A FINITE DIFFERENCE APPROACH TO BUCKLING OF CONCRETE PLATES

by

FRANCIS ALAN WILEY

B. S. Civil Engineering, 1974

Kansas State University

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

1976

Approved by:

Stuart E Swartz

Major Professor

TABLE OF CONTENTS

175

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

One area of structural behavior which has been intensively studied for many years is the problem of plates buckling when subjected to in-plane forces. Numerical solutions, along with design tables and graphs, do exist for many cases of elastic buckling, with several different loading and support conditions (1,3). However, the nature of these solutions is farily limited. A relatively short and inexpensive computer program which could be applicable to most of the various loading and support conditions would be desirable.

Furthermore, concrete is a material which seldom buckles elastically. Swartz and Rosebraugh (4) describe two methods of obtaining buckling loads for steel-reinforced concrete plates. By incorporating one of these methods, the tangent modulus method, into a program for elastic buckling, a quick and inexpensive method of accurately predicting the buckling loads for concrete plates would be obtained.

This thesis describes the development of a computer program to predict the elastic buckling loads for plates with several different loading and support cases. This program was modified to predict buckling in reinforced concrete plates, subjected to various in-plane distributed forces and support conditions.

DEVELOPMENT OF ELASTIC BUCKLING PROGRAM

Assuming that the material is isotropic, linearly elastic, and that small deflection theory is applicable, the equilibrium equation stating the interaction between the in-plane applied forces and the resulting lateral displacements when the plate begins to buckle (3) takes the following form:

$$\nabla^4 w = \frac{-1}{D} N_x \left(\frac{\partial^2 w}{\partial x^2} + 2\xi \frac{\partial^2 w}{\partial x \partial y} + \zeta \frac{\partial^2 w}{\partial y^2} \right) \text{-----------------} [1]$$

In this equation, $\xi = \frac{N_{xy}}{N_x}$, $\zeta = \frac{N_y}{N_x}$, $D = \frac{Eh^3}{12(1-\nu^2)}$ is the plate modulus, and w is the lateral displacement of the middle surface of the plate. In the equation for D, E represents Young's Modulus, ν is Poisson's ratio, and h is the plate thickness. The positive sense of the applied forces, and the notation for plate geometry and loads are shown in Fig. 1a.

Using the finite difference approximations (2), equation [1] may be stated as:

$$\begin{aligned} & w_{i,j-2} + 2w_{i-1,j-1} - 8w_{i,j-1} + 2w_{i+1,j-1} + w_{i-2,j} \\ & - 8w_{i-1,j} + 20w_{i,j} - 8w_{i+1,j} + w_{i+2,j} + 2w_{i-1,j+1} \\ & - 8w_{i,j+1} + 2w_{i+1,j+1} + w_{i,j+2} + \frac{\Delta^2}{D} N_x \left[w_{i-1,j} \right. \\ & - 2w_{i,j} + w_{i+1,j} + \frac{\xi}{2} (w_{i+1,j-1} - w_{i+1,j+1} + w_{i-1,j+1} \\ & \left. - w_{i-1,j-1}) + \zeta (w_{i,j-1} - 2w_{i,j} + w_{i,j+1}) \right] = 0 \text{ ------} [2] \end{aligned}$$

In eq. [2], the subscripts i,j refer to the grid point indexes in the x,y directions, respectively. The term Δ is the width between two grid points in the finite difference mesh, as shown in Fig. 1a. The derivation of equation [2] is developed in Appendix C.

Applying eq. [2] to all MxN internal grid points and including boundary conditions, the problem reduces to solving the following matrix equation:

$$AW + N_x BW = 0 \qquad [3]$$

A and B are square matrices of order (MN x MN), while W is a column vector of order (MN). Matrix A contains the finite difference coefficients relating to plate geometry, boundary conditions, and material properties; matrix B is composed of the loading condition coefficients. W contains the relative lateral displacements of all MN grid points.

