and the coefficients related to $\partial u/\partial x$ are eliminated.

SOLUTION VECTOR:SKYLINE PROGRAM

This program finds the solution vectors. In the global stiffness matrix, a number of zero terms must still be stored and manipulated during the solution process. The manipulation of zero terms can be reduced by placing further limitations on the range over which the Gaussian forward elimination algorithms are applied. To impose these limitations, it is convenient to change the sequence of the application of the algorithms. Skylines are found first and all zeros above the skyline of the stiffness matrix are not considered in calculation process. The elements of stiffness matrix are

stored in a one-dimensional array, starting with the diagonal term of each column. The Gaussian algorithms can be modified so that elimination can be carried out directly on the one-dimensional stiffness array. This program was developed in the Advanced Structural Analysis II course (32) and was modified slightly in this study.

REARRANGED PARAMETERS:REPARAM PROGRAM

This program inserts proper values of nodal parameters to the solution vectors found by the use of skyline technique. Zero values of displacement parameters or these dependent nodal parameters were eliminated in the previous program. After restored in order, they can be used for finding strain and stresses at every desired point of the structure.

REPRESENTATION OF COMPLETE DISPLACEMENT VECTOR:COMDISP PROGRAM

In REDUC program, a half of the structure including nodes of center line was considered. In REPARAM program, the solution of a half of the structure was found. Therefore, solution of another half of the structure must be added in actual solution vector. Relationships between nodal parameters which are active nodal displacement and eliminated nodal displacement vectors. By this relationships, the solution vectors which were eliminated during calculating process must be added.

3-D REPRESENTATION OF STRESS:STRESS GRAPH PROGRAM

Nodal displacement vectors were found by COMDISP program. This program calculates the stresses and strains at the points of interest on the shear walls by using the solution

vectors. Three dimensional graphic display of each stress, such as normal stress σ_x , σ_y , and shearing stress τ_{xy} , are plotted by this program.

2-D REPRESENTATION OF DISPLACEMENT: DISPLACE PROGRAM

This program will show the two-dimensional graphic display of the displacement of shear wall by using solution vectors calculated in COMDISP program.

3-D REPRESENTATION OF STRAIN: STRAIN PROGRAM

This program is used to calculate the strains at the points of interest. Three dimensional graphic displays of each strain are plotted .

TRAJECTORIES OF STRESSES:TRAJECTORY PROGRAM

This program finds the trajectories of the stresses and gives the two-dimensional graphic display. After finding normal stresses (σ_x , σ_y) and shearing stress τ_{xy} , the magnitude of the maximum and minimum normal stresses, which are principal stresses, is found.

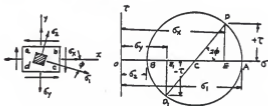


Fig. 38 Mohr's circle for principal stresses

Maximum and minimum normal stresses are

$$\sigma_1 = \overline{OA} = \overline{OC} + \overline{CD} = \frac{\sigma_x + \sigma_y}{2} + \sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \tau_{xy}^2} \quad (74)$$

$$\epsilon_2 = \overline{OB} = \overline{OC} - \overline{CD} = \frac{\epsilon_x + \epsilon_y}{2} - \sqrt{\left(\frac{\epsilon_x - \epsilon_y}{2}\right)^2 + \tau_{xy}^2} \quad (75)$$

The directions of principal stresses can also be calculated from the Fig. 38 . The angle DCA is the double angle between the stress ϵ_1 and the x axis and since 2ϕ is measured from D to A in the clockwise direction, the direction of ϵ_1 must be as indicated in Fig. 38 .

$$\tan 2\phi = \frac{\overline{DE}}{\overline{CE}} \quad (76)$$

Regarding the sign of the angle ϕ , it must be taken negative in this case since it is measured from the x axis in the clockwise direction. Hence

$$\tan 2\phi = - \frac{\overline{DE}}{\overline{CE}} = - \frac{2\tau_{xy}}{\epsilon_x - \epsilon_y} \quad (77)$$

If we take a point at the neutral surface, then ϵ_x becomes zero. An element at this point will be in the condition of pure shear. The directions of the principal stresses will be at 45° to the x and y axes. The trajectories of the stresses are two systems of orthogonal curves whose tangents at each point are in the directions of the principal stresses at this point. All these curves intersect the neutral surface at 45° and have horizontal or vertical tangents at points where shear stress τ_{xy} is zero.

CHAPTER 6

CONCLUSIONS

From this numerical experimental research, the following conclusions can be reached:

1. Finite element method provides point-wise value of stresses and strains in the shear walls.

2. After the reduction of the dimensionality of the matrix equation, the Zenith 100 computer can handle relatively large problem with high efficiency and accuracy. For the use of the micro-computer, the reduction of matrix dimension is necessary if finite element method is used.

3. By correctly finding the principal stresses and principal direction, failure of shear walls can be reduced and the optimal design of reinforcement in the shear walls can be obtained.

4. Linear elastic theory is not adequate for predicting the structural response of the structure like shear wall. It is necessary to develop the non-linear finite element program to understand the non-linear behavior of shear walls.

5. The stress concentration near the intersection of the wall and footing can not be accurately calculated without further development.

6. The experience learned through the development has provided the capability to modify and improve the software available.

7. The micro-computer is too slow for carrying out the project of this size.

8. Buckling analysis of shear walls is not included in this study. For thin shear walls, buckling analysis should be included.

SUGGESTIONS FOR FURTHER RESEARCH

In order to get better results related to the stress analysis of shear walls, the following suggestions for further research are recommended.

1. The number of elements should be increased especially near the points of singularities.

2. Need the development of nonlinear finite element program for the effective solution of shear wall problems.

3. Making the finite element method as a part of optimal design of shear wall.

4. Apply the modified compression-field theory for reinforced concrete elements subjected to shear (33) for the analysis of the shear walls and compare the results to the experimental data.

5. The additional stress due to the interaction of shearing forces, axial forces, and bending moments to be included in the investigation.

REFERENCE

1. Gould, Phillip L., "Interaction of shear wall-frame systems in multistory buildings", American Concrete Institute Journal, Title No. 62.4 Jan.1965, PP. 45-65
2. Coull, Alexander and Choudhury, J.R., "Analysis of coupled shear walls", American Concrete Institute Journal, Title no. 64-51, Sept.1967, PP. 587-593
3. Schwisghofer, Joseph, "Door openings in shear walls". American Concrete Institute Journal, Title no. 64-64, Nov. 1967, PP. 730-734
4. Coull, A., "Interaction of coupled shear walls with elastic foundations", American Concrete Institute Journal, Title no. 68-42, Jun.1971, PP. 456-461
5. Tso, W.K., Chen, P.C.K., "Flexible foundation effect on coupled shear walls", American Concrete Institute Journal, Title no. 69-65, Nov.1972, PP.678-683
6. Arvidsson, Kent, " Shear Walls with door openings near the edge of the wall", American Concrete Institute Journal, Title no. 71-72, Jul.1974, PP.353-357
7. Arvidsson, Kent, "Elastically founded shearwalls with two rows of openings", American Concrete Institute Journal, Title no. 73-13, Mar.1976, PP.151-154
8. Arvidsson, K., "Interaction between coupled shear walls and frames", Proceedings of Institution Civil Engineers, Part 2(GB), Vol. 67, Sept.1979, PP.589-596
9. Coull, A., Low, C.K., "Analysis of stiffened non-planar coupled shear walls", Proceedings of Institution Civil Engineers, Part 2(GB), Vol. 67, Dec.1979, PP.971-986
10. Tso, Wai K., Biswas, Jayanta K., "General analysis of nonplanar coupled shear walls", Journal of the Structural Division Proceedings of the American Society of Civil Engineers, Mar.1973, PP.365-380
11. Biswas, Jayanta K., Tso, Wai K., "Three-dimensional analysis of shear wall buildings to lateral load", Journal of the Structural Division Proceedings of the American Society of Civil Engineers, May.1974
12. Winokur, Arnold., Gluck, Jacob., "Ultimate strength analysis of coupled shear walls", American Concrete Institute Journal, Title no. 65-81, Dec.1968, PP.1029-1036

13. Paulay, Thomas, "Elasto-plastic analysis of coupled shear walls", American Concrete Institute Journal, Title no. 67-60, Nov.1970, PP.915-922
14. Coull, A., Choo, B.S., "Simplified elasto-plastic analysis of coupled shear walls", Proceedings of Institution Civil Engineers, (GB), Vol. 73, Part. 2, Jun.1982, PP.365-381
15. Gluck, Jacob., "Elasto-plastic analysis of coupled shear walls", Journal of the Structural Division Proceedings of the American Society of Civil Engineers, Aug.1973, PP.1743-1759
16. MacLeod, A., "Analysis of the shear wall buildings by the frame method", Proceedings of Institution Civil Engineers, Part. 2, (GB), 1973, PP.593-601
17. MacLeod, A., "General frame element for shear wall analysis", Proceedings of Institution Civil Engineers, Part. 2, (GB), Vol.61, Dec.1976, PP.785-790
18. Heidebrecht, Arthur C., Swift, R. David, "Analysis of asymmetrical coupled shear walls", Journal of the Structural Division Proceedings of the American Society of Civil Engineers, May.1971, PP.1407-1422
19. MacLeod, Iain A., Hosny, Hassan M., "Frame analysis of shear wall cores", Journal of the Structural Division Proceedings of the American Society of Civil Engineers, Oct.1977, PP.2037-2047
20. Nayar, Kumarapillai K., Coull, Alexander, "Elastoplastic analysis of coupled shear walls", Journal of the Structural Division Proceedings of the American Society of Civil Engineers, Sept.1976, PP.1845-1859
21. MacLeod, Iain A., "New rectangular finite element for shear wall analysis", Journal of the Structural Division Proceedings of the American Society of Civil Engineers, Mar.1969, PP.399-409
22. Girijavallabhan, Chiyarath V., "Analysis of shear walls with openings", Journal of the Structural Division Proceedings of the American Society of Civil Engineers, Oct.1969, PP.2093-2103
23. Al-Mahaidi, Riadh S., Nilson, Arthur H., "Coupled shear wall analysis by lagrange multipliers", Journal of the Structural Division Proceedings of the American Society of Civil Engineers, Nov.1975, PP.2359-2366
24. Paulay, Thomas, "Coupling beams of reinforced concrete shear walls", Journal of the Structural Division

- Proceedings of the American Society of Civil Engineers, Mar.1971, PP.843-862
25. Paulay, Thomas, "Simulated seismic loading of spandrel beams", Journal of the Structural Division Proceedings of the American Society of Civil Engineers, Sept.1971, PP.2407-2419
 26. Paulay, Thomas, Santhakumar, Arumbakkam R., "Ductile behavior of coupled shear walls", Journal of the Structural Division Proceedings of the American Society of Civil Engineers, Jan.1976, PP.93-108
 27. Paulay, T., Priestley, M. J. N., Syngge, A. J., "Ductility in earthquake resisting aquat shear walls", American Concrete Institute Journal, Jul-Aug.1982, PP.257-269
 28. Hu, K. K., Wang, S. Y., Swartz, S. E., "Bi-quasi-linear and bi-quasi-quadratic interpolating functions for quadrilateral elements", Proceedings of SECTAM XII, May 10-11, 1984, PP.453-457
 29. Hu, K. K., Kirsser, P. G., Swartz, S. E., Wang, S. Y., "On the construction of n-th order Hermitian shape functions for 2- and 3-dimensional elements", 26th structures, structural dynamics and materials conference Part 1, Apr 15-17,1985 PP.561-564
 30. Zienkiewicz, "The finite element method", third ed. McGRAW-HILL book co., 1977
 31. Strang, G, Fix, G. J, "An analysis of the finite element method", Prantica-Hall,Inc., 1973
 32. Class notes of the Advanced structural analysis II, Department of Civil Engineering, Spring 1985
 33. Vacchio, F. J., Collins, M. P., "The modified compression-field theory for reinforced concrete elements subjected to shear", Journal of the American Concrete Institute, Proceedings V.83 no.2 Mar-Apr,1986, PP.219-231
 34. Building code requirements for rienforced concrete(ACI 318-83) American concrete inatitute
 35. Winter, George, Nilson, Arther H., "Design of concrete structurea", MaGraw-Hill, Inc., 1979

ACKNOWLEDGMENTS

The author wishes to thank his major advisor, Dr. Kuo Kuang Hu, for his invaluable guidance and efforts through the author's entire research program.

The author would also like to extend his sincere thanks to Dr. Harry D. Knostman and Dr. Ted O. Hodges for their service on advisory committee.

Special thanks are given to Hiremagalur Narayan Babu, a former graduate student at KSU, for his help in the early stage of making computer program.

Special thanks must be given to my wife, Jin-Seon, for her support and endurance during the preparation of this thesis.

APPENDIX
LIST OF THE PROGRAMS DEVELOPED

```

10 REM
20 REM                      MAIN PROGRAM
30 REM                      *****
40 REM PROGRAM TO FIND STIFFNESS MATRIX OF THE ELEMENTS BY
   USING FINITE ELEMENT METHOD
50 DIM ND(71,4),NN(4),X(71),Y(71),DC(3,3),B(3,24),DB(3,24),
   SM(24,24),W#(3),H#(3),FX(4),FY(4),PX(4),PY(4),BDB(24,24),
   SX(4),SY(4),SXX(4),SYX(4),SXY(4)
60 DIM SYX(4),T(4),N(2,24),TX(4),TY(4),FFX(2),FFY(2),FXX(2),
   FYX(2),FXY(2),FFY(2),GF(24),FF(24),FC(24),XB(52,14),
   YB(52,14),XE(52,14),BAR(52),TH(4,9)
70 DIM DST(3),DS(3,3),SSM(24,24),CSM(24,24),D(3,3),YE(52,14)
80 EC=2468410!! REM modulus of elasticity of conc.-- unit
   (N/sq.cm)
90 ES=2E+07 :REM modulus of elasticity of steel bar-- unit
   (N/sq.cm)
100 ESEC=ES-EC
110 M=.2:REM poisson's ratio
120 'AHB=2.01:REM area of horizontal bar-- unit(sq.cm)
130 AVB=1.1:REM area of diagonal bar-- unit(sq.cm)
140 TH1=60
150 TH2=10
160 TH3=55
170 REM x,y coordinate of each node
180 NM=71:REM total node numbers
190 NE=52:REM no of total elements
200 FOR I=1 TO NM
210 READ X(I)
220 NEXT I
230 DATA      0,      0,  32.5,  32.5,  40.0,  40.0,  40.0,  40.0,
40.0,  40.0,  40.0
240 DATA  50.0,  50.0,  50.0,  50.0,  50.0,  50.0,  50.0,  50.0
250 DATA  60.0,  60.0,  60.0,  60.0,  60.0,  60.0,  60.0,  60.0
260 DATA 120.0,120.0,120.0,120.0,120.0,120.0,120.0,120.0
270 DATA 190.0,190.0,190.0,190.0,190.0,190.0,190.0,190.0
280 DATA 260.0,260.0,260.0,260.0,260.0,260.0,260.0,260.0
290 DATA 320.0,320.0,320.0,320.0,320.0,320.0,320.0,320.0
300 DATA 330.0,330.0,330.0,330.0,330.0,330.0,330.0,330.0
310 DATA 340.0,340.0,340.0,340.0,340.0,340.0,340.0,340.0
320 DATA 347.5,347.5,380.0,380.0
330 REM
340 FOR I=1 TO NM
350 READ Y(I)
360 NEXT I
370 DATA      0.0,  43.5,193.5,233.5
380 DATA      0.0,  43.5,  53.5,  63.5,118.5,193.5,233.5
390 DATA      0.0,  43.5,  53.5,  63.5,118.5,193.5,233.5
400 DATA      0.0,  43.5,  53.5,  63.5,118.5,193.5,233.5
410 DATA      0.0,  43.5,  53.5,  63.5,118.5,193.5,233.5
420 DATA      0.0,  43.5,  53.5,  63.5,118.5,193.5,233.5
430 DATA      0.0,  43.5,  53.5,  63.5,118.5,193.5,233.5
440 DATA      0.0,  43.5,  53.5,  63.5,118.5,193.5,233.5

```

```

450 DATA 0.0, 43.5, 53.5, 63.5, 118.5, 193.5, 233.5
460 DATA 0.0, 43.5, 53.5, 63.5, 118.5, 193.5, 233.5
470 DATA 193.5, 233.5, 0.0, 43.5
480 REM
490 REM NODE NO. CORRESPONDING TO EACH ELEMENT
500 FOR EL=1 TO NE
510 FOR EN=1 TO 4
520 READ ND(EL,EN)
530 NEXT EN:NEXT EL
540 DATA 1.5,6.2,5.12,13,6,12,19,20,13,19,26,27,20,26,33,34,
27
550 DATA 33,40,41,34,40,47,48,41,47,54,55,48,54,61,62,55,61,7
0,71,62
560 DATA 6,13,14,7,13,20,21,14,20,27,28,21,27,34,35,28,34,41,
42,35
570 DATA 41,48,49,42,48,55,56,49,55,62,63,56,7,14,15,8,14,21,
22,15
580 DATA 21,28,29,22,28,35,36,29,35,42,43,36,42,49,50,43,49,5
6,57,50
590 DATA 56,63,64,57,8,15,16,9,15,22,23,16,22,29,30,23,29,36,
37,30
600 DATA 36,43,44,37,43,50,51,44,50,57,58,51,57,64,65,58,9,16
,17,10
610 DATA 16,23,24,17,23,30,31,24,30,37,38,31,37,44,45,38,44,5
1,52,45
620 DATA 51,58,59,52,58,65,66,59,3,10,11,4,10,17,18,11,17,24,
25,18
630 DATA 24,31,32,25,31,38,39,32,38,45,46,39,45,52,53,46,52,5
9,60,53
640 DATA 59,66,67,60,66,68,69,67
650 REM bar coordinate
660 FOR I=1 TO NE
670 READ BAR(I)
680 NEXT I
690 DATA 0,6,6,10,10,10,10,6,6,0,3,3,6,7,7,6,3,3,3,3,6,7,7,6,
3,3,7,8,12,12,12,12
700 DATA 8,7,10,10,12,14,14,12,10,10,0,6,6,8,10,10,8,6,6,0
710 REM
720 FOR I=1 TO NE
730 BN=BAR(I)
740 IF BN=0 THEN 780
750 FOR J=1 TO BN
760 READ XB(I,J)
770 NEXT J
780 NEXT I
790 DATA 43,43,43,43,43,43,50,50,50,50,57,57,60,60,70,90,110,
60,60,70,90,110,120,120,130,150
800 DATA 170,120,120,130,150,170,190,190,210,230,250,190,190,
210,230,250,260,260
810 DATA 270,290,310,260,260,270,290,310,320,320,320,320,323,
323,330,330,330,330,337,337,43,43,43,50,57,57
820 DATA 70,90,110,60,80,100,130,150,170,120,140,160,180,190,
210,230,250,200

```

```

830 DATA 220,240,270,290,310,260,280,300,320,323,323,330,337,
337,43,43,43,50,57,57,60,80,100,70,90,110,120
840 DATA 140,160,180,130,150,170,200,220,240,190,210,230,250,
260,280,300,270,290
850 DATA 310,320,323,323,330,337,337,43,43,43,43,43,43,50,
50,50,50,55,57,57,60,60,60,70,90,110,60,60
860 DATA 60,75,95,115,120,120,120,130,150,170,120,120,120,135
,155,175,190,190
870 DATA 190,210,230,250,190,190,195,215,235,255,260,260,260,
270,290,310,260,260
880 DATA 260,275,295,315,320,320,320,320,320,320,323,323,330,
330,330,330,330,337,337,43,43,43,45
890 DATA 43,43,43,43,43,43,50,50,50,50,50,50,57,57,60,6
0,60,65,85,105,60,60,60,60,80
900 DATA 100,120,120,120,125,145,165,185,120,120,120,140,
160,180,190,190,190
910 DATA 190,205,225,245,190,190,190,190,200,220,240,260,260,
260,265,285,303,260
920 DATA 260,260,260,280,300,320,320,320,325,320,320,320,320,
323,323,330,330,330,330,330
930 DATA 330,330,330,337,337,43,43,43,43,43,43,50,50,50,50,57
,57,60,60,80,100,60,63,83,103,120,120
940 DATA 140,160,180,120,123,143,163,183,190,190,200,220,240,
190,190,203,223,243
950 DATA 260,260,280,300,260,263,283,303,320,320,320,323,323,
323,330,330,330,330
960 DATA 337,337
970 FOR I=1 TO NE
980 BN=BAR(I)
990 IF BN=0 THEN 1030
1000 FOR J=1 TO BN
1010 READ YB(I,J):YB(I,J)=YB(I,J)+.5
1020 NEXT J
1030 NEXT I
1040 DATA 36,16,10,30,2.5,2.5,23,3,23,43,2.5,2.5,33,13,3,3,3,
13,33,43,43,43,33,13,3,3,3,13,33,43,43
1050 DATA 43,23,3,3,3,3,23,43,43,43,33,13,3,3,3,13,33,43,4
3,43,33,13,33,3,2.5,2.5,23
1060 DATA 3,23,43,2.5,2.5,50,43,43,43,43,43,43,43,43,53,53,53
,43,43,43,53,53,53,53,43,43,43,43,53,53
1070 DATA 53,43,43,43,53,53,53,53,43,43,43,43,43,56,53,53,63,
53,53,53,53,63,63,63,53,53,53,53,63,63
1080 DATA 63,53,53,53,63,63,63,63,53,53,53,53,63,63,63,53,53,
63,53,53,96,76,70,90,110,63,63,103,83
1090 DATA 63,83,103,118,63,63,113,93,73,63,63,63,73,93,113,11
8,118,118,113,93,73,63,63
1100 DATA 63,73,93,113,118,118,118,103,83,63,63,63,63,83,100,
118,118,118,118,113
1110 DATA 93,73,63,63,73,73,93,113,118,118,118,113,93,73,73,9
3,113,63,63,103,83,63,83
1120 DATA 103,63,63,176,156,136,118,130,150,170,190,118,118,1
83,163,143,123,123,143,163,183,118,118
1130 DATA 173,153,133,118,118,118,133,153,173,193,193,193,173