Many numerical methods are available to solve eq. [3]; the method chosen (2) recasts equation [3] as:

$$GW = \frac{1}{N_x} W \qquad [4]$$

where $$G = -A^{-1}B \qquad [5]$$

Equation [4] is solved by an iteration process, which gives both N_x and the shape of the associated buckling mode when the process converges. To begin the solution, an initial trial vector, W_0 , is assumed and a new vector, W_1 , is computed. The relationship between W_0 and W_1 is $W_1 = GW_0$; in general, the k and k+1 iterations are related by

$$W_{k+1} = GW_k \qquad [6]$$

A scalar ratio, $s_{k,\alpha}$ is obtained as $s_{k,\alpha} = \dfrac{w_{k,\alpha}}{|\mathrm{Max}\ (w_{k,}\ \alpha)|}$.

The process converges when $\left| \dfrac{s_{k,\alpha} - s_{k-1,\alpha}}{s_{k-1,\alpha}} \right| \le \varepsilon$ for all

α, $1 \leq \alpha \leq MN$. ε is the convergence tolerance. The final value of the scalar ratio at convergence is S and is defined as $S = \frac{\text{Max } (w_{k+1,\alpha})}{\text{Max } (w_{k,\alpha})}$. This gives the elastic buckling load and associated plate buckling coefficient as:

$$N_x = \frac{1}{S} \qquad [7]$$

and

$$K = \frac{N_x b^2}{\pi^2 D} \qquad [8]$$

For the case of applied shear only, equation [7] gives N_{xy}, instead of N_x; eq. [8] then uses N_{xy} to compute K.

This iterative process can be shown to converge to the largest value of S and therefore the smallest buckling load (2). The initial trial vector is a unit vector ($W_0 = 1$). Secondary modes influenced the solution for the case of pure shear enough to create convergence problems. To overcome this without using an excessive amount of iteration time, the Aitkens δ^2-extrapolation (2) was applied after every five iteration cycles. The procedure for applying the δ^2-extrapolation is detailed in Appendix D.

NUMERICAL RESULTS FOR ELASTIC CASES

The above algorithm was programmed using FORTRAN IV and run on the IBM 360-158 computer at the KSU Computation Center. Several sample problems were run to obtain a check on the program accuracy, to ascertain the effect of grid size on accuracy, and to gain the computing times. These results are given in Table 1. For a 5x5 grid network (the minimum size allowed in the program), the maximum computation time is less than one and one-half seconds. The times for the finest square mesh (11x11) are about ninety seconds. As for accuracy, the largest difference between a program value and one given in the literature (1,3) when the most accurate mesh (11x11) is used is 3.9%. This occurs for a plate subjected to pure shear with two opposite edges hinged and the other two clamped.

APPLICATION TO REINFORCED CONCRETE

The loads and notations for the reinforced concrete plate are shown in Fig. 1. Assuming the tangent modulus theory, small deflection theory, isotropic behavior, no shear interaction between the steel and concrete, and the concrete does not crack, the governing equation (4) is:

$$K_s \left(\frac{\partial^4 w}{\partial x^4} + \frac{\partial^4 w}{\partial y^4} \right) + K_c \left(\frac{\partial^4 w}{\partial x^4} + 2\beta \frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4} \right)$$

$$+ N_{x_c} \left(\frac{\partial^2 w}{\partial x^2} + 2\xi \frac{\partial^2 w}{\partial x \partial y} + \zeta \frac{\partial^2 w}{\partial y^2} \right) = 0 \quad \text{----------------[9]}$$

In this equation, $K_s = E_s h \sum_{n=1}^{KSTL} \rho_n \bar{z}_n^2$ ----------------------[9a]

$$K_c = \frac{\tilde{E}_c h^3}{12(1-\nu_c^2)} \quad \text{--------------------------------[9b]}$$