```

```

,153,133,118,118
1140 DATA 118,118,133,153,173,193,193,193,193,183,163,143,123
,118,118,118,123
1150 DATA 143,163,183,193,193,193,173,153,133,118,118,118,133
,153,173,193,193
1160 DATA 193,173,153,133,118,133,153,173,193,118,118,183,163
,143,123,123,143,163,183,118,118
1170 DATA 216,196,210,230,193,193,223,203,203,223,193,193,213
,193,193,193,213,230,230,230,213
1180 DATA 193,193,193,193,213,230,230,230,230,223,203,193,193
,193,203,223,230
1190 DATA 230,230,213,193,193,193,213,230,230,230,213,193,213
,230,193,193,223,203,203
1200 DATA 223,193,193
1210 REM
1220 FOR I=1 TO NE
1230 8N=8AR(I)
1240 IF 8N=0 THEN 1280
1250 FOR J=1 TO 8N
1260 READ XE(I,J)
1270 NEXT J
1280 NEXT I
1290 DATA 50,50,50,50,43,43,60,60,60,60,57,57,70,90,110,120,12
0,70,90,110,120,120,130,150
1300 DATA 170,190,190,130,150,170,190,190,210,230,250,260,260,
210,230,250,260
1310 DATA 260,270,290,310,320,320,270,290,310,320,320,330,330,
330,330,323,323,337,337
1320 DATA 337,337,337,337,50,43,43,60,43,43,80,100,120,70,90,1
10,140,160,180,130,150,170,190,200,220
1330 DATA 240,260,210,230,250,280,300,320,270,290,310,330,323,
323,337,337,337,50,43,43,60,57,57,70,90,110,80
1340 DATA 100,120,130,150,170,190,140,160,180,210,230,250,200,
220,240,260,270,290
1350 DATA 310,280,300,320,330,323,323,337,337,337,50,50,50,50,
50,43,43,60,60,60,60,60,60,60,57,57,65,85,105,120
1360 DATA 120,120,70,90,110,120,120,120,125,145,165,185,190,1
90,130,150,170,190
1370 DATA 190,190,205,225,245,260,260,260,210,230,250,260,260
,260,265,285,305
1380 DATA 320,320,320,270,290,310,320,320,320,325,330,330,330
,330,330,323,323,337,337
1390 DATA 337,337,337,337,50,50,50,50,50,50,50,50,43,43,6
0,60,60,60,55,60,60,60,57,57,80,100,120
1400 DATA 120,120,120,75,95,115,120,120,120,140,160,180,190,1
90,190,190,135,155
1410 DATA 175,190,190,190,190,200,220,240,260,260,260,260,195
,215,235,255,260
1420 DATA 260,260,280,300,320,320,320,320,275,295,315,320,320
,320,330,330,330
1430 DATA 330,330,330,330,330,323,323,337,337,337,337,335,337
,337,337,337,50,50,50,50,43,43,57,60
1440 DATA 60,60,57,57,77,97,117,120,80,100,120,120,137,157,17

```

```

7,190,190,140,160,180,190
1450 DATA 190,197,217,237,257,260,200,220,240,260,260,277,297
,317,320,280,300
1460 DATA 320,320,330,330,330,330,323,323,337,337,337,337,337
,337
1470 FOR I=1 TO NE
1480 BN=BAR(I)
1490 IF BN=0 THEN 1530
1500 FOR J=1 TO BN
1510 READ YE(I,J):YE(I,J)=YE(I,J)+.5
1520 NEXT J
1530 NEXT I
1540 DATA 43,23,3,23,43,43,33,13,13,33,43,43,43,43,43,33,13,3
,3,3,13,33,43,43,43,43,23,3,3,3
1550 DATA 3,23,43,43,43,33,13,3,3,3,13,33,43,43,43,33,13,3,3,
3,13,33,43,23,3,23,43,43
1560 DATA 30,10,16,36,43,43,43,53,53,53,53,53,53,53,53,43,43,
43,53,53,53,43,43,43,43,53,53,53,53
1570 DATA 43,43,43,53,53,53,43,43,43,43,53,53,50,53,53,63,63,
63,53,63,63,63,63,63,53,53,53,63,63,63
1580 DATA 53,53,53,63,63,63,53,53,53,53,63,63,63,53,53,53,63,
63,63,56,63,63,103,83,63,83,103,118,118
1590 DATA 113,93,73,73,93,113,118,118,118,118,118,118,113,93,73,6
3,63,63,73,93,113,118,118
1600 DATA 118,118,103,83,63,63,63,63,83,100,118,118,118,113,9
3,73,63,63,63,73,93
1610 DATA 113,118,118,118,113,93,73,63,63,63,73,93,113,118,10
3,83,63,83,103,118,118,110
1620 DATA 90,70,76,96,118,118,183,163,143,123,123,143,163,183
,193,193,193,173,153,133,118,133
1630 DATA 153,173,193,193,193,193,193,173,153,133,118,118,118,
133,153,173,193,193,193
1640 DATA 183,163,143,123,118,118,118,123,143,163,183,193,193,
193,193,173,153
1650 DATA 133,118,118,118,118,133,153,173,193,193,193,173,153,
133,118,118,118
1660 DATA 133,153,173,183,163,143,123,123,143,163,183,193,193,
190,170,150,130,118,136
1670 DATA 156,176,193,193,223,203,203,223,230,230,230,213,193,
213,230,230,230,230,230,213,193,193,193
1680 DATA 213,230,230,230,223,203,193,193,193,203,223,230,230,
230,230,213,193
1690 DATA 193,193,193,213,230,230,230,213,193,193,193,213,223,
203,203,223,230,230,230
1700 DATA 210,196,216,230,230
1710 FOR I=1 TO NE
1720 BN=BAR(I)
1730 FOR J=1 TO BN
1740 PRINT I,J,XB(I,J),YB(I,J),XE(I,J),YE(I,J)
1750 NEXT J
1760 PRINT
1770 NEXT I
1780 FOR I=1 TO 3

```

```

1790 READ W#(I),H#(I)
1800 NEXT I
1810 REM
1820 DATA -.774596669241483, .555555555555556
1830 DATA .0 , .888888888888889
1840 REM
1850 DATA .774596669241483, .555555555555556
1860 D1=EC/(1-M*M)
1870 DC(1,1)=D1:DC(1,2)=D1*M
1880 DC(2,1)=DC(1,2):DC(2,2)=D1:DC(3,3)=(1-M)*D1/2
1890 OPEN "R".#2,"B:FCDATA",4
1900 FIELD #2,4 AS C#
1910 FOR I=1 TO (NE*24)
1920 B=0
1930 LSET C#=MKS#(B)
1940 PUT #2,I
1950 NEXT I
1960 CLOSE #2
1970 OPEN "R",#1,"B:STDATA",4
1980 FIELD #1,4 AS A#
1990 FOR I=1 TO (NE*576)
2000 A=0
2010 LSET A#=MKS#(A)
2020 PUT #1,I
2030 NEXT I
2040 CLOSE #1
2050 FOR EL=1 TO NE
2060 FOR II=1 TO 24
2070 FOR JJ=1 TO 24
2080 SM(II,JJ)=0:CSM(II,JJ)=0:SSM(II,JJ)=0
2090 NEXT JJ
2100 NEXT II
2110 ST=0
2120 FOR K=1 TO 4
2130 NN(K)=0
2140 NEXT K
2150 PRINT "ELEMENT NO";EL
2160 FOR K=1 TO 4
2170 NN(K)=ND(EL,K):PRINT NN(K);
2180 NEXT K
2190 FOR I=1 TO 3
2200 X#=0:HX#=0
2210 X#=#(I):HX#=#(I)
2220 X=(X(NN(1))+X(NN(2)))/2+(X(NN(2))-X(NN(1)))*X#/2
2230 V=X-X(NN(1)):L=X(NN(2))-X(NN(1))
2240 GOSUB 5000
2250 FX(1)=F1:FX(2)=F2:FX(3)=F3:FX(4)=F4
2260 PX(1)=F5:PX(2)=F6:PX(3)=F7:PX(4)=F8
2270 FOR J=1 TO 3
2280 Y#=0:HY#=0
2290 Y#=#(J):HY#=#(J):HXY#=#(J)*HX#*HY#
2300 Y=(Y(NN(1))+Y(NN(4)))/2+(Y(NN(4))-Y(NN(1)))*Y#/2
2310 V=Y-Y(NN(1)):L=Y(NN(4))-Y(NN(1))

```

```

2320 GOSUB 5000
2330 FY(1)=F1:FY(2)=F2:FY(3)=F3:FY(4)=F4
2340 PY(1)=F5:PY(2)=F6:PY(3)=F7:PY(4)=F8
2350 GOSUB 5100
2360 NEXT J, I
2370 A=(X(NN(2))-X(NN(1)))/2:B=(Y(NN(4))-Y(NN(1)))/2
2380 AB=A*B
2390 IF EL<=10 THEN AB=AB*TH1
2400 IF EL>=43 THEN AB=AB*TH3
2410 IF EL>10 AND EL<43 THEN AB=AB*TH2
2420 FOR II=1 TO 24
2430 FOR JJ=1 TO 24
2440 CSM(II, JJ)=SM(II, JJ)*AB
2450 SM(II, JJ)=0
2460 NEXT JJ, II
2470 REM
2480 PRINT
2490 ST=1
2500 PRINT "   STEEL PART"
2510 PRINT
2520 BN=BAR(EL)
2530 IF BN=0 THEN 2940
2540 FOR EL1=1 TO BN
2550 PRINT "   ELEMENT";EL;" BAR NO";EL1
2560 XB1=XB(EL, EL1)
2570 XE1=XE(EL, EL1)
2580 YB1=YB(EL, EL1)
2590 YE1=YE(EL, EL1)
2600 LENG=SQR((XE1-XB1)^2+(YE1-YB1)^2)
2610 HLEN=LENG/2
2620 ' IF (YE1-YB1)=0 THEN BT=2 ELSE BT=1
2630 SN=(YE1-YB1)/LENG
2640 CS=(XE1-XB1)/LENG
2650 DST(1)=CS^2
2660 DST(2)=SN^2
2670 DST(3)=SN*CS
2680 FOR II=1 TO 3
2690 FOR JJ=1 TO 3
2700 DS(II, JJ)=DST(II)*DST(JJ)
2710 NEXT JJ, II
2720 FOR I=1 TO 3
2730 X#=0:HX#=0
2740 X#=#(I):HX#=#(I):HXY#=#(I)*#(I)
2750 X=(XE1+XB1)/2+(XE1-XB1)*X#/2
2760 V=X-X(NN(1)):L=X(NN(2))-X(NN(1))
2770 GOSUB 5000
2780 FX(1)=F1:FX(2)=F2:FX(3)=F3:FX(4)=F4
2790 PX(1)=F5:PX(2)=F6:PX(3)=F7:PX(4)=F8
2800 Y=(YE1+YB1)/2+(YE1-YB1)*X#/2
2810 V=Y-Y(NN(1)):L=Y(NN(4))-Y(NN(1))
2820 GOSUB 5000
2830 FY(1)=F1:FY(2)=F2:FY(3)=F3:FY(4)=F4
2840 PY(1)=F5:PY(2)=F6:PY(3)=F7:PY(4)=F8

```



```

2850 GOSUB 5100
2860 NEXT I
2870 A=AVB
2880 AB=A*HLEN
2890 FOR II=1 TO 24
2900 FOR JJ=1 TO 24
2910 SSM(II,JJ)=SM(II,JJ)*AB-SSM(II,JJ)
2920 SM(II,JJ)=0
2930 NEXT JJ,II
2940 NEXT EL1
2950 FOR II=1 TO 24
2960 FOR JJ=1 TO 24
2970 SSM(II,JJ)=CSM(II,JJ)+SSM(II,JJ)
2980 NEXT JJ,II
2990 OPEN "R",#1,"B:STDATA",4
3000 FIELD #1,4 AS A#
3010 FOR II=1 TO 24
3020 FOR JJ=1 TO 24
3030 IJ=(II-1)*24+JJ+(EL-1)*24*24
3040 LSET A#=MKS#(SSM(II,JJ))
3050 PUT #1,IJ
3060 NEXT JJ,II
3070 CLOSE #1
3080 'PRINT : 'PRINT
3090 'PRINT "ELEMENT NO.=";EL
3100 'PRINT
3110 'FOR II=1 TO 24
3120 'FOR JJ=1 TO 24
3130 'PRINT "SM(";II;JJ;")=";SM(II,JJ);TAB(30);"SM(";JJ;II;")
=";SM(JJ,II)
3140 'NEXT JJ
3150 'PRINT
3160 'NEXT II
3170 IF EL=>11 AND EL=<42 THEN 4980
3180 FOR I=1 TO 4
3190 SX(I)=0:SY(I)=0:SXX(I)=0:SYX(I)=0:SNX(I)=0:SNY(I)=0:T(I)
=0:TX(I)=0:TY(I)=0
3200 NEXT I
3210 FOR II=1 TO 24
3220 FC(II)=0
3230 NEXT II
3240 FOR I=1 TO 4
3250 READ SX(I),SY(I),SXX(I),SYX(I),SNX(I),SNY(I),T(I),TX(I),
TY(I)
3260 NEXT I
3270 DATA 0,1940.44,0,10.2129,0,0, 0, -8.3100,0
3280 DATA 0,1531.93,0,10.2129,0,0, 297.42,-6.561,0
3290 DATA 0,0,0,0,0,0,0,0,0
3300 DATA 0,0,0,0,0,0,0,0,0
3310 DATA 0,1531.93,0,10.2129,0,0, 297.42,-6.561,0
3320 DATA 0,1429.80,0,10.2129,0,0, 360.84,-6.123,0
3330 DATA 0,0,0,0,0,0,0,0,0
3340 DATA 0,0,0,0,0,0,0,0,0

```

3350 DATA 0,1429.80,0,10.2129,0,0, 360.84,-6.123,0
 3360 DATA 0,1327.67,0,10.2129,0,0, 419.89,-5.686,0
 3370 DATA 0,0,0,0,0,0,0,0,0
 3380 DATA 0,0,0,0,0,0,0,0,0
 3390 DATA 0,1327.67,0,10.2129,0,0, 419.89,-5.686,0
 3400 DATA 0, 714.90,0,10.2129,0,0, 682.32,-3.062,0
 3410 DATA 0,0,0,0,0,0,0,0,0
 3420 DATA 0,0,0,0,0,0,0,0,0
 3430 DATA 0, 714.90,0,10.2129,0,0, 682.32,-3.062,0
 3440 DATA 0, 0 ,0,10.2129,0,0, 789.47, 0,0
 3450 DATA 0,0,0,0,0,0,0,0,0
 3460 DATA 0,0,0,0,0,0,0,0,0
 3470 DATA 0, 0 ,0,10.2129,0,0, 789.47, 0,0
 3480 DATA 0,-714.90,0,10.2129,0,0, 682.32, 3.062,0
 3490 DATA 0,0,0,0,0,0,0,0,0
 3500 DATA 0,0,0,0,0,0,0,0,0
 3510 DATA 0,-714.90,0,10.2129,0,0, 682.32, 3.062,0
 3520 DATA 0,-1327.67,0,10.2129,0,0, 419.89, 5.686,0
 3530 DATA 0,0,0,0,0,0,0,0,0
 3540 DATA 0,0,0,0,0,0,0,0,0
 3550 DATA 0,-1327.67,0,10.2129,0,0, 419.89, 5.686,0
 3560 DATA 0,-1429.8,0,10.2129,0,0, 360.84, 6.123,0
 3570 DATA 0,0,0,0,0,0,0,0,0
 3580 DATA 0,0,0,0,0,0,0,0,0
 3590 DATA 0,-1429.8,0,10.2129,0,0, 360.84, 6.123,0
 3600 DATA 0,-1531.93,0,10.2129,0,0, 297.42, 6.561,0
 3610 DATA 0,0,0,0,0,0,0,0,0
 3620 DATA 0,0,0,0,0,0,0,0,0
 3630 DATA 0,-1531.93,0,10.2129,0,0, 297.42, 6.561,0
 3640 DATA 0,-1940.44,0,10.2129,0,0, 0, 8.31 ,0
 3650 DATA 0,0,0,0,0,0,0,0,0
 3660 DATA 0,0,0,0,0,0,0,0,0
 3670 DATA 0,0,0,0,0,0,0,0,0
 3680 DATA 0,0,0,0,0,0,0,0,0
 3690 DATA 0,0,0,0,0,0, 88.54, 11.518,0
 3700 DATA 0,0,0,0,0,0, 0, 12.094,0
 3710 DATA 0,0,0,0,0,0,0,0,0
 3720 DATA 0,0,0,0,0,0,0,0,0
 3730 DATA 0,0,0,0,0,0,199.88, 10.750,0
 3740 DATA 0,0,0,0,0,0, 88.54, 11.518,0
 3750 DATA 0,0,0,0,0,0,0,0,0
 3760 DATA 0,0,0,0,0,0,0,0,0
 3770 DATA 0,0,0,0,0,0,303.54, 9.982,0
 3780 DATA 0,0,0,0,0,0,199.88, 10.750,0
 3790 DATA 0,0,0,0,0,0,0,0,0
 3800 DATA 0,0,0,0,0,0,0,0,0
 3810 DATA 0,0,0,0,0,0,764.26, 5.375,0
 3820 DATA 0,0,0,0,0,0,303.54, 9.982,0
 3830 DATA 0,0,0,0,0,0,0,0,0
 3840 DATA 0,0,0,0,0,0,0,0,0
 3850 DATA 0,0,0,0,0,0,952.38, 0,0
 3860 DATA 0,0,0,0,0,0,764.26, 5.375,0
 3870 DATA 0,0,0,0,0,0,0,0,0

```

3880 DATA 0,0,0,0,0,0,0,0,0
3890 DATA 0,0,0,0,0,0,764.26, -5.375,0
3900 DATA 0,0,0,0,0,0,952.38, 0,0
3910 DATA 0,0,0,0,0,0,0,0,0
3920 DATA 0,0,0,0,0,0,0,0,0
3930 DATA 0,0,0,0,0,0,303.54, -9.982,0
3940 DATA 0,0,0,0,0,0,764.26, -5.375,0
3950 DATA 0,0,0,0,0,0,0,0,0
3960 DATA 0,0,0,0,0,0,0,0,0
3970 DATA 0,0,0,0,0,0,199.88,-10.750,0
3980 DATA 0,0,0,0,0,0,303.54, -9.982,0
3990 DATA 0,0,0,0,0,0,0,0,0
4000 DATA 0,0,0,0,0,0,0,0,0
4010 DATA 0,0,0,0,0,0, 88.54,-11.518,0
4020 DATA 0,0,0,0,0,0,199.88,-10.750,0
4030 DATA 0,0,0,0,0,0,0,0,0
4040 DATA 0,0,0,0,0,0,0,0,0
4050 DATA 0,0,0,0,0,0, 0,-12.094,0
4060 DATA 0,0,0,0,0,0, 88.54,-11.518,0
4070 FOR SIDE=1 TO 4
4080 FOR I=1 TO 24
4090 GF(I)=0:FF(I)=0
4100 NEXT I
4110 I1=SIDE:I2=SIDE+1
4120 IF I2>4 THEN I2=1
4130 L1=X(NN(I2))-X(NN(I1))
4140 L2=Y(NN(I2))-Y(NN(I1))
4150 LN=SQR(L1*L1+L2*L2)
4160 UX=L2/LN:UY=-L1/LN
4170 FOR M=1 TO 2
4180 IF M=1 THEN Z=I1 ELSE Z=I2
4190 FFX(M)=UX*SX(Z)+UY*T(Z)
4200 FFY(M)=UY*SY(Z)+UX*T(Z)
4210 FXX(M)=UX*SXX(Z)+UY*TX(Z)
4220 FYX(M)=UY*SYX(Z)+UX*TX(Z)
4230 FXY(M)=UX*SXY(Z)+UY*TY(Z)
4240 FYY(M)=UY*SY Y(Z)+UX*TY(Z)
4250 NEXT M
4260 IF SIDE=2 OR SIDE=4 THEN 4340
4270 Y#=-1
4280 IF SIDE=3 THEN Y#=1
4290 Y=(Y(NN(1))+Y(NN(4)))/2+(Y(NN(4))-Y(NN(1)))*Y#/2
4300 V=Y-Y(NN(1)):L=Y(NN(4))-Y(NN(1))
4310 GOSUB 5000
4320 FY(1)=F1:FY(2)=F2:FY(3)=F3:FY(4)=F4
4330 GOTO 4400
4340 X#=-1
4350 IF SIDE=4 THEN X#=-1
4360 X=(X(NN(1))+X(NN(2)))/2+(X(NN(2))-X(NN(1)))*X#/2
4370 V=X-X(NN(1)):L=X(NN(2))-X(NN(1))
4380 GOSUB 5000
4390 FX(1)=F1:FX(2)=F2:FX(3)=F3:FX(4)=F4
4400 FOR GQ=1 TO 3

```

```

4410 IF SIDE=2 OR SIDE=4 THEN 4490
4420 X#=0
4430 X#=#(GQ)
4440 X=(X(NN(1))+X(NN(2)))/2+(X(NN(2))-X(NN(1)))*X#/2
4450 V=X-X(NN(1)):L=X(NN(2))-X(NN(1))
4460 GOSUB 5000
4470 FX(1)=F1:FX(2)=F2:FX(3)=F3:FX(4)=F4
4480 GOTO 4550
4490 Y#=0
4500 Y#=#(GQ)
4510 Y=(Y(NN(1))+Y(NN(4)))/2+(Y(NN(4))-Y(NN(1)))*Y#/2
4520 V=Y-Y(NN(1)):L=Y(NN(4))-Y(NN(1))
4530 GOSUB 5000
4540 FY(1)=F1:FY(2)=F2:FY(3)=F3:FY(4)=F4
4550 FOR II=1 TO 4
4560 C=(II-1)*6
4570 IF II=1 OR II=4 THEN K=1 ELSE K=2
4580 IF II<=2 THEN KK=1 ELSE KK=2
4590 N(1,1+C)=FX(K)*FY(KK)
4600 N(1,3+C)=FX(K+2)*FY(KK)
4610 N(1,5+C)=FX(K)*FY(KK+2)
4620 N(2,2+C)=N(1,1+C)
4630 N(2,4+C)=N(1,3+C)
4640 N(2,6+C)=N(1,5+C)
4650 NEXT II
4660 PX=O:PY=O
4670 FOR M=1 TO 2
4680 IF M=1 THEN PT=(I1-1)*6 ELSE PT=(I2-1)*6
4690 PX=FFX(M)*N(1,PT+1)+FXX(M)*N(1,PT+3)+FXY(M)*N(1,PT+5)+PX
4700 PY=FFY(M)*N(2,PT+2)+FYY(M)*N(2,PT+4)+FYY(M)*N(2,PT+6)+PY
4710 NEXT M
4720 FOR I=1 TO 24
4730 GF(I)=N(1,I)*PX+N(2,I)*PY
4740 NEXT I
4750 FOR I=1 TO 24
4760 FF(I)=FF(I)+GF(I)*H#(GQ)
4770 NEXT I
4780 NEXT GQ
4790 FOR I=1 TO 24
4800 IF SIDE=1 OR SIDE=3 THEN L=(X(NN(2))-X(NN(1)))/2 ELSE
L=(Y(NN(4))-Y(NN(1)))/2
4810 FC(I)=FF(I)*L+FC(I)
4820 NEXT I
4830 NEXT SIDE
4840 PRINT :PRINT
4850 PRINT "FORCE VECTOR OF ELEMENT NO.=":EL
4860 PRINT :PRINT
4870 FOR I=1 TO 24
4880 PRINT "FC(":I:")=":FC(I)
4890 NEXT I
4900 OPEN "R",#2,"B:FCDATA",4
4910 FIELD #2,4 AS C#
4920 FOR II=1 TO 24

```

```

4930 PT=(EL-1)*24*II
4940 LSET CS=HKS#(FC(II))
4950 PUT #2,PT
4960 NEXT II
4970 CLOSE #2
4980 NEXT EL
4990 END
5000 LV=L-V
5010 F1=LV/L+V*LV*(L-2*V)/L^3
5020 F2=V/L+V*LV*(2*V-L)/L^3
5030 F3=V*LV^2/L^2
5040 F4=-LV*V^2/L^2
5050 F5=(-6*V*L+6*V^2)/L^3
5060 F6=(6*V*L-6*V^2)/L^3
5070 F7=(L^2-4*L*V+3*V^2)/L^2
5080 F8=(-2*L*V+3*V^2)/L^2
5090 RETURN
5100 FOR II=1 TO 4
5110 C=(II-1)*6
5120 IF II=1 OR II=4 THEN K=1 ELSE K=2
5130 IF II <=2 THEN KK=1 ELSE KK=2
5140 B(1,1+C)=PX(K)*FY(KK):B(2,2+C)=FX(K)*PY(KK)
5150 B(1,3+C)=PX(K+2)*FY(KK):B(2,4+C)=FX(K+2)*PY(KK)
5160 B(1,5+C)=PX(K)*FY(KK+2):B(2,6+C)=FX(K)*PY(KK+2)
5170 B(3,1+C)=B(2,2+C):B(3,2+C)=B(1,1+C)
5180 B(3,3+C)=B(2,4+C):B(3,4+C)=B(1,3+C)
5190 B(3,5+C)=B(2,6+C):B(3,6+C)=B(1,5+C)
5200 NEXT II
5210 FOR II=1 TO 3
5220 FOR JJ=1 TO 3
5230 IF ST=0 THEN 5240 ELSE 5260
5240 D(II,JJ)=DC(II,JJ)
5250 GOTO 5270
5260 D(II,JJ)=DS(II,JJ)
5270 NEXT JJ
5280 NEXT II
5290 FOR II=1 TO 3
5300 FOR JJ=1 TO 24
5310 DB(II,JJ)=0
5320 FOR K=1 TO 3
5330 DB(II,JJ)=DB(II,JJ)+D(II,K)*B(K,JJ)
5340 NEXT K,JJ,II
5350 FOR II=1 TO 24
5360 FOR JJ=1 TO 24
5370 BDB(II,JJ)=0
5380 FOR K=1 TO 3
5390 BDB(II,JJ)=BDB(II,JJ)+B(K,II)*DB(K,JJ)
5400 NEXT K,JJ,II
5410 FOR II=1 TO 24
5420 FOR JJ=1 TO 24
5430 SH(II,JJ)=SM(II,JJ)+BDB(II,JJ)*HXY#
5440 NEXT JJ,II
5450 RETURN

```

```

10 REM          PROGRAM NAME : ASSEM. PROGRAM
20 REM          *****
30 REM Program of assembling the element stiffness matrix
  stored in the random
40 REM access file "STDATA" and of converting to the global
  stiffness matrix which will be stored in a new random
  access file "STDATAN"
60 B=0:S=0:Z=0:ZZ=0:Z1=0:ZZ1=0:BB=0:SS=0:BK=0:BS=0
70 REM
80 DIM ND(4,4)
90 FOR I=1 TO 4
100 FOR J=1 TO 4
110 READ ND(I,J)
120 NEXT J,I
130 DATA 1, 4, 5, 2, 4, 7, 8, 5, 2, 5, 6, 3, 5, 8, 9, 6
140 REM
150 REM
160 OPEN "R",#3,"B:STDATAN",4
170 FIELD #3,4 AS BS
180 FOR I=1 TO 1500
190 B=0
200 LSET BS=MKS$(B)
210 PUT #3,I
220 NEXT I
230 CLOSE #3
240 B=0
250 OPEN "R",#1,"B:STDATA",4
260 FIELD #1,4 AS AS
270 OPEN "R",#2,"B:FCDATA",4
280 FIELD #2,4 AS CS
290 OPEN "R",#3,"B:STDATAN",4
300 FIELD #3,4 AS BS
310 OPEN "R",#4,"B:FCDATAN",4
320 FIELD #4,4 AS DS
330 FOR I=1 TO 4
340 FOR J=1 TO 4
350 REM
360 REM
370 REM
380 NJ=ND(I,J):PRINT NJ
390 NR=(NJ-1)*6
400 FOR K=1 TO 6
410 NK=NR+K:REM NK is the row no. in the global stiffness
  matrix
420 ELM=(NK-1)*NK/2:REM Total nos of lower triangular element
  before node no. considered.
440 SR=(J-1)*6+K:REM SR is the row no. in the element
  stiffness matrix
450 FOR L=1 TO 4
460 NL=ND(I,L)
470 CN=(NL-1)*6
480 FOR M=1 TO 6

```

```

490 KC=CN*M:REM KC is the column no. in the global stiffness
matrix
500 SC=(L-1)*6*M
510 IF KC>NK THEN 700
520 REM Z is the position of the element "STDATA"
522 REM global stiffness matrix
530 Z=(SR-1)*24+SC+(I-1)*24*24
540 REM ZZ is the corresponding position of the element in
"STDATAN"
550 ZZ=ELM+KC
560 GET #1,Z
570 S=CVS(AS)
580 GET #3,ZZ
590 B=CVS(BS):BK=B
600 B=B+S:BS=B
610 LSET BS=MKS(B)
620 PRINT "Z=";Z;TAB(10);"ZZ=";ZZ;TAB(21);"S=";S;TAB(38);"B="
;BK;TAB(56);"BS=";BS
630 REM
640 REM Z is the position of the element in the element
stiffness matrix stored 530 REM in a 1-dimensional array,
and zz is its corresponding position in the
660 REM
670 PUT #3,ZZ
680 B=0:S=0:Z=0:ZZ=0:BS=0:BK=0
690 'PRINT "B=";B;"S=";S;"KC=";KC;"NK=";NK;"ZZ=";ZZ;"Z=";Z
700 NEXT M
710 NEXT L
720 Z1=(I-1)*24*(J-1)*6+K
730 ZZ1=NK
740 GET #2,Z1
750 SS=CVS(CS)
760 GET #4,ZZ1
770 BB=CVS(DS)
780 BB=BB+SS
790 LSET DS=MKS(BB)
800 PUT #4,ZZ1
810 BB=0:SS=0:Z1=0:ZZ1=0
820 NEXT K
830 NEXT J
840 NEXT I
850 CLOSE #1
860 CLOSE #2
870 CLOSE #3
880 CLOSE #4
890 END

```

```

10 REM          PROGRAM NAME:  REDUC. PROGRAM
20 REM          *****
30 REM Program of assembling the element stiffness matrix
  stored in the random access file "STDATA" and of converting
40 REM to the reduced global stiffness matrix which will be
  stored in a new random access file "STDATAN"
60 B=0:S=0:Z=0:ZZ=0:Z1=0:ZZ1=0:BB=0:SS=0:BK=0:BS=0
70 REM
80 DIM ND(54,4),COMP(426,2)
90 TN=71:REM TN=TOTAL NODE NUMBER
100 N=TN*6
110 TEL=52:REM TEL=TOTAL ELEMENT NO.
112 LN=39:REM LAST NODE NUMBER OF THE REDUCED STRUCTURE
113 L1=LN*6:L2=L1+1
120 FOR I=1 TO TEL
130 FOR J=1 TO 4
140 READ ND(I,J)
150 NEXT J,I
160 DATA 1, 5, 6, 2, 5,12,13, 6, 12,19,20,13, 19,26,27,20,
  26,33,34,27
170 DATA 33,40,41,34, 40,47,48,41, 47,54,55,48, 54,61,62,55,
  61,70,71,62
180 DATA 6,13,14, 7, 13,20,21,14, 20,27,28,21, 27,34,35,28,
  34,41,42,35
190 DATA 41,48,49,42, 48,55,56,49, 55,62,63,56, 7,14,15, 8,
  14,21,22,15
200 DATA 21,28,29,22, 28,35,36,29, 35,42,43,36, 42,49,50,43,
  49,56,57,50
210 DATA 56,63,64,57, 8,15,16, 9, 15,22,23,16, 22,29,30,23,
  29,36,37,30
220 DATA 36,43,44,37, 43,50,51,44, 50,57,58,51, 57,64,65,58,
  9,16,17,10
230 DATA 16,23,24,17, 23,30,31,24, 30,37,38,31, 37,44,45,38,
  44,51,52,45
240 DATA 51,58,59,52, 58,65,66,59, 3,10,11, 4, 10,17,18,11,
  17,24,25,18
250 DATA 24,31,32,25, 31,38,39,32, 38,45,46,39, 45,52,53,46,
  52,59,60,53
260 DATA 59,66,67,60, 66,68,69,67
270 IJ=0
280 FOR I=1 TO L1
290 COMP(I,1)=I
300 NEXT I
310 FOR I=L2 TO N
320 READ COMP(I,1)
330 NEXT I
340 DATA 151,152,153,154,155,156,157,158,159,160,161,162,163,
  164,165,166,167,168
342 DATA 169,170,171,172,173,174,175,176,177,178,179,180,181,
  182,183,184,185,186
344 DATA 187,188,189,190,191,192,109,110,111,112,113,114,115,
  116,117,118,119,120

```



```

346 DATA 121,122,123,124,125,126,127,128,129,130,131,132,133,
134,135,136,137,138
348 DATA 139,140,141,142,143,144,145,146,147,148,149,150
403 DATA 67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,8
4,85,86,87,88,89,90
404 DATA 91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,1
06,107,108
410 DATA 25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,4
2,43,44,45,46,47
420 DATA 48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,6
5,66
430 DATA 13,14,15,16,17,18,19,20,21,22,23,24
440 DATA 1,2,3,4,5,6,7,8,9,10,11,12
450 FOR I=1 TO L1
460 COMP(I,2)=1
470 NEXT I
480 FOR I=L2 TO N
490 READ COMP(I,2)
500 NEXT I
510 DATA 1,-1,-1,1,1,-1, 1,-1,-1,1,1,-1, 1,-1,-1,1,1,-1, 1,-
1,-1,1,1,-1
520 DATA 1,-1,-1,1,1,-1, 1,-1,-1,1,1,-1, 1,-1,-1,1,1,-1, 1,-
1,-1,1,1,-1
530 DATA 1,-1,-1,1,1,-1, 1,-1,-1,1,1,-1, 1,-1,-1,1,1,-1, 1,-
1,-1,1,1,-1
540 DATA 1,-1,-1,1,1,-1, 1,-1,-1,1,1,-1, 1,-1,-1,1,1,-1, 1,-
1,-1,1,1,-1
550 DATA 1,-1,-1,1,1,-1, 1,-1,-1,1,1,-1, 1,-1,-1,1,1,-1, 1,-
1,-1,1,1,-1
560 DATA 1,-1,-1,1,1,-1, 1,-1,-1,1,1,-1, 1,-1,-1,1,1,-1, 1,-
1,-1,1,1,-1
570 DATA 1,-1,-1,1,1,-1, 1,-1,-1,1,1,-1, 1,-1,-1,1,1,-1, 1,-
1,-1,1,1,-1
580 DATA 1,-1,-1,1,1,-1, 1,-1,-1,1,1,-1, 1,-1,-1,1,1,-1, 1,-
1,-1,1,1,-1
590 OPEN "R",#3,"B:STDATAN",4
600 FIELD #3,4 AS B$
610 FOR I=1 TO (L1*L2/2)
620 B=0
630 LSET B$=MK$$(B)
640 PUT #3,I
650 NEXT I
660 CLOSE #3
670 OPEN "R",#4,"B:FCDATAN",4
680 FIELD #4,4 AS D$
690 FOR I=1 TO L1
700 B=0
710 LSET D$=MK$$(B)
720 PUT #4,I
730 NEXT I
740 CLOSE #4
750 OPEN "R",#1,"B:STDATA",4
760 FIELD #1,4 AS A$

```

```

770 OPEN "R",#2,"B:FCDATA",4
780 FIELD #2,4 AS CS
790 OPEN "R",#3,"B:STDATAN",4
800 FIELD #3,4 AS B$
810 OPEN "R",#4,"B:FCDATAN",4
820 FIELD #4,4 AS D$
830 FOR ELMT=1 TO TEL
835 PRINT "ELEMENT NO";ELMT
840 FOR RCRN=1 TO 4
850 REM
860 REM
870 REM
880 NP=ND(ELMT,RCRN):PRINT NP
890 NR=(NP-1)*6
900 FOR K=1 TO 6
910 NNK=NR+K:REM NK is the row no. in the global stiffness
matrix
920 NK=COMP(NNK,1)
930 FC=COMP(NNK,2)
940 ELM=(NK-1)*NK/2:REM Total noa of lower triangular element
before node no. considered.
960 SR=(RCRN-1)*6+K:REM SR is the row no. in the element
stiffness matrix
970 FOR L=1 TO 4
980 NL=ND(ELMT,L)
990 CN=(NL-1)*6
1000 FOR M=1 TO 6
1010 KKC=CN+M:REM KC is the coluan no in the global stiffness
matrix
1020 KC=COMP(KKC,1):FFC=COMP(KKC,2)
1030 SC=(L-1)*6+M
1040 IF KC>NK THEN 1230
1050 REM Z is the position of the element "STDATA"
1060 Z=(SR-1)*24+SC+(ELMT-1)*24*24
1070 REM ZZ is the corresponding position of the element in
"STDATAN"
1080 GET #1,Z
1090 S=CVS(A$)
1100 ZZ=ELM+KC
1110 GET #3,ZZ
1120 B=CVS(B$):BK=B
1130 B=B+S*FC*FFC
1140 LSET B$=MKSS(B)
1150 PRINT"Z=";Z;TAB(9);"ZZ=";ZZ;TAB(19);"S=";S;TAB(35);"B="
;BK;TAB(53);"BS=";B;TAB(67);"FC=";FC;TAB(73);"FFC=";FFC
1160 REM
1170 REM Z is the position of the element in the element
stiffness matrix stored 530 REM in a 1-dimensional array
, and zz is its corresponding position in the
1180 REM global stiffness matrix
1190 REM
1200 PUT #3,ZZ
1210 B=0:S=0:Z=0:ZZ=0:BK=0

```

```
1220 'PRINT "B=";B;"S=";S;"KC=";KC;"NK=";NK;"ZZ=";ZZ;"Z=";Z
1230 NEXT M
1240 NEXT L
1250 Z1=(ELMT-1)*24+(RCRN-1)*6+K
1260 ZZ1=NK
1270 GET #2,Z1
1280 SS=CVS(C#)
1290 GET #4,ZZ1
1300 BB=CVS(D#)
1310 BB=BB+SS*FC
1320 LSET D#=MKS$(BB)
1330 PUT #4,ZZ1
1340 BB=0;SS=0;Z1=0;ZZ1=0
1350 NEXT K
1360 NEXT RCRN
1370 NEXT ELMT
1380 CLOSE #1
1390 CLOSE #2
1400 CLOSE #3
1410 CLOSE #4
1420 END
```

```

10 REM          PROGRAM NAME : MODIF PROGRAM
20 REM          *****
30 REM In this progra boundary condition is considered. In
  case of fixed end,
40 REM set rows and columns of the global stiffness matrix to
  be zero except
50 REM making diagonal elements be zero.
60 REM
70 DIM ZERO(6)
80 FOR I=1 TO 6
90 READ ZERO(I)
100 NEXT I
110 REM
120 DATA 1,2,19,20,37,38
130 FOR I=1 TO 38
140 P=0
150 REM
160 REM
170 FOR J=1 TO 6
180 IF P=1 THEN 200
190 IF I=ZERO(J) THEN P=1
200 NEXT J
210 IF P=0 THEN 330
220 PRINT I,P
230 FOR K=1 TO I
240 S=0
250 IF I=K THEN S=1
260 IK=((I-1)+I/2)+K
270 OPEN "R",#3,"B:STDATAN",4
280 FIELD #3,4 AS A$
290 LSET A$=MK$$(S)
300 PUT #3,IK
310 CLOSE #3
320 NEXT K
330 NEXT I
340 FOR I=1 TO 38
350 P=0
360 REM
370 REM The following makes the coluan elements be zero
380 REM
390 FOR J=1 TO 6
400 IF P=1 THEN 420
410 IF I=ZERO(J) THEN P=1
420 NEXT J
430 IF P=0 THEN 540
440 PRINT I,P
450 FOR K=I TO 53
460 S=0
470 IK=K*(K+1)/2+I
480 OPEN "R",#3,"B:STDATAN",4
490 FIELD #3,4 AS A$
500 LSET A$=MK$$(S)

```

```
510 PUT #3,IK
520 CLOSE #3
530 NEXT K
540 NEXT I
550 FOR I=1 TO 6
560 OPEN "R",#4,"B:FCDATAN",4
570 FIELD #4,4 AS B#
580 S=0
590 K=ZERO(I)
600 PRINT K
610 LSET B#=MK$$(S)
620 PUT #4,K
630 CLOSE #4
640 NEXT I
650 END
```

```

10 REM
20 REM          PROGRAM NAME: ENBOUN
30 REM          *****
40 REM This program considers matrix superposition of the
   known nodal parameters
50 REM and deletes the rows and columns which have zero nodal
   parameters, and finally gives the reduced lower triangular
   matrix.
70 DIM ND(50),ACTND(50,3),ZERO(50,9),CBN(3),WK1(300),WK2(300)
   ,WK3(300),WK4(300)
80 DIM ZERO1(50)
90 INPUT"POISSON'S RATIO";M
100 INPUT"TOTAL NODE NO";TN
110 TL=TN*6
120 INPUT"HOW MANY NODES ARE CONSIDERED";HN
130 FOR I=1 TO HN
140 PRINT I
150 INPUT"NODE NUMBER";ND(I)
160 PT=(ND(I)-1)*6
170 INPUT"      SIGMA X=0 (yes-----type 1  else  type 0) ";Z
180 IF Z=1 THEN 190 ELSE 200
190 ACTND(I,1)=PT+6;ZERO(I,1)=PT+3;GOTO 210
200 ACTND(I,1)=0
210 INPUT"      TAU XY=0 (yes-----type 1  else  type 0) ";Z
220 IF Z=1 THEN 230 ELSE 240
230 ACTND(I,2)=PT+4;ZERO(I,2)=PT+5;GOTO 250
240 ACTND(I,2)=0
250 INPUT"      SIGMA Y=0 (yes-----type 1  else  type 0) ";Z
260 IF Z=1 THEN 270 ELSE 280
270 ACTND(I,3)=PT+3;ZERO(I,3)=PT+6;GOTO 290
280 ACTND(I,3)=0
290 INPUT"      U=0          (yes-----type 1  else  type 0)
   ";Z
300 IF Z=1 THEN ZERO(I,4)=PT+1 ELSE ZERO(I,4)=0
310 INPUT"      V=0          (yes-----type 1  else  type 0)
   ";Z
320 IF Z=1 THEN ZERO(I,5)=PT+2 ELSE ZERO(I,5)=0
330 INPUT"      U,y=0       (yes-----type 1  else  type 0)
   ";Z
340 IF Z=1 THEN ZERO(I,6)=PT+5 ELSE ZERO(I,6)=0
350 INPUT"      V,x=0       (yes-----type 1  else  type 0)
   ";Z
360 IF Z=1 THEN ZERO(I,7)=PT+4 ELSE ZERO(I,7)=0
370 INPUT"      SIGMA X=0 AND SIGMA Y=0 (yes-----type 1
   else  type 0) ";Z
380 IF Z=1 THEN 390 ELSE 400
390 ZERO(I,8)=PT+3;ZERO(I,9)=PT+6;GOTO 410
400 ZERO(I,8)=0;ZERO(I,9)=0
410 NEXT I
420 FOR I=1 TO HN
430 LPRINT "ACTND(";I;")=";ACTND(I,1);TAB(20);"ACTND(";I;")2)
   =";ACTND(I,2);TAB(40);"ACTND(";I;")3)=";ACTND(I,3)