β is the shear coefficient, $0 \leq \beta \leq 1$, and N_{x_c} is the buckling load. In eq. [9a], E_s is the steel modulus of elasticity, h is the plate thickness, ρ_n is the steel ratio of the n-th layer of steel, $\bar{z}_n$ is the distance from the center of the plate to the centroid of the n-th layer, and KSTL is the number of layers of steel reinforcing. In eq. [9b], $\tilde{E}_c$ is the concrete tangent modulus at the stress level at the onset of buckling, and

ν_c is Poisson's ratio for concrete. Eq. [9] can be written as:

$$(K_s + K_c)\frac{\partial^4 w}{\partial x^4} + 2\beta K_c \frac{\partial^4 w}{\partial x^2 \partial y^2} + (K_s + K_c)\frac{\partial^4 w}{\partial y^4}$$

$$+ N_{x_c}\left(\frac{\partial^2 w}{\partial x^2} + 2\xi \frac{\partial^2 w}{\partial x \partial y} + \zeta \frac{\partial^2 w}{\partial y^2}\right) = 0 \text{ ------------------[10]}$$

Expressing eq. [10] in finite difference form (2) yields:

$$\left(\frac{K_s + K_c}{\Delta^2}\right)(w_{i-2,j} - 4w_{i-1,j} + 12w_{i,j} - 4w_{i+1,j} + w_{i+2,j}$$

$$+ w_{i,j-2} - 4w_{i,j-1} - 4w_{i,j+1} + w_{i,j+2}) + \frac{2\beta K_c}{\Delta^2}(w_{i-1,j-1}$$

$$- 2w_{i,j+1} + w_{i+1,j-1} - 2w_{i-1,j} + 4w_{i,j} - 2w_{i+1,j}$$

$$+ w_{i-1,j+1} - 2w_{i,j+1} + w_{i+1,j+1}) + N_{x_c}\ [\ \frac{\xi}{2}\ (-w_{i-1,j-1}$$

$$+ w_{i+1,j-1} + w_{i-1,j+1} - w_{i+1,j+1}) + \zeta\ (w_{i,j-1} + w_{i,j+1})$$

$$+ w_{i-1,j} + w_{i+1,j} - 2(1 + \zeta)w_{i,j}\ ] = 0 \text{ ---------------[11]}$$

Applying eq. [11] to all internal MN grid points and including boundary effects, the problem takes the form of equation [3], ie. $AW + N_{x_c} BW = 0.$

The difference between the elastic buckling case and reinforced concrete is that the material properties for concrete are not independent of the buckling load; $\tilde{E}_c$ is related to the buckling load by:

$$\tilde{E}_c = \frac{1.7f'_c}{\varepsilon_0}\left[1- \frac{N_{x_c}}{0.85f'_c h}(1 + 3\xi^2 + \zeta^2)^{\frac{1}{2}}\right]^{\frac{1}{2}} \text{ ------------[12]}$$

In the above equation, f'_c and ε_0 are the concrete cylinder strength and the strain at which it is attained, respectively.

The derivation of equation [12] from $\tilde{E}_c = \left.\frac{df_e}{d\varepsilon_e}\right|_{\varepsilon_{e_c}}$

using the octahedral shear theory (1) and the Madrid parabolic stress-strain relationship (4) is shown in Appendix E.

The plate buckling coefficient, K , can be shown to be independent of the concrete material properties. Substituting the relationship for K_c from eq. [9b] into $N_{x_c} \simeq \frac{K\pi^2 K_c}{b^2}$ yields:

$$N_{x_c} \simeq \frac{K\pi^2 h^3}{12b^2(1-\nu_c^2)} \tilde{E}_c \text{ ------------------------------[13]}$$