```

```

440 NEXT I
450 CBN(1)=-M:CBN(2)=-1:CBN(3)=-M
460 OPEN "R",#4,"B:FCDATAN",4
470 FIELD #4,4 AS DS
480 OPEN "R",#3,"B:STDATAN",4
490 FIELD #3,4 AS AS
500 FOR CASE=1 TO 3
510 FOR I=1 TO HN
520 PRINT I
530 IK=0:IK1=0
540 IF ACTND(I,CASE)>0 THEN 550 ELSE 1080
550 XN=ACTND(I,CASE)
560 IF CASE=1 THEN XN1=XN-3
570 IF CASE=2 THEN XN1=XN+1
580 IF CASE=3 THEN XN1=XN+3
590 IK=(XN-1)*XN/2
600 FOR J=1 TO XN
610 PT=IK+J
620 GET #3,PT
630 WK1(J)=CVS(A#):WK3(J)=WK1(J)
640 NEXT J
650 IK1=(XN1-1)*XN1/2
660 FOR J=1 TO XN1
670 PT1=IK1+J
680 GET #3,PT1
690 WK2(J)=CVS(A#):WK4(J)=WK2(J)
700 NEXT J
710 FOR K=XN TO TL-1
720 PT=K*(K+1)/2+XN
730 GET #3,PT
740 WK1(K+1)=CVS(A#):WK3(K+1)=WK1(K+1)
750 NEXT K
760 FOR K=XN1 TO TL-1
770 PT1=K*(K+1)/2+XN1
780 GET #3,PT1
790 WK2(K+1)=CVS(A#):WK4(K+1)=WK2(K+1)
800 NEXT K
810 FOR J=1 TO TL
820 WK1(J)=WK1(J)+WK2(J)*CBN(CASE)
830 NEXT J
840 WK4(XN)=WK1(XN1)
850 WK3(XN)=WK1(XN)
860 FOR J=1 TO TL
870 WK3(J)=WK3(J)+WK4(J)*CBN(CASE)
880 NEXT J
890 IK=(XN-1)*XN/2
900 FOR K=1 TO XN-1
910 PT=IK+K
920 LSET A#MKS#(WK1(K))
930 PUT #3,PT
940 NEXT K
950 FOR K=XN TO TL
960 PT=(K-1)*K/2+XN

```

```

970 LSET A#=MKS$(WK3(K))
980 PUT #3,PT
990 NEXT K
1000 GET #4,XN
1010 BB=CVS(D$)
1020 GET #4,XN1
1030 BB1=CVS(D$)
1040 BB=BB+BB1*CBN(CASE)
1050 LSET D$=MKS$(BB)
1060 PUT #4,XN
1070 BB=0:BB1=0
1080 NEXT I
1090 NEXT CASE
1100 CLOSE #3
1110 CLOSE #4
1120 PT=0
1130 FOR I=1 TO HN
1140 FOR J=1 TO 9
1150 IF ZERO(I,J)>0 THEN 1160 ELSE 1180
1160 PT=PT+1
1170 ZERO1(PT)=ZERO(I,J)
1180 NEXT J
1190 NEXT I
1200 FOR I=1 TO PT
1210 FOR J=1 TO PT-1
1220 IF ZERO1(J)>ZERO1(J+1) THEN 1230 ELSE 1260
1230 TEMP=ZERO1(J)
1240 ZERO1(J)=ZERO1(J+1)
1250 ZERO1(J+1)=TEMP
1260 NEXT J
1270 NEXT I
1280 FOR I=1 TO PT
1290 LPRINT "ZERO1(";I;")=";ZERO1(I)
1300 NEXT I
1310 REROW=TL-PT:REEL=REROW*(REROW+1)/2
1320 OPEN"R",#2,"B:RSTDATA",4
1330 FIELD #2,4 AS B$
1340 FOR I=1 TO REEL
1350 B=0
1360 LSET B$=MKS$(B)
1370 PUT #2,I
1380 NEXT I
1390 CLOSE #2
1400 OPEN"R",#1,"B:RFCDATA",4
1410 FIELD #1,4 AS C$
1420 FOR I=1 TO REROW
1430 B=0
1440 LSET C$=MKS$(B)
1450 PUT #1,I
1460 NEXT I
1470 CLOSE #1
1480 ZZ=0:ZZ1=0
1490 OPEN"R",#4,"B:FCDATAN",4

```



```
1500 FIELD #4,4 AS D$
1510 OPEN"R",#3,"B:STDATAN",4
1520 FIELD #3,4 AS A$
1530 OPEN"R",#2,"B:RSTDATA",4
1540 FIELD #2,4 AS B$
1550 OPEN"R",#1,"B:RFCDATA",4
1560 FIELD #1,4 AS C$
1570 FOR I=1 TO TL
1580 PRINT I
1590 FOR IJ=1 TO PT
1600 RN=ZERO1(IJ)
1610 IF I=RN THEN 1800
1620 NEXT IJ
1630 FOR J=1 TO I
1640 FOR IJ=1 TO PT
1650 NC=ZERO1(IJ)
1660 IF J=NC THEN 1740
1670 NEXT IJ
1680 ZZ=ZZ+1
1690 IJK=I*(I-1)/2+J
1700 GET #3,IJK
1710 ST=CVS(A$)
1720 LSET B$=MKS$(ST)
1730 PUT #2,ZZ
1740 NEXT J
1750 ZZ1=ZZ1+1
1760 GET #4,I
1770 FC=CVS(D$)
1780 LSET C$=MKS$(FC)
1790 PUT #1,ZZ1
1800 NEXT I
1810 CLOSE #1
1820 CLOSE #2
1830 CLOSE #3
1840 CLOSE #4
1850 END
```

```

10 REM
20 REM      PROGRAM NAME: SKYLINE
30 REM      *****
40 REM THIS PROGRAM FIND THE SOLUTION VECTORS BY USING
   SKYLINE TECHNIQUE.
50 REM
60 INPUT "TOTAL ROWS":N
70 NP=N+1:NQ=N+2
80 DIM MI(242),PR(242),X(240)
90 OPEN "R",#3,"B:RSTDATA",4
100 FOR I=1 TO N
110 IK=I*(I-1)/2
120 FOR J=1 TO I
130 FIELD #3,4 AS A#
140 IJ=IK+J
150 GET #3,IJ
160 B=CVS(A#)
170 IF B=0 THEN 210
180 IF B>0 AND MI(I)=0 THEN MI(I)=J
190 PRINT "MI(";I;")=";MI(I)
200 GOTO 220
210 NEXT J
220 NEXT I
230 CLOSE #3
240 OPEN "R",#4,"B:RFCDATA",4
250 FOR I=1 TO N
260 FIELD #4,4 AS B#
270 GET #4,I
280 A=CVS(B#)
290 LPRINT "FC(";I;")=";A
300 NEXT I
310 FOR I=1 TO N
320 FIELD #4,4 AS B#
330 GET #4,I
340 A=CVS(B#)
350 IF A=0 THEN 370
360 MI(NP)=I:PRINT "MII(";NP;")=";MI(NP):GOTO 380
370 NEXT I
380 CLOSE #4
390 M=1
400 FOR I=1 TO N
410 M1=I-MI(I)+1
420 IF M<M1 THEN M=M1
430 NEXT I
440 PR(1)=0
450 FOR I=1 TO N
460 PR(I+1)=PR(I)+I-MI(I)+1
470 PRINT "PR(";I+1;")=";PR(I+1)
480 NEXT I
490 PR(NQ)=PR(NP)+N-MI(NP)+1
500 NT=PR(NQ)
510 DIM S(14000)

```

```

520 LPRINT "NT=";NT
530 OPEN "R",#3,"B:RSTDATA",4
540 FIELD #3,4 AS A#
550 IJ=0
560 FOR I=1 TO N
570 PRINT "I=";I
580 J1=MI(I)
590 K=PR(I)
600 IK=I+(I-1)/2:PRINT "IK=";IK
610 FOR J=J1 TO I
620 IJ=IK+J:PRINT "IJ=";IJ
630 K=K+1
640 GET #3,IJ
650 S(K)=CVS(A#)
660 PRINT "SK(";K;")=";S(K)
670 NEXT J
680 NEXT I
690 CLOSE #3
700 OPEN "R",#4,"B:RFCDATA",4
710 FIELD #4, 4 AS B#
720 KK=PR(NP)
730 K=0
740 II=MI(NP)
750 FOR J=II TO N
760 GET #4,J
770 K=K+1
780 S(KK+K)=CVS(B#)
790 PRINT "SK(";KK+K;")=";S(KK+K)
800 NEXT J
810 CLOSE #4
820 REM The following is the elimination
830 N1=N-1
840 FOR I=2 TO N
850 I2=I-M-1:IF I2>N THEN I2=N
860 FOR J=I TO I2
870 R1=MI(I):IF MI(J)>MI(I) THEN R1=MI(J)
880 I1=I-1:IF R1>I1 THEN 940
890 FOR R=R1 TO I1
900 IJ=PR(J)+I-MI(J)+1:RI=PR(I)+R-MI(I)+1:RJ=PR(J)+R-MI(J)+1
910 RR=PR(R)+R-MI(R)+1
920 S(IJ)=S(IJ)-S(RI)*S(RJ)/S(RR)
930 NEXT R
940 NEXT J
950 R1=MI(NP):IF MI(I)>MI(NP) THEN R1=MI(I)
960 IF R1>I-1 THEN 1030
970 FOR R=R1 TO I1
980 RI=PR(I)+R-MI(I)+1:RR=PR(R)+R-MI(R)+1:IJ=PR(NP)+I-MI(NP)
+1
990 RJ=PR(NP)+R-MI(NP)+1
1000 S(IJ)=S(IJ)-S(RI)*S(RJ)/S(RR)
1010 PRINT S(IJ)
1020 NEXT R
1030 NEXT I

```

```

1040 FOR I=1 TO MI(NP)-1:X(I)=0:NEXT I
1050 FOR I=MI(NP) TO N :RI=PR(NP)+I-MI(NP)+1:X(I)=S(RI):NEXT
I
1060 RR=PR(N)+N-MI(N)+1
1070 X(N)=X(N)/S(RR)
1080 FOR II=1 TO N1
1090 NR=N+1-II:XR=X(NR)
1100 J1=MI(NR):J2=NR-1:IR=PR(NR)-MI(NR)+1
1110 FOR I=J1 TO J2
1120 IJ=IR+I
1130 X(I)=X(I)-S(IJ)*XR
1140 NEXT I
1150 RR=PR(J2)+J2-MI(J2)+1
1160 X(J2)=X(J2)/S(RR)
1170 NEXT II
1180 LPRINT "SOLUTION VECTOR"
1190 FOR I=1 TO N
1200 LPRINT "X(";I;")----- ";X(I)
1210 NEXT I
1220 OPEN "R",#5,"B:DISP",4
1230 FIELD #5, 4 AS D#
1240 FOR I=1 TO N
1250 LSET D#=MKS#(X(I))
1260 PUT #5,I
1270 NEXT I
1280 CLOSE #5
1290 END

```

```

10 REM
20 REM          PROGRAM NAME: REPARAM
30 REM          *****
40 REM This program adds all known nodal parameters to the
   reduced nodal parameters.
50 DIM ACTND(50),ZERO1(100),U(300),UU(300)
60 M=.17
70 INPUT "TOTAL NODE NO.":NP
80 N=NP*6
90 INPUT"HOW MANY NODAL PARAMETERS MUST BE MODIFIED":MP
100 FOR I=1 TO MP
110 PRINT I
120 INPUT ACTND(I)
130 NEXT I
140 INPUT "NO. OF GIVEN PARAMETERS":NGP
150 FOR I=1 TO NGP
160 PRINT I
170 INPUT ZERO1(I)
180 NEXT I
190 NN=N-NGP
200 OPEN "R",#5,"B:DISP".4
210 FIELD #5,4 AS E#
220 FOR I=1 TO NN
230 GET #5,I
240 U(I)=CVS(E#)
250 NEXT I
260 CLOSE #5
270 FOR I=1 TO NGP
280 ZERO=ZERO1(I)
290 FOR J=I TO N-1
300 IF J=ZERO THEN 320 ELSE 310
310 NEXT J
320 FOR JJ=J TO N
330 UU(JJ)=U(JJ)
340 NEXT JJ
350 U(ZERO)=0
360 FOR K=J+1 TO N
370 U(K)=UU(K-1)
380 NEXT K
390 NEXT I
400 FOR I=1 TO N
410 PRINT I,U(I)
420 NEXT I
430 FOR I=1 TO MP
440 Z=0
450 Z=ACTND(I)
460 IF INT(Z/6)=Z/6 THEN 490
470 IF INT((Z-2)/6)=(Z-2)/6 THEN 500
480 IF INT((Z+3)/6)=(Z+3)/6 THEN 510
490 U(Z-3)=U(Z)*(-M):GOTO 520
500 U(Z+1)=-U(Z):GOTO 520
510 U(Z+3)=U(Z)*(-M):GOTO 520

```

```

520 NEXT I
530 OPEN "R",#6,"B:RDISP",4
540 FIELD #6,4 AS F#
550 FOR I=1 TO N
560 LSET F#=MKS$(U(I))
570 PUT #6,I
580 NEXT I
590 CLOSE #6
600 LPRINT "SOLUTION VECTORS"
610 LPRINT "*****":LPRINT
620 Z=0
630 FOR I=1 TO N-5 STEP 6
640 Z=Z+1
650 LPRINT "U(";Z;")-----";U(I)
660 LPRINT "V(";Z;")-----";U(I+1)
670 LPRINT "DU/DX/(";Z;")-----";U(I+2)
680 LPRINT "DV/DX/(";Z;")-----";U(I+3)
690 LPRINT "DU/DY/(";Z;")-----";U(I+4)
700 LPRINT "DV/DY/(";Z;")-----";U(I+5)
710 NEXT I

```

```

10 REM          PROGRAM NAME : COMDISP
20 REM          *****
30 REM This program adds all known nodal parameters to the
   reduced nodal parameters.
40 DIM SYM(50,2),CBN(6),U(426)
50 INPUT "TOTAL NODE NO.:";NP
60 N=NP*6
70 INPUT"HOW MANY NODES ARE SYMMETRIC";NPSY
80 FOR I=1 TO NPSY
90 PRINT I
100 INPUT "ADDING NODE NO.:";SYM(I,1)
110 INPUT "ORIGINAL NODE NO.:";SYM(I,2)
120 NEXT I
130 NN=N-NPSY
140 OPEN "R",#5,"B:RDISP",4
150 FIELD #5,4 AS E#
160 FOR I=1 TO NN
170 GET #5,I
180 U(I)=CVS(E#)
190 NEXT I
200 CLOSE #5
210 CBN(1)=1:CBN(2)=-1:CBN(3)=-1:CBN(4)=1:CBN(5)=1:CBN(6)=-1
220 FOR I=1 TO NPSY
230 ND=(SYM(I,1)-1)*6
240 ND1=(SYM(I,2)-1)*6
250 FOR J=1 TO 6
260 PT=ND+J
270 PT1=ND1+J
280 U(PT)=U(PT1)*CBN(J)
290 NEXT J
300 NEXT I
310 OPEN "R",#6,"B:RRDISP",4
320 FIELD #6,4 AS F#
330 FOR I=1 TO N
340 LSET F#=MKS#(U(I))
350 PUT #6,I
360 NEXT I
370 CLOSE #6
380 LPRINT "SOLUTION VECTORS"
390 LPRINT "*****":LPRINT
400 Z=0
410 FOR I=1 TO N-5 STEP 6
420 Z=Z+1
430 LPRINT "U(";Z;")-----";U(I)
440 LPRINT "V(";Z;")-----";U(I+1)
450 LPRINT "DU/DX/(";Z;")-----";U(I+2)
460 LPRINT "DV/DX/(";Z;")-----";U(I+3)
470 LPRINT "DU/DY/(";Z;")-----";U(I+4)
480 LPRINT "DV/DY/(";Z;")-----";U(I+5)
490 NEXT I
500 END

```

```

10 REM
120 REM          PROGRAM NAME: STRESS
130 REM          *****
140 REM PROGRAM TO FIND THE UNIT STRAIN OF CONCRETE AND OF
    STEEL
150 DIM ND(71,4),NN(4),X(71),Y(71),B(3,24),FX(4),FY(4),PX(4),
    PY(4)
160 DIM XB(52,14),YB(52,14),XE(52,14),YE(52,14),BAR(52),U(426)
    ,UU(24)
170 DIM DST(3),EP(3),XST(20),XND(20),SIG(3),D(3,3)
180 REM x,y coordinate of each node
190 NM=71:REM total node numbers
200 NE=52:REM no of total elements
210 PI=3.1415927#
220 CS1=COS(PI/4)
230 SN1=SIN(PI/4)
240 EC=1236364!:M=.2:FCP=2720
250 ES=2E+07:ESEC=ES-EC
260 D1=EC/(1-M*M)
270 D(1,1)=D1:D(1,2)=D1*M
280 D(2,1)=D(1,2):D(2,2)=D1
290 D(3,3)=(1-M)*D1/2
300 FOR I=1 TO NM
310 READ X(I)
320 NEXT I
330 DATA 0, 0, 32.5, 32.5, 40.0, 40.0, 40.0, 40.0,
    40.0, 40.0
340 DATA 50.0, 50.0, 50.0, 50.0, 50.0, 50.0, 50.0
350 DATA 60.0, 60.0, 60.0, 60.0, 60.0, 60.0, 60.0
360 DATA 120.0,120.0,120.0,120.0,120.0,120.0,120.0
370 DATA 190.0,190.0,190.0,190.0,190.0,190.0,190.0
380 DATA 260.0,260.0,260.0,260.0,260.0,260.0,260.0
390 DATA 320.0,320.0,320.0,320.0,320.0,320.0,320.0
400 DATA 330.0,330.0,330.0,330.0,330.0,330.0,330.0
410 DATA 340.0,340.0,340.0,340.0,340.0,340.0,340.0
420 DATA 347.5,347.5,380.0,380.0
430 REM
440 FOR I=1 TO NM
450 READ Y(I)
460 NEXT I
470 DATA 0.0, 43.5,193.5,233.5
480 DATA 0.0, 43.5,53.5,63.5,118.5,193.5,233.5
490 DATA 0.0, 43.5,53.5,63.5,118.5,193.5,233.5
500 DATA 0.0, 43.5,53.5,63.5,118.5,193.5,233.5
510 DATA 0.0, 43.5,53.5,63.5,118.5,193.5,233.5
520 DATA 0.0, 43.5,53.5,63.5,118.5,193.5,233.5
530 DATA 0.0, 43.5,53.5,63.5,118.5,193.5,233.5
540 DATA 0.0, 43.5,53.5,63.5,118.5,193.5,233.5
550 DATA 0.0, 43.5,53.5,63.5,118.5,193.5,233.5
560 DATA 193.5,233.5, 0.0, 43.5
570 REM

```



```

490 REM NODE NO. CORRESPONDING TO EACH ELEMENT
500 FOR EL=1 TO NE
510 FOR EN=1 TO 4
520 READ ND(EL,EN)
530 NEXT EN:NEXT EL
540 DATA 1,5,6,2, 5,12,13,6, 12,19,20,13, 19,26,27,20,
26,33,34,27
550 DATA 33,40,41,34, 40,47,48,41, 47,54,55,48, 54,61,62,55,
61,70,71,62
560 DATA 6,13,14,7, 13,20,21,14, 20,27,28,21, 27,34,35,28,
34,41,42,35
570 DATA 41,48,49,42, 48,55,56,49, 55,62,63,56, 7,14,15, 8,
14,21,22,15
580 DATA 21,28,29,22, 28,35,36,29, 35,42,43,36, 42,49,50,43,
49,56,57,50
590 DATA 56,63,64,57, 8,15,16,9, 15,22,23,16, 22,29,30,23,
29,36,37,30
600 DATA 36,43,44,37, 43,50,51,44, 50,57,58,51, 57,64,65,58,
9,16,17,10
610 DATA 16,23,24,17, 23,30,31,24, 30,37,38,31, 37,44,45,38,
44,51,52,45
620 DATA 51,58,59,52, 58,65,66,59, 3,10,11,4, 10,17,18,11,
17,24,25,18
630 DATA 24,31,32,25, 31,38,39,32, 38,45,46,39, 45,52,53,46,
52,59,60,53
640 DATA 59,66,67,60, 66,68,69,67
650 REM bar coordinate
660 FOR I=1 TO NE
670 READ BAR(I)
680 NEXT I
690 DATA 0,6,6,10,10,10,10,6,6,0,3,3,6,7,7,6,3,3,3,6,7,7,6,
3,3,7,8,12,12,12,12
700 DATA 8,7,10,10,12,14,14,12,10,10,0,6,6,8,10,10,8,6,6,0
710 FOR I=1 TO NE
720 BN=BAR(I)
730 IF BN=0 THEN 770
740 FOR J=1 TO BN
750 READ XB(I,J)
760 NEXT J
770 NEXT I
780 DATA 43,43,43,43,43,43,50,50,50,50,57,57,60,60,70,90,110,
60,60,70,90,110,120,120,130,150
790 DATA 170,120,120,130,150,170,190,190,210,230,250,190,190,
210,230,250,260,260
800 DATA 270,290,310,260,260,270,290,310,320,320,320,320,323,
323,330,330,330,330,337,337,43,43,43,50,57,57
810 DATA 70,90,110,60,80,100,130,150,170,120,140,160,180,190,
210,230,250,200
820 DATA 220,240,270,290,310,260,280,300,320,323,323,330,337,
337,43,43,43,50,57,57,60,80,100,70,90,110,120
830 DATA 140,160,180,130,150,170,200,220,240,190,210,230,250,
260,280,300,270,290
840 DATA 310,320,323,323,330,337,337,43,43,43,43,43,43,43,50,