Equation [13] is exact for plates with no steel reinforcing, and a good approximation for typical steel ratios. The development of K and the justification of eq. [13] are presented in Appendix F. Putting eq. [13] into eq. [12] and reducing gives:

$$\left(\frac{\varepsilon_0}{1.7f'_c}\right)^2 \tilde{E}_c^2 + C\tilde{E}_c - 1 = 0 \text{ --------------------[14]}$$

where

$$C = \frac{K\pi^2 h^2 (1 + 3\xi^2 + \zeta^2)^{\frac{1}{2}}}{0.85f'_c 12b^2(1-\nu_c^2)} \text{ ---------------------[14a]}$$

For the case of pure shear, eq. [14a] is replaced by

$$C = \frac{\sqrt{3}\, K\pi^2 h^2}{0.85f'_c 12b^2(1-\nu_c^2)} \text{ -------------------------[14b]}$$

Solving eq. [14] by the quadratic formula, the expression for $\tilde{E}_c$ is:

$$\tilde{E}_c = \frac{\left[C^2 + 4\left(\frac{\varepsilon_0}{1.7f'_c}\right)^2 \right]^{\frac{1}{2}} - C}{2\left(\frac{\varepsilon_0}{1.7f'_c}\right)^2} \text{ ------------------[15]}$$

Equation [3] and eq. [12] must be solved simultaneously. The method of solution involves a double iteration process. The plate is first assumed to buckle elastically; eq. [3] is solved for K, using the method developed for elastic buckling.

$\tilde{E}_{c_0}$ is taken as the initial secant modulus for concrete, given by the formula: $E_c = 33\gamma^{\frac{3}{2}}\sqrt{f'_c}$, where γ is the unit weight of concrete, ie. $\tilde{E}_{c_0} = E_c$. Knowing K, eq. [15] is solved for $\tilde{E}_{c_1}$. In general, if the following inequality is true, the iteration on $\tilde{E}_c$ using equation [15] has converged. The inequality is $\left| \dfrac{\tilde{E}_{c_m} - \tilde{E}_{c_{m-1}}}{\tilde{E}_{c_{m-1}}} \right| \leq \varepsilon$. If $> \varepsilon$, $\tilde{E}_{c_m}$ is used to obtain K_{c_m} and the process repeated until it converges. As a further check, the buckling load, $N_{x_{c_m}}$, is substituted into eq. [12] to obtain a new value for $\tilde{E}_{c_m}$. If this value also satisfies the convergence tolerance, the problem is solved. If it does not, the iteration is repeated until the tolerance check is satisfied. With both iterative series converged (the outer one on $\tilde{E}_c$ and the inner one on W), $N_{x_c} = \frac{1}{S}$, or in the case of pure shear, $N_{xy_c} = \frac{1}{S}$.

Since the steel reinforcing increases the axial stiffness of the plate in addition to the lateral stiffness, the total axial load, P_c , is the sum of the loads carried by the concrete and by the steel at the common strain, ε_{e_c} . ε_{e_c} is computed from the concrete stress-strain law (see Appendix E).

COMPUTER SOLUTION TO REINFORCED CONCRETE PLATE BUCKLING

The solution algorithm was programmed in FORTRAN IV and implemented on an IBM 360-158 computer. The flow chart is diagrammed in Fig. 2 and the entire program is listed in Appendix G. There are five subprograms in the program. The finite difference coefficients for matrices A and B are computed in Subroutine MATAB. Matrix is inversed in Subroutine MATINV. The δ^2-extrapolation is done by Subroutine DELSQ, while the solution of eq. [15] and the tolerance check on $\tilde{E}_c$ are performed in Subroutine TANMOD. The calculation of ε_{e_c} and P_c are handled by Subroutine PCRIT. The convergence tolerance used in the program is $\varepsilon = 0.01$.

For the first iteration cycle on $\tilde{E}_c$, the initial trial vector is $W_{0,0} = 1$. For each succeeding iteration on $\tilde{E}_c$, the initial trial vector is the converged final vector from the preceding cycle, $W_{0,m+1} = W_{k,m}$. This technique saves much computing time, since the buckled shape of the plate changes little from one cycle on $\tilde{E}_c$ to the next.

As the program stands, the applied forces must be uniformly distributed and the edges are either hinged or clamped. The program could handle uniformly varying loads and bending loads with some minor changes. Modifications to the basic program to handle other boundary conditions and / or uniformly varying loads and bending effects would be made in Subroutine MATAB.

NUMERICAL RESULTS FOR REINFORCED CONCRETE

Several sample problems were run. The results are given in Table 2. Swartz and Rosebraugh (4) give a design formula for simply-supported reinforced concrete plates in uni-axial compression. Comparing the program with the design formula (4) shows that the program accurately predicts both the concrete stress and the total buckling load. The program considers the effect of the steel on both the buckling behavior and the axial stiffness.

The mesh size has relatively little effect on the accuracy of the solution. The cause is that the relationship between $\tilde{E}_c$ and N_{x_c}, given by eq. [12], is not linear but parabolic. At the stress levels at the onset of buckling, a relatively large change in K, and therefore N_{x_c}, creates a small change in $\tilde{E}_c$. For plate 2, a typical shear wall, the least accurate (5x5) mesh gives K = 10.89, while the more accurate 11x11 mesh yields K = 9.66, a difference of about 13%; yet the values for N_{x_c} are within 0.4% of each other for a plate subjected to pure shear, a loading condition which gives the largest errors for K.

The program predicts only the buckling load. For plates with shear loadings, the concrete failure may be caused by either buckling or diagonal tension cracking, depending upon the plate geometry. Several plates with identical properties, except for the thickness, were analyzed under the effect of pure shear to determine whether buckling or diagonal tension controls. The information is given in Table 2. All plates thicker than 0.5 inches (1.3 cm) would fail in diagonal tension before buckling could occur. The failure criteria was that the maximum principal stress be less than f_t' for the plate to buckle: $f_{p_{max}} \leq f_t' = 7.5\sqrt{f_c'}$, a reasonably conservative value. The limiting $\frac{b}{h}$ ratio was determined to be 240. Above this value, buckling controls; below it, the plate develops diagonal tension cracks which lead to failure.

Typical shear walls will have a $\frac{b}{h}$ ratio between 30 and 60; with these proportions buckling is not a factor, and the design is controlled by diagonal tension.

Lateral deflections at the onset of buckling are also given by the program. Deflection contours for some sample plates are given in Figs. 3, 4, and 5. The deflections are primarily useful for testing the program, but they would be useful in developing an analytical method for solving equation [3] directly.

CONCLUSIONS

The results of a computer program for predicting the buckling of plates subjected to uniformly-distributed in-plane forces with several edge conditions are presented in this report. Both elastic buckling and buckling in reinforced concrete are considered.

The conclusions drawn from this study are:

1) The elastic buckling cases are predicted accurately and economically. Using the finest mesh sizes, the program gives results within 4% of those predicted by the

literature (1,3). The program will also predict cases which have not been analyzed in the literature.

2) For reinforced concrete plates, the program predicts both the stress in the concrete and the total buckling load with good accuracy. The effect of the steel reinforcing on the axial stiffness of the plate is included. The mesh size has little effect on the solution; the coarser spacings give acceptable results with a minimum of execution time and associated cost.

3) The program predicts only buckling loads. Whether the plate will fail in buckling, or in diagonal tension is not predicted. Plates without shear loads fail in buckling. The limiting $\frac{b}{h}$ ratio for plates with shearing forces is 240. Below this value, diagonal tension controls with the maximum tension permitted, f'_t , equal to $7.5\sqrt{f'_c}$. Plates with typical dimensions will fail due to formation of diagonal tension cracks, not due to buckling.

4) The buckled shape of the plate is also given. These deflections could be used to create an analytical expression to solve the buckling problem directly.