```

```

50,50,50,50,55,57,57,60,60,60,70,90,110,60,60
850 DATA 60,75,95,115,120,120,120,130,150,170,120,120,120,135
,155,175,190,190
860 DATA 190,210,230,250,190,190,195,215,235,255,260,260,260,
270,290,310,260,260
870 DATA 260,275,295,315,320,320,320,320,320,320,323,323,330,
330,330,330,330,337,337,43,43,43,45
880 DATA 43,43,43,43,43,43,50,50,50,50,50,50,50,57,57,60,6
0,60,65,85,105,60,60,60,80
890 DATA 100,120,120,120,125,145,165,185,120,120,120,120,140,
160,180,190,190,190
900 DATA 190,205,225,245,190,190,190,190,200,220,240,260,260,
260,265,285,305,260
910 DATA 260,260,260,280,300,320,320,320,325,320,320,320,320,
323,323,330,330,330,330,330,330
920 DATA 330,330,330,330,337,337,43,43,43,43,43,50,50,50,50,57
,57,60,60,80,100,60,63,83,103,120,120
930 DATA 140,160,180,120,123,143,163,183,190,190,200,220,240,
190,190,203,223,243
940 DATA 260,260,280,300,260,263,283,303,320,320,320,323,323,
323,330,330,330,330
950 DATA 337,337
960 FOR I=1 TO NE
970 BN=BAR(I)
980 IF BN=0 THEN 1020
990 FOR J=1 TO BN
1000 READ YB(I,J):YB(I,J)=YB(I,J)+.5
1010 NEXT J
1020 NEXT I
1030 DATA 36,16,10,30,2.5,2.5,23,3,23,43,2.5,2.5,33,13,3,3,3,
13,33,43,43,43,33,13,3,3,3,13,33,43,43
1040 DATA 43,23,3,3,3,3,23,43,43,43,33,13,3,3,3,13,33,43,4
3,43,33,13,13,33,2.5,2.5,23
1050 DATA 3,23,43,2.5,2.5,50,43,43,43,43,43,43,43,53,53,53
,43,43,43,53,53,53,53,43,43,43,43,53
1060 DATA 53,43,43,43,53,53,53,53,43,43,43,43,43,56,53,53,63,
53,53,53,53,63,63,63,53,53,53,53,63,63
1070 DATA 63,53,53,53,63,63,63,63,63,63,63,63,63,63,63,63,53,53,
63,53,53,96,76,70,90,110,63,63,103,83
1080 DATA 63,83,103,118,63,63,113,93,73,63,63,63,73,93,113,11
8,118,118,113,93,73,63,63
1090 DATA 63,73,93,113,118,118,118,103,83,63,63,63,63,83,100,
118,118,118,118,113
1100 DATA 93,73,63,63,63,73,93,113,118,118,118,118,113,93,73,73,9
3,113,63,63,103,83,63,83
1110 DATA 103,63,63,176,156,136,118,130,150,170,190,118,118,1
83,163,143,123,123,143,163,183,118,118
1120 DATA 173,153,133,118,118,118,118,133,153,173,193,193,193,173
,153,133,118,118
1130 DATA 118,118,133,153,173,193,193,193,193,183,163,143,123
,118,118,118,123
1140 DATA 143,163,183,193,193,193,173,153,133,118,118,118,133
,153,173,193,193

```

```

1150 DATA 193,173,153,133,118,133,153,173,193,118,118,183,163
,143,123,123,143,163,183,118,118
1160 DATA 216,196,210,230,193,193,223,203,203,223,193,193,213
,193,193,193,213,230,230,230,213
1170 DATA 193,193,193,193,213,230,230,230,230,223,203,193,193
,193,203,223,230
1180 DATA 230,230,213,193,193,193,213,230,230,230,213,193,213
,230,193,193,223,203,203
1190 DATA 223,193,193
1200 FOR I=1 TO NE
1210 BN=BAR(I)
1220 IF BN=0 THEN 1260
1230 FOR J=1 TO BN
1240 READ XE(I,J)
1250 NEXT J
1260 NEXT I
1270 DATA 50,50,50,50,43,43,60,60,60,60,57,57,70,90,110,120,1
20,70,90,110,120,120,130,150
1280 DATA 170,190,190,130,150,170,190,190,210,230,250,260,260
,210,230,250,260
1290 DATA 260,270,290,310,320,320,270,290,310,320,330,330
,330,330,323,323,337,337
1300 DATA 337,337,337,337,50,43,43,60,43,43,80,100,120,70,90,
110,140,160,180,130,150,170,190,200,220
1310 DATA 240,260,210,230,250,280,300,320,270,290,310,330,323
,323,337,337,337,50,43,43,60,57,57,70,90,110,80
1320 DATA 100,120,130,150,170,190,140,160,180,210,230,250,200
,220,240,260,270,290
1330 DATA 310,280,300,320,330,323,323,337,337,337,50,50,50,50
,50,43,43,60,60,60,60,60,60,57,57,65,85,105,120
1340 DATA 120,120,70,90,110,120,120,120,125,145,165,185,190,1
90,130,150,170,190
1350 DATA 190,190,205,225,245,260,260,260,210,230,250,260,260
,260,265,285,305
1360 DATA 320,320,320,270,290,310,320,320,320,325,330,330,330
,330,330,323,323,337,337
1370 DATA 337,337,337,337,337,50,50,50,50,50,50,50,43,43,6
0,60,60,60,55,60,60,60,57,57,80,100,120
1380 DATA 120,120,120,75,95,115,120,120,120,140,160,180,190,1
90,190,190,135,155
1390 DATA 175,190,190,190,190,200,220,240,260,260,260,195
,215,235,255,260
1400 DATA 260,260,280,300,320,320,320,320,275,295,315,320,320
,320,330,330,330
1410 DATA 330,330,330,330,330,323,323,337,337,337,337,335,337
,337,337,337,337,50,50,50,50,43,43,57,60
1420 DATA 60,60,57,57,77,97,117,120,80,100,120,120,137,157,17
7,190,190,140,160,180,190
1430 DATA 190,197,217,237,257,260,200,220,240,260,260,277,297
,317,320,280,300
1440 DATA 320,320,330,330,330,330,323,323,337,337,337,337,337
,337
1450 FOR I=1 TO NE

```

```

1460 BN=BAR(I)
1470 IF BN=0 THEN 1510
1480 FOR J=1 TO BN
1490 READ YE(I,J):YE(I,J)=YE(I,J)+.5
1500 NEXT J
1510 NEXT I
1520 DATA 43,23,3,23,43,43,33,13,13,33,43,43,43,43,33,13,3
,3,3,13,33,43,43,43,23,3,3,3
1530 DATA 3,23,43,43,43,33,13,3,3,13,33,43,43,43,33,13,3,3,
3,13,33,43,23,3,23,43,43
1540 DATA 30,10,16,36,43,43,43,53,53,53,53,53,53,43,43,
43,53,53,53,43,43,43,43,53,53,53
1550 DATA 43,43,43,53,53,53,43,43,43,43,53,53,50,53,53,63,63,
63,53,63,63,63,63,53,53,53,63,63,63,63
1560 DATA 53,53,53,63,63,63,53,53,53,53,63,63,63,53,53,63,
63,63,56,63,63,103,83,63,83,103,118,118
1570 DATA 113,93,73,73,93,113,118,118,118,118,118,113,93,73,6
3,63,63,73,93,113,118,118
1580 DATA 118,118,103,83,63,63,83,100,118,118,118,113,9
3,73,63,63,63,73,93
1590 DATA 113,118,118,118,113,93,73,63,63,63,73,93,113,118,10
3,83,63,83,103,118,118,110
1600 DATA 90,70,76,96,118,118,183,163,143,123,123,143,163,183
,193,193,193,173,153,133,118,133
1610 DATA 153,173,193,193,193,193,193,173,153,133,118,118,118
,133,153,173,193,193,193
1620 DATA 183,163,143,123,118,118,118,123,143,163,183,193,193
,193,193,173,153
1630 DATA 133,118,118,118,118,133,153,173,193,193,193,173,153
,133,118,118,118
1640 DATA 133,153,173,183,163,143,123,123,143,163,183,193,193
,190,170,150,130,118,136
1650 DATA 156,176,193,193,223,203,203,223,230,230,230,213,193
,213,230,230,230,230,230,213,193,193,193
1660 DATA 213,230,230,230,223,203,193,193,193,203,223,230,230
,230,230,213,193
1670 DATA 193,193,193,213,230,230,230,213,193,193,193,213,223
,203,203,223,230,230,230
1680 DATA 210,196,216,230,230
1690 FOR I=1 TO NE
1700 BN=BAR(I)
1710 FOR J=1 TO BN
1720 LPRINT I,J,XB(I,J),YB(I,J),XE(I,J),YE(I,J)
1730 NEXT J
1740 LPRINT
1750 NEXT I
1760 OPEN "R",#5,"B:RRDISP",4
1770 FIELD #5,4 AS D$
1780 FOR I=1 TO 6*NM
1790 GET #5,I
1800 U(I)=CVS(D$)
1810 NEXT I
1820 CLOSE #5

```

```

1830 DX=5:DY=5
1840 FOR CASE=1 TO 8
1850 MAX=-1000:MIN=1000
1860 FOR EL=27 TO 42
1870 FOR K=1 TO 4
1880 NN(K)=0
1890 NEXT K
1900 FOR K=1 TO 4
1910 NN(K)*ND(EL,K):'PRINT NN(K)
1920 NEXT K
1930 T=0
1940 FOR II=1 TO 4
1950 FOR JJ=1 TO 6
1960 T=T+1
1970 IJ=(NN(II)-1)*6+JJ
1980 UU(T)=U(IJ)
1990 NEXT JJ,II
2000 XN=(X(NN(2))-X(NN(1)))/5+1
2010 YN=(Y(NN(4))-Y(NN(1)))/5+1
2020 FOR I=1 TO YN
2030 XST(I)=X(NN(1))
2040 XND(I)=X(NN(2))
2050 NEXT I
2060 IF EL>27 THEN 2080
2070 CLS
2080 FOR I=1 TO 4
2090 J=I-1:IF J=5 THEN J=1
2100 XP=5+X(NN(I))+Y(NN(I))*CS1
2110 YP=10+Y(NN(I))*SN1
2120 XQ=5+X(NN(J))+Y(NN(J))*CS1
2130 YQ=10+Y(NN(J))*SN1
2140 LINE (XP*1.3-55,180-YP*.65)-(XQ*1.3-55,180-YQ*.65)
2150 NEXT I
2160 IF CASE>=7 THEN 2460
2170 FOR I=1 TO 4
2180 X1=X(NN(I)):Y1=Y(NN(I))
2190 I2=I+1
2200 IF I2=5 THEN I2=1
2210 X2=X(NN(I2)):Y2=Y(NN(I2))
2220 DPX=(X2-X1)/(XN-1):DPY=(Y2-Y1)/(YN-1)
2230 IF DPX=0 THEN ED=YN ELSE ED=XN
2240 FOR J=1 TO ED
2250 X=X1+(J-1)*DPX
2260 Y=Y1+(J-1)*DPY
2270 GOSUB 3200
2280 NEXT J
2290 NEXT I
2300 FOR IK=1 TO YN
2310 XO=XST(IK)
2320 XEND=XND(IK)
2330 DDX=(XEND-XO)/(XN-1)
2340 Y=Y(NN(1))+5*(IK-1)
2350 FOR J=1 TO XN

```

```

2360 X=XO+(J-1)*DDX
2370 GOSUB 3200
2380 IF MAX<MGT THEN 2390 ELSE 2400
2390 MAX=MGT:XX=X:YY=Y
2400 IF MIN>MGT THEN 2410 ELSE 2420
2410 MIN=MGT:XX=X:YY=Y
2420 NEXT J
2430 NEXT IK
2440 IF CASE < 7 THEN 2820
2450 REM calculate steel part
2460 BN=BAR(EL)
2470 FOR EL1=1 TO BN
2480 XB1=XB(EL,EL1):YB1=YB(EL,EL1)
2490 XE1=XE(EL,EL1):YE1=YE(EL,EL1)
2500 LENG=SQR((XE1-XB1)^2+(YE1-YB1)^2)
2510 SN=(YE1-YB1)/LENG
2520 CS=(XE1-XB1)/LENG
2530 IF CASE=7 THEN 2550
2540 IF CASE=8 THEN 2560
2550 IF SN=0 OR (YE1-YB1)<0 THEN 2570 ELSE 2810
2560 IF SN>0 OR (YE1-YB1)>0 THEN 2570 ELSE 2810
2570 DST(1)=CS^2
2580 DST(2)=SN^2
2590 DST(3)=SN*CS
2600 DX=(XE1-XB1)/20
2610 DY=(YE1-YB1)/20
2620 FOR J=1 TO 21
2630 X=XB1+DX*(J-1)
2640 V=X-X(NN(1))
2650 L=X(NN(2))-X(NN(1))
2660 GOSUB 3100
2670 FX(1)=F1:FX(2)=F2:FX(3)=F3:FX(4)=F4
2680 PX(1)=F5:PX(2)=F6:PX(3)=F7:PX(4)=F8
2690 Y=YB1+DY*(J-1)
2700 V=Y-Y(NN(1))
2710 L=Y(NN(4))-Y(NN(1))
2720 GOSUB 3100
2730 FY(1)=F1:FY(2)=F2:FY(3)=F3:FY(4)=F4
2740 PY(1)=F5:PY(2)=F6:PY(3)=F7:PY(4)=F8
2750 GOSUB 3340
2760 IF MAX<ET THEN 2770 ELSE 2780
2770 MAX=ET:XX=X:YY=Y
2780 IF MIN>ET THEN 2790 ELSE 2800
2790 MIN=ET:XX=X:YY=Y
2800 NEXT J
2810 NEXT EL1
2820 NEXT EL
2830 IF CASE=1 THEN 2910
2840 IF CASE=2 THEN 2930
2850 IF CASE=3 THEN 2950
2860 IF CASE=4 THEN 2970
2870 IF CASE=5 THEN 2990
2880 IF CASE=6 THEN 3010

```

```

2890 IF CASE=7 THEN 3030
2900 IF CASE=8 THEN 3050
2910 LOCATE 23,5:PRINT "MAXIMUM PRINCIPAL STRESS "
2920 GOTO 3060
2930 LOCATE 23,5:PRINT "MINIMUM PRINCIPAL STRESS "
2940 GOTO 3060
2950 LOCATE 23,5:PRINT "F FOR CRACKING(OR CRUSHING) OF
CONCRETE"
2960 GOTO 3060
2970 LOCATE 23,5:PRINT "NORMAL STRESS IN X-DIRECTION"
2980 GOTO 3060
2990 LOCATE 23,5:PRINT "NORMAL STRESS IN Y-DIRECTION"
3000 GOTO 3060
3010 LOCATE 23,5:PRINT "SHEARING STRESS WITH RESPECT TO X,Y
AXES"
3020 GOTO 3060
3030 LOCATE 23,5:PRINT "STRESS OF STEEL BAR(horizontal or
greater than 90 deg)"
3040 GOTO 3060
3050 LOCATE 23,5:PRINT "STRESS OF STEEL BAR(vertical or less
than 90 deg)"
3060 LOCATE 24,9:PRINT "MAX";MAX;"x coor.";XMX;"y coor.";YMX;
"MIN";MIN;"x coor.";XMN;"y coor.";YMN
3070 AS=INKEY$:IF AS="" THEN 3070
3080 NEXT CASE
3090 END
3100 LV=L-V
3110 F1=LV/L+V*LV*(L-2*V)/L^3
3120 F2=V/L+V*LV*(2+V-L)/L^3
3130 F3=V*LV^2/L^2
3140 F4=-LV*V^2/L^2
3150 F5=(-6*V*L+6*V^2)/L^3
3160 F6=(6*V*L-6*V^2)/L^3
3170 F7=(L^2-4*L*V+3*V^2)/L^2
3180 F8=(-2*L*V+3*V^2)/L^2
3190 RETURN
3200 V1=X-X(NN(1)):L1=X(NN(2))-X(NN(1))
3210 V2=Y-Y(NN(1)):L2=Y(NN(4))-Y(NN(1))
3220 FOR II=1 TO 2
3230 V=V1:L=L1
3240 IF II=2 THEN V=V2
3250 IF II=2 THEN L=L2
3260 GOSUB 3100
3270 IF II=2 THEN 3310
3280 FX(1)=F1:FX(2)=F2:FX(3)=F3:FX(4)=F4
3290 PX(1)=F5:PX(2)=F6:PX(3)=F7:PX(4)=F8
3300 GOTO 3330
3310 FY(1)=F1:FY(2)=F2:FY(3)=F3:FY(4)=F4
3320 PY(1)=F5:PY(2)=F6:PY(3)=F7:PY(4)=F8
3330 NEXT II
3340 FOR II=1 TO 4
3350 C=(II-1)*6
3360 IF II=1 OR II=4 THEN K=1 ELSE K=2

```

```

3370 IF II <=2 THEN KK=1 ELSE KK=2
3380 B(1,1+C)=PX(K)*FY(KK):B(2,2+C)=FX(K)*PY(KK)
3390 B(1,3+C)=PX(K+2)*FY(KK):B(2,4+C)=FX(K+2)*PY(KK)
3400 B(1,5+C)=PX(K)*FY(KK+2):B(2,6+C)=FX(K)*PY(KK+2)
3410 B(3,1+C)=B(2,2+C):B(3,2+C)=B(1,1+C)
3420 B(3,3+C)=B(2,4+C):B(3,4+C)=B(1,3+C)
3430 B(3,5+C)=B(2,6+C):B(3,6+C)=B(1,5+C)
3440 NEXT II
3450 FOR I1=1 TO 3
3460 EP(I1)=0
3470 FOR J1=1 TO 24
3480 EP(I1)=EP(I1)+B(I1,J1)*UU(J1)
3490 NEXT J1
3500 NEXT I1
3510 FOR II=1 TO 3
3520 SIG(II)=0
3530 FOR KK=1 TO 3
3540 SIG(II)=SIG(II)+D(II,KK)*EP(KK)
3550 NEXT KK
3560 NEXT II
3570 IF CASE<7 THEN 3650
3580 ET=0
3590 FOR J1=1 TO 3
3600 ET=DST(J1)*EP(J1)+ET
3610 NEXT J1
3620 ET=ET+ESEC
3630 Z=ET:SC=.03
3640 GOTO 3800
3650 IF CASE<4 THEN 3660 ELSE 3780
3660 HE1=(SIG(1)+SIG(2))/2
3670 HE2=((SIG(1)-SIG(2))/2)^2
3680 HE3=(HE2+SIG(3)^2)^.5
3690 IF CASE=3 THEN 3730 ELSE 3700
3700 IF CASE=1 THEN 3710 ELSE 3720
3710 MGT=HE1+HE3:Z=MGT:SC=.3:GOTO 3800
3720 MGT=HE1-HE3:Z=MGT:SC=.3:GOTO 3800
3730 S1=SIG(1):S2=SIG(2):MGT=0
3740 MGT=.23*((S1-S2)^2+S2^2+S1^2)/(FCP^2/10-.9*FCP*(S1+S2))
3750 MGT=MGT*.54*(S1-.17*S2)+1.1*FCP/(FCP*(FCP+S1+S2)/10+.17*
    FCP*(FCP/10-S1-S2))
3760 Z=MGT:SC=50
3770 GOTO 3800
3780 MGT=SIG(CASE-3)
3790 Z=MGT:SC=.3
3800 XP=5+X+Y*CS1
3810 YP=10+Y*SN1+SC*Z
3820 IF J>1 THEN 3850
3830 LINE (1.3*XP-55,180-YP*.65)-(1.3*XP-55,180-YP*.65)
3840 GOTO 3860
3850 LINE -(1.3*XP-55,180-YP*.65)
3860 RETURN

```



```

10 REM
20 REM          PROGRAM NAME : DIPLACE
30 REM          *****
40 REM This program gives the graph of displacement.
50 DIM X(71),Y(71),ND(52,4),NN(4),U(426),UU(24)
60 REM x,y coordinate of each node
70 NM=71:REM total node numbers
80 NE=52:REM no of total elements
90 FOR I=1 TO NM
100 READ X(I)
110 NEXT I
120 DATA 0, 0,32.5,32.5,40.0,40.0,40.0,40.0, 40.0, 40.0, 40.0
130 DATA 50.0, 50.0, 50.0, 50.0, 50.0, 50.0, 50.0
140 DATA 60.0, 60.0, 60.0, 60.0, 60.0, 60.0, 60.0
150 DATA 120.0,120.0,120.0,120.0,120.0,120.0,120.0,120.0
160 DATA 190.0,190.0,190.0,190.0,190.0,190.0,190.0
170 DATA 260.0,260.0,260.0,260.0,260.0,260.0,260.0
180 DATA 320.0,320.0,320.0,320.0,320.0,320.0,320.0
190 DATA 330.0,330.0,330.0,330.0,330.0,330.0,330.0
200 DATA 340.0,340.0,340.0,340.0,340.0,340.0,340.0
210 DATA 347.5,347.5,380.0,380.0
220 REM
230 FOR I=1 TO NM
240 READ Y(I)
250 NEXT I
260 DATA 0.0, 43.5,193.5,233.5
270 DATA 0.0, 43.5, 53.5, 63.5,118.5,193.5,233.5
280 DATA 0.0, 43.5, 53.5, 63.5,118.5,193.5,233.5
290 DATA 0.0, 43.5, 53.5, 63.5,118.5,193.5,233.5
300 DATA 0.0, 43.5, 53.5, 63.5,118.5,193.5,233.5
310 DATA 0.0, 43.5, 53.5, 63.5,118.5,193.5,233.5
320 DATA 0.0, 43.5, 53.5, 63.5,118.5,193.5,233.5
330 DATA 0.0, 43.5, 53.5, 63.5,118.5,193.5,233.5
340 DATA 0.0, 43.5, 53.5, 63.5,118.5,193.5,233.5
350 DATA 0.0, 43.5, 53.5, 63.5,118.5,193.5,233.5
360 DATA 193.5,233.5, 0.0, 43.5
370 REM
380 REM NODE NO. CORRESPONDING TO EACH ELEMENT
390 FOR EL=1 TO NE
400 FOR EN=1 TO 4
410 READ ND(EL,EN)
420 NEXT EN:NEXT EL
430 DATA 1, 5, 6, 2, 5,12,13, 6, 12,19,20,13, 19,26,27,20,
26,33,34,27
440 DATA 33,40,41,34, 40,47,48,41, 47,54,55,48, 54,61,62,55,
61,70,71,62
450 DATA 6,13,14, 7, 13,20,21,14, 20,27,28,21, 27,34,35,28,
34,41,42,35
460 DATA 41,48,49,42, 48,55,56,49, 55,62,63,56, 7,14,15, 8,
14,21,22,15
470 DATA 21,28,29,22, 28,35,36,29, 35,42,43,36, 42,49,50,43,
49,56,57,50