APPENDIX A: NOTATION

a, b, h = length, width, thickness, respectively, of the plate

A, B = matrices containing coefficients for plate geometry and properties, loading conditions, respectively

D = elastic plate stiffness coefficient

E, ν = modulus of elasticity and Poisson's ratio, respectively, for elastic buckling

E_c = $33\gamma^{\frac{3}{2}}\sqrt{f'_c}$; γ is the unit weight of concrete

$\tilde{E}_c$ = tangent modulus of elasticity for concrete for isotropic behavior

E_s = modulus of elasticity for steel

e_e, e_{e_c} = nondimensionalized strain in concrete prior to buckling and at buckling, respectively

f'_c, ε_0 = strength of concrete cylinder and strain at which it is attained, respectively

f'_t = maximum tensile strength of concrete, set equal to $7.5\sqrt{f'_c}$

G = $-A^{-1}B$

k, m = subscripts denoting iteration on W, and $\tilde{E}_c$, respectively

K = elastic buckling coefficient

K_c, K_s = plate stiffness coefficients for concrete and steel reinforcing, respectively

KSTL = number of layers of steel reinforcing in the plate

M, N = number of internal grid points in the x, y directions, respectively

N_x = buckling load per unit length for elastic cases

N_{xy} = ξN_x, ξ assumed to be constant

N_y = ζN_x, ζ assumed to be constant

N_{x_c} = concrete contribution to buckling load for reinforced plates

P_c = total plate buckling load, including the effects of both concrete and steel

S = largest eigenvalue for the iteration on W, equals $1/N_{x_c}$

w = normal plate displacement

W = column vector containing lateral displacements of the finite difference grid points

x, y, z = plate coordinate directions

$\bar{Z}_n$ = distance from centroid of n-th layer of steel to middle surface of the plate, assumed same in both coordinate directions

α = subscript denoting an individual element in W

β = shear coefficient

Δ = width between two finite difference grid points, same in both coordinate directions

ε = convergence tolerance

ν_c = Poisson's ratio for plain concrete

ρ_n = steel ratio for n-th layer, same in both coordinate directions

APPENDIX B: TABLES AND FIGURES

Table 1 - Elastic Buckling Coefficients for Rectangular Plates with Various Support and Load Conditions (a)

Plate Dimensions and Finite Difference Grid		Poisson's Ratio	Boundary Conditions (c)	Load Ratios		Elastic Buckling Coefficient K		Percent Difference in K	Execution Time on IBM 360/158 Seconds
a x b	M x N (b)			$\frac{\tau_{xy}}{\sigma_x}$	$\frac{\sigma_y}{\sigma_x}$	Prog.	Ref. 1,3		
18 x 18	11 x 11	1/3	SS	0	0	3.99	4.00	0.4	87.3
24 x 24	5 x 5	1/3	SS	0	0	3.91	4.00	2.2	1.2
24 x 24	11 x 11	1/4	SS	0	1	1.99	2.00	0.6	89.5
24 x 24	11 x 11	1/4	SS	(d)	0	9.66	9.34	3.4	90.0
42 x 12	20 x 5	1/3	SS	(d)	0	6.12	5.80	5.5	52.4
40 x 40	5 x 5	1/6	SS	(d)	0	10.89	9.34	16.6	1.3
40 x 40	11 x 11	1/6	SS	4	1	1.36	(e)	--	89.2
32 x 32	5 x 5	1/3	SS	4	0	1.91	1.75	9.1	1.3
32 x 32	11 x 11	1/3	SS	4	0	1.78	1.75	1.7	87.4
96 x 48	11 x 5	1/6	SS	(d)	0	7.99	6.60	21.1	22.0
96 x 48	15 x 7	1/6	SS	(d)	0	6.81	6.60	3.1	60.1
96 x 48	11 x 5	1/6	SS	0	0	3.94	4.00	1.5	22.5
18 x 18	5 x 5	1/4	FF	0	0	9.07	10.07	9.9	1.3
18 x 18	11 x 11	1/4	FF	0	0	9.71	10.07	3.6	88.7
24 x 24	5 x 5	1/4	SF	0	0	6.37	6.74	5.5	1.2
24 x 24	11 x 11	1/4	SF	0	0	6.65	6.74	1.3	86.1
24 x 24	11 x 11	0	SF	0	0	6.65	6.74	1.3	90.0
24 x 24	11 x 11	1/4	SF	(d)	0	12.76	12.28	3.9	95.3