```

```

480 DATA 56,63,64,57, 8,15,16, 9, 15,22,23,16, 22,29,30,23,
29,36,37,30
490 DATA 36,43,44,37, 43,50,51,44, 50,57,58,51, 57,64,65,58,
9,16,17,10
500 DATA 16,23,24,17, 23,30,31,24, 30,37,38,31, 37,44,45,38,
44,51,52,45
510 DATA 51,58,59,52, 58,65,66,59, 3,10,11, 4, 10,17,18,11,
17,24,25,18
520 DATA 24,31,32,25, 31,38,39,32, 38,45,46,39, 45,52,53,46,
52,59,60,53
530 DATA 59,66,67,60, 66,68,69,67
540 OPEN "R",#5,"B:RRDISP",4
550 FIELD #5,4 AS D#
560 FOR I=1 TO 426
570 GET #5,I
580 U(I)=CVS(D#)
590 NEXT I
600 CLOSE #5
610 REM
620 FOR EL=1 TO 52
630 FOR K=1 TO 4
640 NN(K)=ND(EL,K)
650 NEXT K
660 FOR II=1 TO 4
670 FOR JJ=1 TO 6
680 T=T+1
690 IJ=(NN(II)-1)*6+JJ
700 UU(T)=U(IJ)
710 NEXT JJ,II
720 IF EL>1 THEN 740
730 CLS
740 FOR I=1 TO 4
750 J=I+1:IF J=5 THEN J=1
760 XP=20+X(NN(I)):YP=Y(NN(I))
770 XQ=20+X(NN(J)):YQ=Y(NN(J))
780 LINE (XP=1.3,200-YP)-(XQ=1.3,200-YQ)
790 NEXT I
800 FOR I=1 TO 4
810 J=I+1:IF J=5 THEN J=1
820 XP=20+X(NN(I))+UU((I-1)*6+1)*50:YP=Y(NN(I))+UU((I-1)*6+2)
*50
830 XQ=20+X(NN(J))+UU((J-1)*6+1)*50:YQ=Y(NN(J))+UU((J-1)*6+2)
*50
840 LINE (XP=1.3,200-YP)-(XQ=1.3,200-YQ)
850 LOCATE 24,22:PRINT "DISPLACEMENT SHAPE"
860 A$=INKEY$:IF A$="" THEN 860
870 END

```

```

10 REM
20 REM          PROGRAM NAME: STRAIN
30 REM          *****
40 REM PROGRAM TO FIND THE UNIT STRAIN OF CONCRETE AND OF
STEEL
50 DIM ND(71,4),NN(4),X(71),Y(71),B(3,24),FX(4),FY(4),PX(4),
PY(4)
60 DIM XB(52,14),YB(52,14),XE(52,14),YE(52,14),BAR(52),U(426),
UU(24)
70 DIM DST(3),EP(3),XST(20),XND(20)
80 REM x,y coordinate of each node
90 NM=71:REM total node numbers
100 NE=52:REM no of total elements
110 PI=3.1415927#
120 CS1=COS(PI/4)
130 SN1=SIN(PI/4)
140 FOR I=1 TO NM
150 READ X(I)
160 NEXT I
170 DATA 0.0,32.5,32.5,40.0,40.0,40.0,40.0,40.0,40.0,40.0
180 DATA 50.0, 50.0, 50.0, 50.0, 50.0, 50.0, 50.0, 50.0, 50.0
190 DATA 60.0, 60.0, 60.0, 60.0, 60.0, 60.0, 60.0, 60.0
200 DATA 120.0,120.0,120.0,120.0,120.0,120.0,120.0,120.0
210 DATA 190.0,190.0,190.0,190.0,190.0,190.0,190.0,190.0
220 DATA 260.0,260.0,260.0,260.0,260.0,260.0,260.0,260.0
230 DATA 320.0,320.0,320.0,320.0,320.0,320.0,320.0,320.0
240 DATA 330.0,330.0,330.0,330.0,330.0,330.0,330.0,330.0
250 DATA 340.0,340.0,340.0,340.0,340.0,340.0,340.0,340.0
260 DATA 347.5,347.5,380.0,380.0
270 REM
280 FOR I=1 TO NM
290 READ Y(I)
300 NEXT I
310 DATA 0.0, 43.5,193.5,233.5
320 DATA 0.0, 43.5,53.5,63.5,118.5,193.5,233.5
330 DATA 0.0, 43.5,53.5,63.5,118.5,193.5,233.5
340 DATA 0.0, 43.5,53.5,63.5,118.5,193.5,233.5
350 DATA 0.0, 43.5,53.5,63.5,118.5,193.5,233.5
360 DATA 0.0, 43.5,53.5,63.5,118.5,193.5,233.5
370 DATA 0.0, 43.5,53.5,63.5,118.5,193.5,233.5
380 DATA 0.0, 43.5,53.5,63.5,118.5,193.5,233.5
390 DATA 0.0, 43.5,53.5,63.5,118.5,193.5,233.5
400 DATA 0.0, 43.5,53.5,63.5,118.5,193.5,233.5
410 DATA 193.5,233.5, 0.0, 43.5
420 REM
430 REM NODE NO. CORRESPONDING TO EACH ELEMENT
440 FOR EL=1 TO NE
450 FOR EN=1 TO 4
460 READ ND(EL,EN)
470 NEXT EN:NEXT EL
480 DATA 1, 5, 6, 2, 5,12,13, 6, 12,19,20,13, 19,26,27,20,
26,33,34,27

```

```

490 DATA 33,40,41,34, 40,47,48,41, 47,54,55,48, 54,61,62,55,
    61,70,71,62
500 DATA 6,13,14, 7, 13,20,21,14, 20,27,28,21, 27,34,35,28,
    34,41,42,35
510 DATA 41,48,49,42, 48,55,56,49, 55,62,63,56, 7,14,15, 8,
    14,21,22,15
520 DATA 21,28,29,22, 28,35,36,29, 35,42,43,36, 42,49,50,43,
    49,56,57,50
530 DATA 56,63,64,57, 8,15,16, 9, 15,22,23,16, 22,29,30,23,
    29,36,37,30
540 DATA 36,43,44,37, 43,50,51,44, 50,57,58,51, 57,64,65,58,
    9,16,17,10
550 DATA 16,23,24,17, 23,30,31,24, 30,37,38,31, 37,44,45,38,
    44,51,52,45
560 DATA 51,58,59,52, 58,65,66,59, 3,10,11, 4, 10,17,18,11,
    17,24,25,18
570 DATA 24,31,32,25, 31,38,39,32, 38,45,46,39, 45,52,53,46,
    52,59,60,53
580 DATA 59,66,67,60, 66,68,69,67
590 REM bar coordinate
600 FOR I=1 TO NE
610 READ BAR(I)
620 NEXT I
630 DATA 0,6,6,10,10,10,10,6,6,0,3,3,6,7,7,6,3,3,3,3,6,7,7,6,
    3,3,7,8,12,12,12,12
640 DATA 8,7,10,10,12,14,14,12,10,10,0,6,6,8,10,10,8,6,6,0
650 FOR I=1 TO NE
660 BN=BAR(I)
670 IF BN=0 THEN 710
680 FOR J=1 TO BN
690 READ XB(I,J)
700 NEXT J
710 NEXT I
720 DATA 43,43,43,43,43,50,50,50,50,57,57,60,60,70,90,110,
    60,60,70,90,110,120,120,130,150
730 DATA 170,120,120,130,150,170,190,190,210,230,250,190,190,
    210,230,250,260,260
740 DATA 270,290,310,260,260,270,290,310,320,320,320,320,323,
    323,330,330,330,330,337,337,43,43,43,50,57,57
750 DATA 70,90,110,60,80,100,130,150,170,120,140,160,180,190,
    210,230,250,200
760 DATA 220,240,270,290,310,260,280,300,320,323,323,330,337,
    337,43,43,43,50,57,57,60,80,100,70,90,110,120
770 DATA 140,160,180,130,150,170,200,220,240,190,210,230,250,
    260,280,300,270,290
780 DATA 310,320,323,323,330,337,337,43,43,43,43,43,43,50,
    50,50,50,55,57,57,60,60,70,90,110,60,60
790 DATA 60,75,95,115,120,120,120,130,150,170,120,120,120,135,
    155,175,190,190
800 DATA 190,210,230,250,190,190,195,215,235,255,260,260,260,
    270,290,310,260,260
810 DATA 260,275,295,315,320,320,320,320,320,320,323,323,330,
    330,330,330,330,337,337,43,43,43,45

```

```

820 DATA 43,43,43,43,43,43,50,50,50,50,50,50,50,50,57,57,60,6
0,60,65,85,105,60,60,60,60,80
830 DATA 100,120,120,120,120,125,145,165,185,120,120,120,120,140,
160,180,190,190,190
840 DATA 190,205,225,245,190,190,190,190,200,220,240,260,260,
260,265,285,305,260
850 DATA 260,260,260,280,300,320,320,320,325,320,320,320,320,
323,323,330,330,330,330,330
860 DATA 330,330,330,337,337,43,43,43,43,43,50,50,50,50,57
,57,60,60,80,100,60,63,83,103,120,120
870 DATA 140,160,180,120,123,143,163,183,190,190,200,220,240,
190,190,203,223,243
880 DATA 260,260,280,300,260,263,283,303,320,320,320,323,323,
323,330,330,330,330
890 DATA 337,337
900 FOR I=1 TO NE
910 BN=BAR(I)
920 IF BN=0 THEN 960
930 FOR J=1 TO BN
940 READ YB(I,J):YB(I,J)=YB(I,J)+.5
950 NEXT J
960 NEXT I
970 DATA 36,16,10,30,2.5,2.5,23,3,23,43,2.5,2.5,33,13,3,3,3,13
,33,43,43,43,33,13,3,3,13,33,43,43
980 DATA 43,23,3,3,3,3,23,43,43,43,33,13,3,3,3,13,33,43,43,
43,33,13,13,33,2.5,2.5,23
990 DATA 3,23,43,2.5,2.5,50,43,43,43,43,43,43,43,43,53,53,53,4
3,43,43,53,53,53,43,43,43,43,43,53,53
1000 DATA 53,43,43,43,53,53,53,53,43,43,43,43,56,53,53,63,5
3,53,53,53,53,63,63,63,53,53,53,53,63,63
1010 DATA 63,53,53,53,63,63,63,63,53,53,63,63,63,53,53,6
3,53,53,96,76,70,90,110,63,63,103,83
1020 DATA 63,83,103,118,63,63,113,93,73,63,63,63,73,93,113,118
,118,118,113,93,73,63,63
1030 DATA 63,73,93,113,118,118,118,103,83,63,63,63,63,83,100,1
18,118,118,118,113
1040 DATA 93,73,63,63,63,73,93,113,118,118,118,113,93,73,73,93
,113,63,63,103,83,63,83
1050 DATA 103,63,63,176,156,136,118,130,150,170,190,118,118,18
3,163,143,123,123,143,163,183,118,118
1060 DATA 173,153,133,118,118,118,133,153,173,193,193,193,173,
153,133,118,118
1070 DATA 118,118,133,153,173,193,193,193,193,183,163,143,123,
118,118,118,123
1080 DATA 143,163,183,193,193,193,173,153,133,118,118,118,133,
153,173,193,193
1090 DATA 193,173,153,133,118,133,153,173,193,118,118,183,163,
143,123,123,143,163,183,118,118
1100 DATA 216,196,210,230,193,193,223,203,203,223,193,193,213,
193,193,193,213,230,230,230,213
1110 DATA 193,193,193,193,213,230,230,230,230,223,203,193,193,
193,203,223,230
1120 DATA 230,230,213,193,193,193,213,230,230,230,213,193,213,

```

```

230,193,193,223,203,203
1130 DATA 223,193,193
1140 FOR I=1 TO NE
1150 8N=BAR(I)
1160 IF 8N=0 THEN 1200
1170 FOR J=1 TO 8N
1180 READ XE(I,J)
1190 NEXT J
1200 NEXT I
1210 DATA 50,50,50,50,43,43,60,60,60,60,57,57,70,90,110,120,
120,70,90,110,120,120,130,150
1220 DATA 170,190,190,130,150,170,190,190,210,230,250,260,26
0,210,230,250,260
1230 DATA 260,270,290,310,320,320,270,290,310,320,320,330,33
0,330,330,323,323,337,337
1240 DATA 337,337,337,337,50,43,43,60,43,43,80,100,120,70,90,
110,140,160,180,130,150,170,190,200,220
1250 DATA 240,260,210,230,250,280,300,320,270,290,310,330,323
,323,337,337,337,50,43,43,60,57,57,70,90,110,80
1260 DATA 100,120,130,150,170,190,140,160,180,210,230,250,200
,220,240,260,270,290
1270 DATA 310,280,300,320,330,323,323,337,337,337,50,50,50,50
,50,43,43,60,60,60,60,60,60,57,57,65,85,105,120
1280 DATA 120,120,70,90,110,120,120,120,125,145,165,185,190,1
90,130,150,170,190
1290 DATA 190,190,205,225,245,260,260,260,210,230,250,260,260,
260,265,285,305
1300 DATA 320,320,320,270,290,310,320,320,320,325,330,330,330,
330,330,323,323,337,337
1310 DATA 337,337,337,337,337,50,50,50,50,50,50,50,50,43,43,6
0,60,60,60,55,60,60,60,57,57,80,100,120
1320 DATA 120,120,120,75,95,115,120,120,120,140,160,180,190,1
90,190,190,135,155
1330 DATA 175,190,190,190,190,200,220,240,260,260,260,195
,215,235,255,260
1340 DATA 260,260,280,300,320,320,320,320,275,295,315,320,320
,320,330,330,330
1350 DATA 330,330,330,330,330,323,323,337,337,337,337,335,337
,337,337,337,337,50,50,50,50,43,43,57,60
1360 DATA 60,60,57,57,77,97,117,120,80,100,120,120,137,157,17
7,190,190,140,160,180,190
1370 DATA 190,197,217,237,257,260,200,220,240,260,260,277,297
,317,320,280,300
1380 DATA 320,320,330,330,330,330,323,323,337,337,337,337,337
,337
1390 FOR I=1 TO NE
1400 8N=BAR(I)
1410 IF 8N=0 THEN 1450
1420 FOR J=1 TO 8N
1430 READ YE(I,J):YE(I,J)=YE(I,J)+.5
1440 NEXT J
1450 NEXT I
1460 DATA 43,23,3,23,43,43,33,13,13,33,43,43,43,43,33,13,3

```

```

      ,3,3,13,33,43,43,43,43,23,3,3,3
1470 DATA 3,23,43,43,43,33,13,3,3,13,33,43,43,43,33,13,3,3,
      3,13,33,43,23,3,23,43,43
1480 DATA 30,10,16,36,43,43,43,53,53,53,53,53,53,53,43,43,
      43,53,53,53,43,43,43,43,53,53,53,53
1490 DATA 43,43,43,53,53,53,43,43,43,43,53,53,50,53,53,63,63,
      63,53,63,63,63,63,53,53,53,63,63,63,63
1500 DATA 53,53,53,63,63,63,53,53,53,63,63,63,53,53,63,
      63,63,56,63,63,103,83,63,83,103,118,118
1510 DATA 113,93,73,73,93,113,118,118,118,118,118,113,93,73,6
      3,63,63,73,93,113,118,118
1520 DATA 118,118,103,83,63,63,63,83,100,118,118,118,113,9
      3,73,63,63,63,73,93
1530 DATA 113,118,118,118,113,93,73,63,63,63,73,93,113,118,10
      3,83,63,83,103,118,118,110
1540 DATA 90,70,76,96,118,118,183,163,143,123,123,143,163,183
      ,193,193,193,173,153,133,118,133
1550 DATA 153,173,193,193,193,193,193,173,153,133,118,118,118
      ,133,153,173,193,193
1560 DATA 183,163,143,123,118,118,118,123,143,163,183,193,193
      ,193,193,173,153
1570 DATA 133,118,118,118,118,118,133,153,173,193,193,193,173,153
      ,133,118,118,118
1580 DATA 133,153,173,183,163,143,123,123,143,163,183,193,193
      ,190,170,150,130,118,136
1590 DATA 156,176,193,193,223,203,203,223,230,230,230,213,193
      ,213,230,230,230,230,230,213,193,193,193
1600 DATA 213,230,230,230,223,203,193,193,193,203,223,230,230
      ,230,230,213,193
1610 DATA 193,193,193,213,230,230,230,213,193,193,193,213,223
      ,203,203,223,230,230,230
1620 DATA 210,196,216,230,230
1630 FOR I=1 TO NE
1640 BN=BAR(I)
1650 FOR J=1 TO BN
1660 LPRINT I,J,XB(I,J),YB(I,J),XE(I,J),YE(I,J)
1670 NEXT J
1680 LPRINT
1690 NEXT I
1700 OPEN "R",#5,"B:RRDISP",4
1710 FIELD #5,4 AS D#
1720 FOR I=1 TO 6*NM
1730 GET #5,I
1740 U(I)=CVS(D#)
1750 NEXT I
1760 CLOSE #5
1770 DX=5:DY=5
1780 FOR CASE=1 TO 7
1790 MAX=-1000:MIN=1000
1800 FOR EL=27 TO 42
1810 FOR K=1 TO 4
1820 NN(K)=0
1830 NEXT K

```

```

1840 FOR K=1 TO 4
1850 NN(K)=ND(EL,K):PRINT NN(K)
1860 NEXT K
1870 T=0
1880 FOR II=1 TO 4
1890 FOR JJ=1 TO 6
1900 T=T+1
1910 IJ=(NN(II)-1)*6+JJ
1920 UU(T)=U(IJ)
1930 NEXT JJ,II
1940 XN=(X(NN(2))-X(NN(1)))/5+1
1950 YN=(Y(NN(4))-Y(NN(1)))/5+1
1960 FOR I=1 TO YN
1970 XST(I)=X(NN(1))
1980 XND(I)=X(NN(2))
1990 NEXT I
2000 IF EL>27 THEN 2020
2010 CLS
2020 FOR I=1 TO 4
2030 J=I+1:IF J=5 THEN J=1
2040 XP=5+X(NN(I))+Y(NN(I))*CS1
2050 YP=10+Y(NN(I))*SN1
2060 XQ=5+X(NN(J))+Y(NN(J))*CS1
2070 YQ=10+Y(NN(J))*SN1
2080 LINE (XP=1.3-55,180-YP*.65)-(XQ=1.3-55,180-YQ*.65)
2090 NEXT I
2100 IF CASE>=6 THEN 2400
2110 FOR I=1 TO 4
2120 X1=X(NN(I)):Y1=Y(NN(I))
2130 I2=I+1
2140 IF I2=5 THEN I2=1
2150 X2=X(NN(I2)):Y2=Y(NN(I2))
2160 DPX=(X2-X1)/(XN-1):DPY=(Y2-Y1)/(YN-1)
2170 IF DPX=0 THEN ED=YN ELSE ED=XN
2180 FOR J=1 TO ED
2190 X=X1+(J-1)*DPX
2200 Y=Y1+(J-1)*DPY
2210 GOSUB 3110
2220 NEXT J
2230 NEXT I
2240 FOR IK=1 TO YN
2250 XO=XST(IK)
2260 XEND=XND(IK)
2270 DDX=(XEND-XO)/(XN-1)
2280 Y=Y(NN(1))+5*(IK-1)
2290 FOR J=1 TO XN
2300 X=XO+(J-1)*DDX
2310 GOSUB 3110
2320 IF MAX<MGT THEN 2330 ELSE 2340
2330 MAX=MGT:XXM=X:YMX=Y
2340 IF MIN>MGT THEN 2350 ELSE 2360
2350 MIN=MGT:XMN=X:YMN=Y
2360 NEXT J

```



```

2370 NEXT IK
2380 IF CASE <6 THEN 2760
2390 REM calculate steel part
2400 BN=BAR(EL)
2410 FOR EL1=1 TO BN
2420 XB1=XB(EL,EL1):YB1=YB(EL,EL1)
2430 XE1=XE(EL,EL1):YE1=YE(EL,EL1)
2440 LENG=SQR((XE1-XB1)^2+(YE1-YB1)^2)
2450 SN=(YE1-YB1)/LENG
2460 CS=(XE1-XB1)/LENG
2470 IF CASE=6 THEN 2490
2480 IF CASE=7 THEN 2500
2490 IF SN=0 OR (YE1-YB1)<0 THEN 2510 ELSE 2750
2500 IF SN>0 OR (YE1-YB1)>0 THEN 2510 ELSE 2750
2510 DST(1)=CS^2
2520 DST(2)=SN^2
2530 DST(3)=SN*CS
2540 DX=(XE1-XB1)/20
2550 DY=(YE1-YB1)/20
2560 FOR J=1 TO 21
2570 X=XB1+DX*(J-1)
2580 V=X-X(NN(1))
2590 L=X(NN(2))-X(NN(1))
2600 GOSUB 3010
2610 FX(1)=F1:FX(2)=F2:FX(3)=F3:FX(4)=F4
2620 PX(1)=F5:PX(2)=F6:PX(3)=F7:PX(4)=F8
2630 Y=YB1+DY*(J-1)
2640 V=Y-Y(NN(1))
2650 L=Y(NN(4))-Y(NN(1))
2660 GOSUB 3010
2670 FY(1)=F1:FY(2)=F2:FY(3)=F3:FY(4)=F4
2680 PY(1)=F5:PY(2)=F6:PY(3)=F7:PY(4)=F8
2690 GOSUB 3250
2700 IF MAX<ET THEN 2710 ELSE 2720
2710 MAX=ET:XMN=X:YMN=Y
2720 IF MIN>ET THEN 2730 ELSE 2740
2730 MIN=ET:XMN=X:YMN=Y
2740 NEXT J
2750 NEXT EL1
2760 NEXT EL
2770 IF CASE=1 THEN 2840
2780 IF CASE=2 THEN 2860
2790 IF CASE=3 THEN 2880
2800 IF CASE=4 THEN 2900
2810 IF CASE=5 THEN 2920
2820 IF CASE=6 THEN 2940
2830 IF CASE=7 THEN 2960
2840 LOCATE 23,5:PRINT "MAXIMUM PRINCIPAL STRAIN "
2850 GOTO 2970
2860 LOCATE 23,5:PRINT "MINIMUM PRINCIPAL STRAIN "
2870 GOTO 2970
2880 LOCATE 23,5:PRINT "NORMAL STRAIN IN X-DIRECTION"
2890 GOTO 2970