a. For all plates, Young's Modulus = 10×10^6 psi (6.9×10^7 kN/$_m$2) and thickness = 1.0 in. (2.5 cm). Plate dimensions in inches (1 in. = 2.54 cm)

b. M,N = grid points in the directions parallel to a,b, respectively

c. SS = all edges have simple supports, FF = all edges are clamped, SF = edges at y = 0, b have simple supports, edges at x = 0, a are fixed

d. Shear load only

e. No value for K is listed in the reference (2)

Table 2 - Buckling Loads for Simply-supported Reinforced Concrete Plates

Plate No. (a)	h inch.	Load Ratios $\frac{\tau_{xy}}{\sigma_x}$	Load Ratios $\frac{\sigma_y}{\sigma_x}$	Buckling Loads kips P_{x_c} (b) Prog.	Buckling Loads kips P_{x_c} (b) Ref. 4	Buckling Loads kips P_{xy_c}	Buckling Loads kips P_{y_c} (b)	Compressive Stress σ_x psi Prog.	Compressive Stress σ_x psi Ref. 4	Diagonal Tension Failure
1	1	(c)	0	0	(d)	68.3	0	1424(e)	(d)	Yes
1	1	0	0	95.5	94.0	0	0	1937	1906	No
2.1	3	(c)	0	0	(d)	525.2	0	1459(e)	(d)	Yes
2.2	3	(c)	0	0	(d)	527.2	0	1464(e)	(d)	Yes
2.2	2	(c)	0	0	(d)	341.3	0	1422(e)	(d)	Yes
2.2	1	(c)	0	0	(d)	124.9	0	1040(e)	(d)	Yes
2.2	3/4	(c)	0	0	(d)	67.9	0	755(e)	(d)	Yes
2.2	1/2	(c)	0	0	(d)	24.5	0	408(e)	(d)	No
2.2	1/4	(c)	0	0	(d)	3.5	0	115(e)	(d)	No
2.1	3	4	0	282	(d)	516.0	0	358	(d)	Yes
2.1	3	0	1	652	(d)	0	652	1433	(d)	No
2.1	3	0	0	911	904	0	0	2233	2225	No
2.1	3	4	1	274	(d)	505	274	351	(d)	Yes

1 kip = 4.45 kN, 1 psi = 6.9 kN/m^2, 1 inch = 2.54 cm

a. For all plates: f'_c = 3000 psi (20700 kN/m^2), ν_c = 0.17

$E_s = 29 \times 10^6$ psi (20×10^7 kN/m^2)

For Plate No. 1: a = 96 in, b = 48 in (244 cm x 122 cm)
$\varepsilon_0 = 1.924 \times 10^{-3}$ in/in (cm/cm)
ρ = 0.002, $\overline{Z}$ = 0.375 in (0.953 cm)
mesh size = 11 x 5