```

```

2900 LOCATE 23,5:PRINT "NORMAL STRAIN IN Y-DIRECTION"
2910 GOTO 2970
2920 LOCATE 23,5:PRINT "SHEARING STRAIN WITH RESPECT TO X,Y
AXES"
2930 GOTO 2970
2940 LOCATE 23,5:PRINT "STRAIN OF STEEL BAR(horizontal or
greater than 90 deg)"
2950 GOTO 2970
2960 LOCATE 23,5:PRINT "STRAIN OF STEEL BAR(vertical or less
than 90 deg)"
2970 LOCATE 24,9:PRINT "MAX";MAX;"x coord.":XMX;"y coord.":YMX
;"MIN";MIN;"x coord.":XMN;"y coord.":YMN
2980 AS=INKEY$:IF AS="" THEN 2980
2990 NEXT CASE
3000 END
3010 LV=L-V
3020 F1=LV/L+V*LV*(L-2*V)/L^3
3030 F2=V/L+V*LV*(2*V-L)/L^3
3040 F3=V*LV^2/L^2
3050 F4=-LV*V^2/L^2
3060 F5=(-6*V*L+6*V^2)/L^3
3070 F6=(6*V*L-6*V^2)/L^3
3080 F7=(L^2-4*L*V+3*V^2)/L^2
3090 F8=(-2*L*V+3*V^2)/L^2
3100 RETURN
3110 V1=X-X(NN(1)):L1=X(NN(2))-X(NN(1))
3120 V2=Y-Y(NN(1)):L2=Y(NN(4))-Y(NN(1))
3130 FOR II=1 TO 2
3140 V=V1:L=L1
3150 IF II=2 THEN V=V2
3160 IF II=2 THEN L=L2
3170 GOSUB 3010
3180 IF II=2 THEN 3220
3190 FX(1)=F1:FX(2)=F2:FX(3)=F3:FX(4)=F4
3200 PX(1)=F5:PX(2)=F6:PX(3)=F7:PX(4)=F8
3210 GOTO 3240
3220 FY(1)=F1:FY(2)=F2:FY(3)=F3:FY(4)=F4
3230 PY(1)=F5:PY(2)=F6:PY(3)=F7:PY(4)=F8
3240 NEXT II
3250 FOR II=1 TO 4
3260 C=(II-1)*6
3270 IF II=1 OR II=4 THEN K=1 ELSE K=2
3280 IF II <=2 THEN KK=1 ELSE KK=2
3290 B(1,1+C)=PX(K)*FY(KK):B(2,2+C)=FX(K)*PY(KK)
3300 B(1,3+C)=PX(K+2)*FY(KK):B(2,4+C)=FX(K+2)*PY(KK)
3310 B(1,5+C)=PX(K)*FY(KK+2):B(2,6+C)=FX(K)*PY(KK+2)
3320 B(3,1+C)=B(2,2+C):B(3,2+C)=B(1,1+C)
3330 B(3,3+C)=B(2,4+C):B(3,4+C)=B(1,3+C)
3340 B(3,5+C)=B(2,6+C):B(3,6+C)=B(1,5+C)
3350 NEXT II
3360 FOR I1=1 TO 3
3370 EP(I1)=0
3380 FOR J1=1 TO 24

```

```
3390 EP(I1)=EP(I1)+B(I1,J1)*UU(J1)
3400 NEXT J1
3410 NEXT I1
3420 IF CASE<6 THEN 3490
3430 ET=0
3440 FOR J1=1 TO 3
3450 ET=DST(J1)*EP(J1)+ET
3460 NEXT J1
3470 Z=ET
3480 GOTO 3590
3490 IF CASE<3 THEN 3500 ELSE 3570
3500 HE1=(EP(1)+EP(2))/2
3510 HE2=((EP(1)-EP(2))/2)^2
3520 HE3=(HE2+EP(3)^2)^.5
3530 IF CASE=1 THEN 3540 ELSE 3550
3540 MGT=HE1+HE3:Z=MGT:GOTO 3590
3550 MGT=HE1-HE3:Z=MGT
3560 GOTO 3590
3570 MGT=EP(CASE-2)
3580 Z=MGT
3590 XP=5*X+Y*CS1
3600 YP=10+Y*SN1+50000!*Z
3610 IF J>1 THEN 3640
3620 LINE (1.3*XP-55.180-YP*.65)-(1.3*XP-55.180-YP*.65)
3630 GOTO 3650
3640 LINE -(1.3*XP-55.180-YP*.65)
3650 RETURN
```

```

10 REM
20 REM          PROGRAM NAME:  TRAJECTORIES
30 REM          *****
40 REM This program gives the graph of trajectories of the
   stresses.
50 DIM X(71),Y(71),ND(52,4),W#(3),H#(3),NN(4),FX(4),PX(4),FY(
   4),PY(4),B(3,24),E(3)
60 DIM D(3,3),DB(3,24),U(426),UU(24),UT(426),SIG(3),ELMT(2)
70 EC=2468410! :REM modulus of elasticity of conc.-- unit
   (N/sq.cm )
80 M=.2:REM poisson's ratio
90 TH1=60
100 TH2=10
110 TH3=55
120 REM x,y coordinate of each node
130 NM=71:REM total node numbers
140 NE=52:REM no of total elements
150 FOR I=1 TO NM
160 READ X(I)
170 NEXT I
180 DATA      0,      0,  32.5,  32.5,  40.0,  40.0,  40.0,  40.0,
   40.0,  40.0,  40.0
190 DATA  50.0,  50.0,  50.0,  50.0,  50.0,  50.0,  50.0,  50.0
200 DATA  60.0,  60.0,  60.0,  60.0,  60.0,  60.0,  60.0,  60.0
210 DATA 120.0,120.0,120.0,120.0,120.0,120.0,120.0,120.0
220 DATA 190.0,190.0,190.0,190.0,190.0,190.0,190.0,190.0
230 DATA 260.0,260.0,260.0,260.0,260.0,260.0,260.0,260.0
240 DATA 320.0,320.0,320.0,320.0,320.0,320.0,320.0,320.0
250 DATA 330.0,330.0,330.0,330.0,330.0,330.0,330.0,330.0
260 DATA 340.0,340.0,340.0,340.0,340.0,340.0,340.0,340.0
270 DATA 347.5,347.5,380.0,380.0
280 FOR I=1 TO NM
290 READ Y(I)
300 NEXT I
310 DATA 0.0,43.5,193.5,233.5
320 DATA 0.0, 43.5, 53.5, 63.5,118.5,193.5,233.5
330 DATA 0.0, 43.5, 53.5, 63.5,118.5,193.5,233.5
340 DATA 0.0, 43.5, 53.5, 63.5,118.5,193.5,233.5
350 DATA 0.0, 43.5, 53.5, 63.5,118.5,193.5,233.5
360 DATA 0.0, 43.5, 53.5, 63.5,118.5,193.5,233.5
370 DATA 0.0, 43.5, 53.5, 63.5,118.5,193.5,233.5
380 DATA 0.0, 43.5, 53.5, 63.5,118.5,193.5,233.5
390 DATA 0.0, 43.5, 53.5, 63.5,118.5,193.5,233.5
400 DATA 0.0, 43.5, 53.5, 63.5,118.5,193.5,233.5
410 DATA 193.5,233.5, 0.0, 43.5
420 REM
430 REM NODE NO. CORRESPONDING TO EACH ELEMENT
440 FOR EL=1 TO NE
450 FOR EN=1 TO 4
460 READ ND(EL,EN)
470 NEXT EN:NEXT EL
480 DATA 1, 5, 6, 2, 5,12,13, 6, 12,19,20,13, 19,26,27,20.

```

```

26,33,34,27
490 DATA 33,40,41,34, 40,47,48,41, 47,54,55,48, 54,61,62,55,
61,70,71,62
500 DATA 6,13,14, 7, 13,20,21,14, 20,27,28,21, 27,34,35,26,
34,41,42,35
510 DATA 41,48,49,42, 48,55,56,49, 55,62,63,56, 7,14,15, 8,
14,21,22,15
520 DATA 21,28,29,22, 28,35,36,29, 35,42,43,36, 42,49,50,43,
49,56,57,50
530 DATA 56,63,64,57, 8,15,16, 9, 15,22,23,16, 22,29,30,23,
29,36,37,30
540 DATA 36,43,44,37, 43,50,51,44, 50,57,58,51, 57,64,65,58,
9,16,17,10
550 DATA 16,23,24,17, 23,30,31,24, 30,37,38,31, 37,44,45,36,
44,51,52,45
560 DATA 51,58,59,52, 58,65,66,59, 3,10,11, 4, 10,17,18,11,
17,24,25,18
570 DATA 24,31,32,25, 31,38,39,32, 38,45,46,39, 45,52,53,46,
52,59,60,53
580 DATA 59,66,67,60, 66,68,69,67
590 D1=EC/(1-M*M)
600 D(1,1)=D1:D(1,2)=D1*M
610 D(2,1)=D(1,2):D(2,2)=D1
620 D(3,3)=(1-M)*D1/2
630 OPEN "R",#5,"B:RRDISP",4
640 FIELD #5,4 AS D8
650 FOR I=1 TO 426
660 GET #5,I
670 U(I)=CVS(D8)
680 NEXT I
690 CLOSE #5
700 REM
710 PI=3.1415927#
720 CS=COS(PI/4)
730 SN=SIN(PI/4)
740 INPUT"NO. OF LINES ":NLS
750 INPUT"NO. OF LINES IN HORIZONTAL PART":NLS$
760 INPUT"HOW MANY ELEMENTS ARE YOU GOING TO INVOLVE ON
GRAPH(type no.)":HM
770 FOR I=1 TO HM
780 INPUT"ELEMENT NO.(type no.)":ELMT(I)
790 NEXT I
800 FOR EL=1 TO HM
810 IF ELMT(EL)=30 OR ELMT(EL)=31 THEN PP=-50 ELSE PP=-100
820 IF EL=1 THEN SIGN=-1 ELSE SIGN=1
830 IF EL=1 THEN Z=2 ELSE Z=1
840 IF EL=1 THEN SN=1 ELSE SN=-1
850 FOR K=1 TO 4
860 NN(K)=ND(ELMT(EL),K)
870 NEXT K
880 SP=Y(NN(3))-Y(NN(2))
890 DPTH=X(NN(2))-X(NN(1))
900 T=0

```

```

910 FOR II=1 TO 4
920 FOR JJ=1 TO 6
930 T=T+1
940 IJ=(NN(II)-1)*6+JJ
950 UU(T)=U(IJ)
960 NEXT JJ,II
970 IF EL>1 THEN 990
980 CLS
990 FOR I=1 TO 4
1000 J=I+1:IF J=5 THEN J=1
1010 XP=20-X(NN(I)):YP=Y(NN(I))+PP
1020 XQ=20-X(NN(J)):YQ=Y(NN(J))+PP
1030 LINE (XP+4-520,200-YP+2)-(XQ+4-520,200-YQ+2)
1040 NEXT I
1050 DLT Y=SP/(NLS-1)
1060 DLT X=DPTH/100
1070 DLT T X=DPTH/NLSS
1080 FOR JJ=1 TO 2
1090 Y=0:X=0
1100 FOR IY=1 TO NLS
1110 Y=Y(NN(Z))+(IY-1)*DLTY
1120 FOR IX=1 TO 100
1130 YT=Y
1140 X=X(NN(Z))+(IX-1)*DLTX+SIGN
1150 XT=X
1160 X=X+DLTX+SIGN
1170 GOSUB 1500
1180 IF Y>Y(NN(3)) OR Y<Y(NN(2)) THEN 1230
1190 IF X>X(NN(2)) OR X<X(NN(1)) THEN 1230
1200 IF YT>Y(NN(3)) OR YT<Y(NN(2)) THEN 1230
1210 IF XT>X(NN(2)) OR XT<X(NN(1)) THEN 1230
1220 NEXT IX
1230 NEXT IY
1240 NEXT JJ
1250 FOR JJ=1 TO 2
1260 Y=0:X=0
1270 FOR XX=1 TO NLSS-1
1280 IF EL=1 AND JJ=1 THEN Z=3
1290 IF EL=1 AND JJ=2 THEN Z=2
1300 IF EL=2 AND JJ=1 THEN Z=4
1310 IF EL=2 AND JJ=2 THEN Z=1
1320 XS=X(NN(Z))+XX*DLT T X+SIGN:Y=Y(NN(Z))
1330 FOR IX=1 TO 100
1340 YT=Y
1350 X=XS+(IX-1)*DLTX+SIGN
1360 XT=X
1370 X=X+DLT X+SIGN
1380 GOSUB 1500
1390 IF Y>Y(NN(3)) OR Y<Y(NN(2)) THEN 1440
1400 IF X>X(NN(2)) OR X<X(NN(1)) THEN 1440
1410 IF YT>Y(NN(3)) OR YT<Y(NN(2)) THEN 1440
1420 IF XT>X(NN(2)) OR XT<X(NN(1)) THEN 1440
1430 NEXT IX

```

```

1440 NEXT XX
1450 NEXT JJ
1460 NEXT EL
1470 LOCATE 24,22:PRINT "TRAJECTORIES OF THE STRESSES"
1480 A$=INKEY$:IF A$="" THEN 1480
1490 END
1500 V1=X-X(NN(1)):L1=X(NN(2))-X(NN(1))
1510 V2=Y-Y(NN(1)):L2=Y(NN(4))-Y(NN(1))
1520 FOR II=1 TO 2
1530 V=V1:L=L1
1540 IF II=2 THEN V=V2
1550 IF II=2 THEN L=L2
1560 GOSUB 2070
1570 IF II=2 THEN 1610
1580 FX(1)=F1:FX(2)=F2:FX(3)=F3:FX(4)=F4
1590 PX(1)=F5:PX(2)=F6:PX(3)=F7:PX(4)=F8
1600 GOTO 1630
1610 FY(1)=F1:FY(2)=F2:FY(3)=F3:FY(4)=F4
1620 PY(1)=F5:PY(2)=F6:PY(3)=F7:PY(4)=F8
1630 NEXT II
1640 FOR II=1 TO 4
1650 C=(II-1)*6
1660 IF II=1 OR II=4 THEN K=1 ELSE K=2
1670 IF II<=2 THEN KK=1 ELSE KK=2
1680 B(1,1+C)=PX(K)*FY(KK):B(2,2+C)=FX(K)*PY(KK)
1690 B(1,3+C)=PX(K+2)*FY(KK):B(2,4+C)=FX(K+2)*PY(KK)
1700 B(1,5+C)=PX(K)*FY(KK+2):B(2,6+C)=FX(K)*PY(KK+2)
1710 B(3,1+C)=B(2,2+C)
1720 B(3,2+C)=B(1,1+C)
1730 B(3,3+C)=B(2,4+C)
1740 B(3,4+C)=B(1,3+C)
1750 B(3,5+C)=B(2,6+C)
1760 B(3,6+C)=B(1,5+C)
1770 NEXT II
1780 FOR K=1 TO 3
1790 E(K)=0
1800 FOR KK=1 TO 24
1810 E(K)=E(K)+B(K, KK)*UU(KK)
1820 NEXT KK,K
1830 FOR K=1 TO 3
1840 SIG(K)=0
1850 FOR KK=1 TO 3
1860 SIG(K)=SIG(K)+D(K, KK)*E(KK)
1870 NEXT KK,K
1880 PHI=0
1890 S1=SIG(1):S2=SIG(2):TXY=SIG(3)
1900 SS=S1-S2
1910 IF SS=0 THEN PHI=0:GOTO 1980
1920 IF EL=1 AND JJ=2 THEN 1960
1930 IF EL=1 AND JJ=1 THEN 1970
1940 IF EL=2 AND JJ=2 THEN 1960
1950 IF EL=2 AND JJ=1 THEN 1970
1960 PHI=ATN((-2*TXY/(S1-S2))/2):GOTO 1980

```

```

1970 PHI=(ATN(-2*TX/(S1-S2))+PI)/2
1980 TN=TAN(PHI)
1990 Y=YT+TN*DLTX*SN
2000 IF Y>Y(NN(3)) OR Y<Y(NN(2)) THEN 2060
2010 IF X>X(NN(2)) OR X<X(NN(1)) THEN 2060
2020 IF YT>Y(NN(3)) OR YT<Y(NN(2)) THEN 2060
2030 IF XT>X(NN(2)) OR XT<X(NN(1)) THEN 2060
2040 XPT=XT+20:YPT=YT+PP:YP=Y+PP:XP=X+20
2050 LINE (XPT*4-520,200-YPT*2)-(XP*4-520,200-YP*2)
2060 RETURN
2070 LV=L-V
2080 F1=LV/L+V*LV*(L-2*V)/L^3
2090 F2=V/L+V*LV*(2*V-L)/L^3
2100 F3=V*LV^2/L^2
2110 F4=-LV*V^2/L^2
2120 F5=(-6*V*L+6*V^2)/L^3
2130 F6=(6*V*L-6*V^2)/L^3
2140 F7=(L^2-4*L*V+3*V^2)/L^2
2150 F8=(-2*L*V+3*V^2)/L^2
2160 RETURN

```



```

10 REM
20 REM          PROGRAM NAME:ENERGY
30 REM          *****
40 REM PROGRAM TO VERIFY ZERO STRAIN ENERGY PICTURE OF RIGID
   BODY MOTION
50 DIM XST(21),XND(21),YST(21),YND(21),X(4),Y(4),D(3,3),E(3),
   G(3),U(24),B(3,24),N(2,24),FX(4),FY(4),PX(4),PY(4)
60 FOR I=1 TO 4
70 READ X(I),Y(I)
80 NEXT I
90 DATA 0,0,160,0,160,160,0,160
100 PI=3.1415927#
110 CS=COS(PI/4)
120 SN=SIN(PI/4)
130 PRINT
140 INPUT"WHAT KIND OF STRAIN CONDITION APPLIED [ (Ui+(-).1
   & Vi+(-).1)---- type 1 (THETA=.00125)---- type 2
   (Ui+(-).1 & Vi+(-).1 & THETA=.00125)----type 3]";A
150 PRINT
160 INPUT"DO YOU WANT U or V";AS
170 IF AS="N" OR AS="n" THEN 210
180 PRINT
190 INPUT"WHAT DO YOU WANT U-----type 1 or V----- type 2
   ";US
200 GOTO 270
210 PRINT :INPUT"DO YOU WANT EACH OF STRAIN      Y/N";AS
220 IF AS="N" OR AS="n" THEN 250
230 PRINT :INPUT"WHAT DO YOU WANT(Epsilon x(1)      Epsilon
   y(2)      Gamma xy(3)      TYPE NO.";AA
240 GOTO 270
250 PRINT :INPUT"DO YOU WANT THE STRAIN ENERGY PICTURE
   TYPE Y ";A#
260 SE=1
270 IF A=2 THEN 360
280 FOR I=1 TO 24
290 READ U(I)
300 NEXT I
310 DATA .1,-.1,0,0,0,0
320 DATA .1,.1,0,0,0,0
330 DATA -.1,.1,0,0,0,0
340 DATA -.1,-.1,0,0,0,0
350 IF A=1 THEN 440
360 THTA=.00125
370 FOR I=1 TO 4
380 P=6*(I-1)+1
390 U(P+2)=0
400 U(P+3)=THTA
410 U(P+4)=-THTA
420 U(P+5)=0
430 NEXT I
440 E=60000!:M=.17
450 D1=E/(1-M*M)

```

```

460 D(1,1)=D1:D(1,2)=D1*M
470 D(2,1)=D(1,2):D(2,2)=D1:D(3,3)=(1-M)*D1/2
480 DX=(X(2)-X(1))/20
490 DY=(Y(4)-Y(1))/20
500 FOR I=1 TO 21
510 XST(I)=X(1)
520 XND(I)=X(2)
530 YST(I)=Y(3)
540 YND(I)=Y(2)
550 NEXT I
560 H=0:V=0
570 CLS
580 FOR I=1 TO 4
590 J=I+1:IF J=5 THEN J=1
600 XP=20+X(I)+Y(I)*CS:YP=25+Y(I)*SN
610 XQ=20+X(J)+Y(J)*CS:YQ=25+Y(J)*SN
620 LINE (XP*2,220-YP)-(XQ*2,220-YQ)
630 NEXT I
640 FOR I=1 TO 4
650 X1=X(I):Y1=Y(I)
660 I2=I+1
670 IF I2=5 THEN I2=1
680 X2=X(I2):Y2=Y(I2)
690 DPX=(X2-X1)/100:DPY=(Y2-Y1)/100
700 FOR J=1 TO 101
710 X=X1+(J-1)*DPX:Y=Y1+(J-1)*DPY
720 GOSUB 1280
730 NEXT J
740 NEXT I
750 FOR I=1 TO 2
760 FOR XI=1 TO 21
770 IF I=2 THEN 820
780 XO=XST(XI)
790 XEND=XND(XI)
800 DDX=(XEND-XO)/100
810 Y=YMN+DY*(XI-1):GOTO 860
820 YO=YST(XI)
830 YEND=YND(XI)
840 DDY=(YEND-YO)/100
850 X=XMN+DX*(XI-1)
860 FOR J=1 TO 101
870 IF I=2 THEN 890
880 X=XO+(J-1)*DDX:GOTO 900
890 Y=YO+(J-1)*DDY
900 GOSUB 1280
910 NEXT J
920 NEXT XI
930 NEXT I
940 IF SE=1 THEN 1180
950 IF US=1 THEN 970
960 IF US=2 THEN 990 ELSE 1010
970 LOCATE 24,15:PRINT "U"
980 GOTO 1090

```

```

990 LOCATE 24,15:PRINT "V"
1000 GOTO 1090
1010 IF AA=1 THEN 1040
1020 IF AA=2 THEN 1060
1030 IF AA=3 THEN 1080
1040 LOCATE 24,5:PRINT "Epsilon x"
1050 GOTO 1090
1060 LOCATE 24,5:PRINT "Epsilon y"
1070 GOTO 1090
1080 LOCATE 24,5:PRINT "GAMMA xy"
1090 IF A=1 THEN 1120
1100 IF A=2 THEN 1140
1110 IF A=3 THEN 1160
1120 LOCATE 24,20:PRINT "DUE TO U1,2=.1 U3,4=-.1 & V1,4=-.1
      V2,3=.1"
1130 GOTO 1260
1140 LOCATE 24,20:PRINT "DUE TO THETA=.00125"
1150 GOTO 1260
1160 LOCATE 24,20:PRINT "DUE TO U1,2=.1 U3,4=-.1,V1,4=-.1
      V2,3=.1 & THETA=.00125"
1170 GOTO 1260
1180 IF A=1 THEN 1210
1190 IF A=2 THEN 1230
1200 IF A=3 THEN 1250
1210 LOCATE 24,5 :PRINT"STRAIN ENERGY DUE TO U1,2=.1 U3,4=-.1
      & V1,4=-.1 V2,3=.1"
1220 GOTO 1260
1230 LOCATE 24,5 :PRINT"STRAIN ENERGY DUE TO THETA=.00125"
1240 GOTO 1260
1250 LOCATE 24,5 :PRINT"STRAIN ENERGY DUE TO U1,2=.1 U3,4=-
      .1, V1,4=-.1 V2,3=.1 , and THETA=.00125"
1260 A$=INKEY$:IF A$="" THEN 1260
1270 END
1280 V1=X-X(1):L1=X(2)-X(1)
1290 V2=Y-Y(1):L2=Y(4)-Y(1)
1300 FOR II=1 TO 2
1310 V=V1:L=L1
1320 IF II=2 THEN V=V2
1330 IF II=2 THEN L=L2
1340 LV=L-V
1350 F1=LV/L+V*LV*(L-2*V)/L^3
1360 F2=V/L+V*LV*(2*V-L)/L^3
1370 F3=V*LV^2/L^2
1380 F4=-LV*V^2/L^2
1390 IF US=1 OR US=2 THEN 1440
1400 F5=(-6*V*L+6*V^2)/L^3
1410 F6=(6*V*L-6*V^2)/L^3
1420 F7=(L^2-4*L*V+3*V^2)/L^2
1430 F8=(-2*L*V+3*V^2)/L^2
1440 IF II=2 THEN 1480
1450 FX(1)=F1:FX(2)=F2:FX(3)=F3:FX(4)=F4
1460 PX(1)=F5:PX(2)=F6:PX(3)=F7:PX(4)=F8
1470 GOTO 1500

```

```

1480 FY(1)=F1:FY(2)=F2:FY(3)=F3:FY(4)=F4
1490 PY(1)=F5:PY(2)=F6:PY(3)=F7:PY(4)=F8
1500 NEXT II
1510 FOR II=1 TO 4
1520 IF II=1 THEN C=0
1530 IF II=2 THEN C=6
1540 IF II=3 THEN C=12
1550 IF II=4 THEN C=18
1560 IF II=1 OR II=4 THEN K=1 ELSE K=2
1570 IF II<=2 THEN KK=1 ELSE KK=2
1580 N(1,1+C)=FX(K)*FY(KK)
1590 N(1,3+C)=FX(K+2)*FY(KK)
1600 N(1,5+C)=FX(K)*FY(KK+2)
1610 N(2,2+C)=N(1,1+C)
1620 N(2,4+C)=N(1,3+C)
1630 N(2,6+C)=N(1,5+C)
1640 IF US=1 OR US=2 THEN 1710
1650 B(1,1+C)=PX(K)*FY(KK):B(2,2+C)=FX(K)*PY(KK)
1660 B(1,3+C)=PX(K+2)*FY(KK):B(2,4+C)=FX(K+2)*PY(KK)
1670 B(1,5+C)=PX(K)*FY(KK+2):B(2,6+C)=FX(K)*PY(KK+2)
1680 B(3,1+C)=B(2,2+C):B(3,2+C)=B(1,1+C)
1690 B(3,3+C)=B(2,4+C):B(3,4+C)=B(1,3+C)
1700 B(3,5+C)=B(2,6+C):B(3,6+C)=B(1,5+C)
1710 NEXT II
1720 IF US=1 OR US=2 THEN 1940
1730 FOR K=1 TO 3
1740 E(K)=0
1750 FOR KK=1 TO 24
1760 E(K)=E(K)+B(K, KK)*U(KK)
1770 NEXT KK, K
1780 IF SE=0 THEN 1900
1790 FOR K=1 TO 3
1800 G(K)=0
1810 FOR KK=1 TO 3
1820 G(K)=G(K)+D(K, KK)*E(KK)
1830 NEXT KK, K
1840 FOR K=1 TO 3
1850 GG=GG+E(K)*G(K)
1860 NEXT K
1870 GG=GG/2
1880 Z=GG:GG=0
1890 SZ=800:GOTO 2010
1900 IF AA=1 THEN Z=E(1)
1910 IF AA=2 THEN Z=E(2)
1920 IF AA=3 THEN Z=E(3)
1930 SZ=10000:GOTO 2010
1940 UU=0
1950 FOR II=1 TO 24
1960 IF US=1 THEN G=1 ELSE G=2
1970 UU=UU+U(II)*N(G, II)
1980 NEXT II
1990 Z=UU:SZ=150
2000 IF A=2 THEN SZ=500

```

```
2010 XP=20+X+Y*CS
2020 YP=25+Y*SN+SZ*2
2030 IF J>1 THEN 2050
2040 LINE (2*XP,220-YP)-(2*XP,220-YP):GOTO 2060
2050 LINE-(2*XP,220-YP)
2060 RETURN
```

```

10 REM
20 REM          PROGRAM NAME: HERMITIAN
30 REM          *****
40 DIM XST(21),XND(21),YST(21),YND(21),X(4),Y(4)
50 FOR I=1 TO 4
60 READ X(I),Y(I)
70 NEXT I
80 DATA 0,0,120,32,168,100,84,128
90 PI=3.1415927#
100 CS=COS(PI/4)
110 SN=SIN(PI/4)
120 INPUT"NODE OF INTEREST";ISN
130 INPUT"DO YOU WANT THE GRAPH WHEN UNIT NODAL PARAMETER
    APPLIES Y/N";AS
140 IF AS="Y" OR AS="y" THEN 270
150 INPUT"DO YOU WANT 2nd ORDER f(x,y) Y/N";A#
160 IF AS="N" OR AS="n" THEN 190
170 INPUT"WHAT DO YOU WANT (W=1 W,x=2 W,y=3 W,xx=4 W,xy=5
    W,yy=6) TYPE NO.";F
180 GOTO 290
190 INPUT"DO YOU WANT f(x,y) Y/N";A#
200 IF AS="N" OR AS="n" THEN 220
210 F=7:GOTO 290
220 INPUT"DO YOU WANT df/dx Y/N";A#
230 IF AS="N" OR AS="n" THEN 250
240 F=8:GOTO 290
250 INPUT"DO YOU WANT df/dy Y/N";A#
260 IF AS="Y" OR AS="y" THEN F=9:GOTO 290
270 INPUT"DO YOU WANT THE GRAPH BY (1st(1) OR 2nd (2) ORDER)
    TYPE NO.";OD
280 INPUT"WHAT GRAPH DO YOU WANT(u=1 v=2 u,x=3 v,x=4 u,y=5
    v,y=6) TYPE NO.";S
290 XM=1:YM=1:YN=1:YN=1:YMN=X(1):YMX=Y(1):XMN=X(1):YMN=Y(1)
300 FOR K=1 TO 4
310 IF XMX>X(K) THEN 330
320 XMX=X(K):XM=K
330 IF YMX>Y(K) THEN 350
340 YMX=Y(K):YM=K
350 IF XMN<X(K) THEN 370
360 XMN=X(K):XN=K
370 IF YMN<Y(K) THEN 390
380 YMN=Y(K):YN=K
390 NEXT K
400 DX=(XMX-XMN)/20
410 DY=(YMX-YMN)/20
420 N1=YN:N2=YN+1:IF N2=5 THEN N2=1
430 N3=N1+2:IF N3=5 THEN N3=1
440 N4=N1-1:IF N4=0 THEN N4=4
450 NN1=XN:NN2=XN+1:IF NN2=5 THEN NN2=1
460 NN3=NN1+2:IF NN3=5 THEN NN3=1
470 NN4=NN1-1:IF NN4=0 THEN NN4=4
480 FOR I=1 TO 21

```

```

490 Y=YMN+(I-1)*DY
500 IF Y<=Y(N4) THEN 540
510 K=(Y-Y(N4))/(Y(N3)-Y(N4))
520 XST(I)=X(N4)+K*(X(N3)-X(N4))
530 GOTO 560
540 K=(Y-Y(N1))/(Y(N4)-Y(N1))
550 XST(I)=X(N1)+K*(X(N4)-X(N1))
560 IF Y<=Y(N2) THEN 620
570 IF YH=N4 AND Y>Y(N3) THEN 600
580 K=(Y-Y(N2))/(Y(N3)-Y(N2))
590 XND(I)=X(N2)+K*(X(N3)-X(N2)):GOTO 640
600 K=(Y-Y(N3))/(Y(N4)-Y(N3))
610 XND(I)=X(N3)+K*(X(N4)-X(N3)):GOTO 640
620 K=(Y-Y(N1))/(Y(N2)-Y(N1))
630 XND(I)=X(N1)+K*(X(N2)-X(N1))
640 X=XMN+(I-1)*DX
650 IF X<=X(NN4) THEN 690
660 K=(X-X(NN4))/(X(NN3)-X(NN4))
670 YST(I)=Y(NN4)+K*(Y(NN3)-Y(NN4))
680 GOTO 710
690 K=(X-X(NN1))/(X(NN4)-X(NN1))
700 YST(I)=Y(NN1)+K*(Y(NN4)-Y(NN1))
710 IF X<=X(NN2) THEN 740
720 K=(X-X(NN2))/(X(NN3)-X(NN2))
730 YND(I)=Y(NN2)+K*(Y(NN3)-Y(NN2)):GOTO 760
740 K=(X-X(NN1))/(X(NN2)-X(NN1))
750 YND(I)=Y(NN1)+K*(Y(NN2)-Y(NN1))
760 NEXT I
770 H=0:V=0
780 CLS
790 FOR I=1 TO 4
800 J=I+1:IF J=5 THEN J=1
810 XT=X(I):YT=Y(I)
820 XP=20+X(I)+Y(I)*CS:YP=25+Y(I)*SN
830 XT=X(J):YT=Y(J)
840 XQ=20+X(J)+Y(J)*CS:YQ=25+Y(J)*SN
850 IF H<XP THEN H=XP
860 IF H<XQ THEN H=XQ
870 IF V<YP THEN V=YP
880 IF V<YQ THEN V=YQ
890 LINE (XP=2,220-YP)-(XQ=2,220-YQ)
900 NEXT I
910 FOR I=1 TO 4
920 X1=X(I):Y1=Y(I)
930 I2=I+1
940 IF I2=5 THEN I2=1
950 X2=X(I2):Y2=Y(I2)
960 DPX=(X2-X1)/100:DPY=(Y2-Y1)/100
970 FOR J=1 TO 101
980 X=X1+(J-1)*DPX:Y=Y1+(J-1)*DPY
990 GOSUB 1740
1000 NEXT J
1010 NEXT I

```

```

1020 FOR I=1 TO 2
1030 FOR XI=1 TO 21
1040 IF I=2 THEN 1090
1050 XO=XST(XI)
1060 XEND=XND(XI)
1070 DDX=(XEND-XO)/100
1080 Y=YMN+DY*(XI-1):GOTO 1130
1090 YO=YST(XI)
1100 YEND=YND(XI)
1110 DDY=(YEND-YO)/100
1120 X=XMN+DX*(XI-1)
1130 FOR J=1 TO 101
1140 IF I=2 THEN 1160
1150 X=XO+(J-1)*DDX:GOTO 1170
1160 Y=YO+(J-1)*DDY
1170 GOSUB 1740
1180 NEXT J
1190 NEXT XI
1200 NEXT I
1210 IF F>1 GOTO 1340
1220 IF S>1 AND OD=1 THEN 1240
1230 IF S>1 AND OD=2 THEN 1260
1240 LOCATE 2,20:PRINT "By using 1st order Hermitian Shape
Function"
1250 GOTO 1280
1260 LOCATE 2,20:PRINT "By using 2nd order Hermitian Shape
Function"
1270 GOTO 1280
1280 IF S=1 THEN 1430
1290 IF S=2 THEN 1450
1300 IF S=3 THEN 1470
1310 IF S=4 THEN 1490
1320 IF S=5 THEN 1510
1330 IF S=6 THEN 1530
1340 IF F=1 THEN 1550
1350 IF F=2 THEN 1570
1360 IF F=3 THEN 1590
1370 IF F=4 THEN 1610
1380 IF F=5 THEN 1630
1390 IF F=6 THEN 1650
1400 IF F=7 THEN 1670
1410 IF F=8 THEN 1690
1420 IF F=9 THEN 1710
1430 LOCATE 24,20:PRINT"Epsilon x due to u=1 & Gamma xy due
to v=1 at node";ISN
1440 GOTO 1720
1450 LOCATE 24,20:PRINT"Epsilon y due to v=1 & Gamma xy due
to u=1 at node";ISN
1460 GOTO 1720
1470 LOCATE 24,20:PRINT"Epsilon x due to u,x=1 & Gamma xy
due to v,x=1 at node";ISN
1480 GOTO 1720
1490 LOCATE 24,20:PRINT "Epsilon y due to v,x=1 & Gamma xy

```



```

    due to u,x=1 at node";ISN
1500 GOTO 1720
1510 LOCATE 24,20:PRINT "Epsilon x due to u,y=1 & Gamma xy
    due to v,y=1 at node";ISN
1520 GOTO 1720
1530 LOCATE 24,20:PRINT "Epsilon y due to v,y=1 & Gamma xy
    due to u,y=1 at node";ISN
1540 GOTO 1720
1550 LOCATE 24,5:PRINT "2nd Order Hermitian Shape Function
    using nodal w of node";ISN
1560 GOTO 1720
1570 LOCATE 24,5:PRINT "2nd Order Hermitian Shape Function
    using nodal dw/dx of node";ISN
1580 GOTO 1720
1590 LOCATE 24,5:PRINT "2nd Order Hermitian Shape Function
    using nodal dw/dy of node";ISN
1600 GOTO 1720
1610 LOCATE 24,5:PRINT "2nd Order Hermitian Shape Function
    using nodal d(dw/dx)/dx of node";ISN
1620 GOTO 1720
1630 LOCATE 24,5:PRINT "2nd Order Hermitian Shape Function
    using nodal d(dw/dx)/dy of node";ISN
1640 GOTO 1720
1650 LOCATE 24,5:PRINT "2nd Order Hermitian Shape Function
    using nodal d(dw/dy)/dy of node";ISN
1660 GOTO 1720
1670 LOCATE 24,5:PRINT "1st Order Hermitian Shape Function
    using nodal w of node";ISN
1680 GOTO 1720
1690 LOCATE 24,5:PRINT "1st Order Hermitian Shape Function
    using nodal dw/dx of node";ISN
1700 GOTO 1720
1710 LOCATE 24,5:PRINT "1st Order Hermitian Shape Function
    using nodal dw/dy of node";ISN
1720 AS=INKEYS : IF AS="" THEN 1720
1730 END
1740 K2=ISN+1:IF K2>4 THEN K2=K2-4
1750 K3=ISN+2:IF K3>4 THEN K3=K3-4
1760 K4=ISN+3:IF K4>4 THEN K4=K4-4
1770 L1=(X(K2)-X)*(Y(K3)-Y)-(X(K3)-X)*(Y(K2)-Y)
1780 L2=(X(K3)-X)*(Y(K4)-Y)-(X(K4)-X)*(Y(K3)-Y)
1790 L3=(X(K2)-X(ISN))*(Y(K3)-Y(ISN))-(X(K3)-X(ISN))*(Y(K2)-
    Y(ISN))
1800 L4=(X(K3)-X(ISN))*(Y(K4)-Y(ISN))-(X(K4)-X(ISN))*(Y(K3)-
    Y(ISN))
1810 L34=L3*L4
1820 L=L1*L2/L34
1830 XX=X-X(ISN):YY=Y-Y(ISN)
1840 OML=1-L
1850 IF OD=2 OR (F>=1 AND F<7) THEN 1880
1860 H=1-3*OML^2+2*OML^3
1870 PHL=6*OML-6*OML^2 :GOTO 1900
1880 H=1-10*OML^3+15*OML^4-6*OML^5

```

```
1890 PHL=30*OML^2-60*OML^3+30*OML^4
1900 IF F>=7 THEN 2340
1910 IF F>=1 AND F<7 THEN 2490
1920 LX1=Y(K2)-Y(K3)
1930 LY1=X(K3)-X(K2)
1940 LX2=Y(K3)-Y(K4)
1950 LY2=X(K4)-X(K3)
1960 LX=(L1*LX2+L2*LX1)/L34
1970 LY=(L1*LY2+L2*LY1)/L34
1980 PHX=PHL*LX
1990 PHY=PHL*LY
2000 IF S=2 THEN 2190
2010 IF S=3 THEN 2100
2020 IF S=4 THEN 2230
2030 IF S=5 THEN 2140
2040 IF S=6 THEN 2280
2050 Z=PHX:SZ=1000
2060 IF ISN=2 THEN SZ=4000
2070 IF ISN=3 THEN SZ=3000
2080 IF ISN=4 THEN SZ=2000
2090 GOTO 2780
2100 Z=H*XX*PHX:SZ=30
2110 IF ISN=2 THEN SZ=10
2120 IF ISN>=3 THEN SZ=45
2130 GOTO 2780
2140 Z=YY*PHX:SZ=60
2150 IF ISN=2 THEN SZ=30
2160 IF ISN=3 THEN SZ=135
2170 IF ISN=4 THEN SZ=90
2180 GOTO 2780
2190 Z=PHY:SZ=8000
2200 IF ISN=2 THEN SZ=800
2210 IF ISN>=3 THEN SZ=4000
2220 GOTO 2780
2230 Z=XX*PHY:SZ=180
2240 IF ISN=2 THEN SZ=60
2250 IF ISN=3 THEN SZ=45
2260 IF ISN=4 THEN SZ=30
2270 GOTO 2780
2280 Z=H*YY*PHY:SZ=100
2290 IF ISN=2 THEN SZ=60
2300 IF ISN=3 THEN SZ=35
2310 IF ISN=4 THEN SZ=25
2320 GOTO 2780
2330 IF F<=6 THEN 2490
2340 IF F=7 THEN 2370
2350 IF F=8 THEN 2400
2360 IF F=9 THEN 2450
2370 Z=H:SZ=150
2380 IF ISN>=3 THEN SZ=75
2390 GOTO 2780
2400 Z=H*XX:SZ=6
2410 IF ISN=2 THEN SZ=.24
```

```
2420 IF ISN=3 THEN SZ=.8
2430 IF ISN=4 THEN SZ=2
2440 GOTO 2780
2450 Z=H*YY:SZ=3
2460 IF ISN=2 THEN SZ=1.1
2470 IF ISN=4 THEN SZ=1.8
2480 GOTO 2780
2490 IF F=1 THEN 2550
2500 IF F=2 THEN 2600
2510 IF F=3 THEN 2650
2520 IF F=4 THEN 2690
2530 IF F=5 THEN 2730
2540 IF F=6 THEN 2760
2550 Z=H
2560 IF ISN=1 OR ISN=2 THEN SZ=150
2570 IF ISN=3 THEN SZ=75
2580 IF ISN=4 THEN SZ=50
2590 GOTO 2780
2600 Z=H*XX:SZ=4
2610 IF ISN=2 THEN SZ=.4
2620 IF ISN=3 THEN SZ=2
2630 IF ISN=4 THEN SZ=1.6
2640 GOTO 2780
2650 Z=H*YY:SZ=3
2660 IF ISN=2 THEN SZ=1.3
2670 IF ISN=4 THEN SZ=1.5
2680 GOTO 2780
2690 Z=H*XX^2/2:SZ=.12
2700 IF ISN=2 OR ISN=3 THEN SZ=.07
2710 IF ISN=4 THEN SZ=.24
2720 GOTO 2780
2730 Z=H*XX*YY:SZ=.05
2740 IF ISN=2 OR ISN=3 THEN SZ=.1
2750 GOTO 2780
2760 Z=H*YY^2/2:SZ=.08
2770 IF ISN=2 OR ISN=3 THEN SZ=.24
2780 XP=20+X+Y*CS
2790 YP=25+Y*SN+SZ*Z
2800 IF J>1 THEN 2820
2810 LINE (2*XP,220-YP)-(2*XP,220-YP):GOTO 2830
2820 LINE-(2*XP,220-YP)
2830 RETURN
```

ON THE DEVELOPMENT OF COMPUTER PROGRAMS
FOR THE STRESS ANALYSIS OF SHEAR WALLS
USING HERMITIAN INTERPOLATION

by

KI-JANG LEE

B.S., Han Yang University, 1981
Seoul, Korea

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Civil Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1986

ABSTRACT

The state-of-the-art of stress analysis and design of shear walls were reviewed. The mathematical properties of Hermitian shape functions were investigated.

The development of a sequence of computer programs based on the use of finite element method as well as the data management and reduction for the Zenith 100 computer was discussed and presented. The applications of these programs were illustrated by solving a model shear wall. The obtained results were displayed by computer plots. The numerical values were compared to the experimental data. The findings through the investigation were summarized in the conclusions.