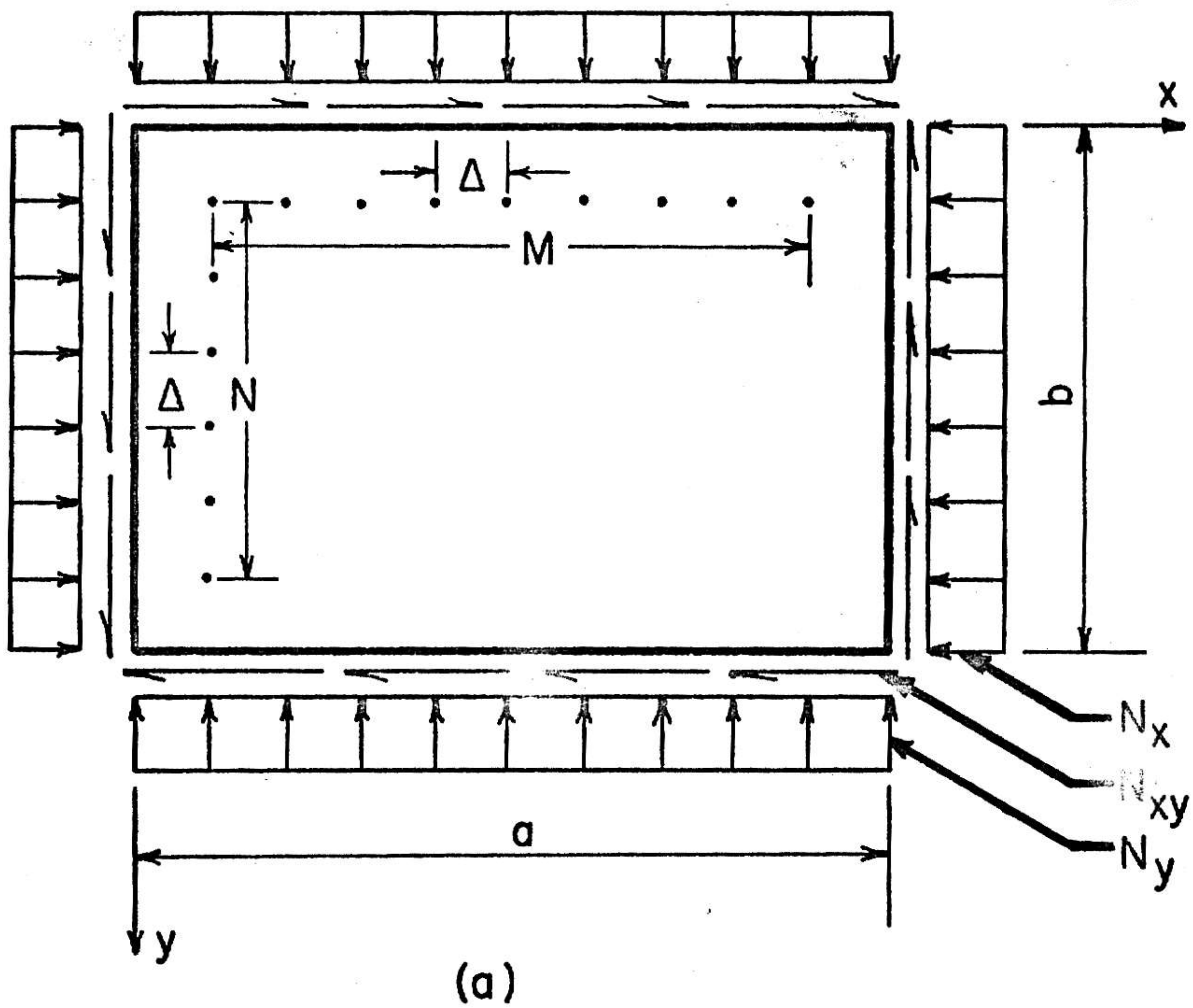
For Plate No. 2: a = 120 in, b = 120 in (305 x 305 cm)
$\varepsilon_0 = 1.7 \times 10^{-3}$ in/in (cm/cm)
ρ = 0.01, $\overline{Z}$ = 0.0
Plate 2.1 has a mesh of 11 x 11
Plate 2.2 has a mesh of 5 x 5

b. Includes the effect of the steel on the axial stiffness

c. Applied shear load only

d. No value is given in the reference for this loading combination

e. Shearing stress, τ



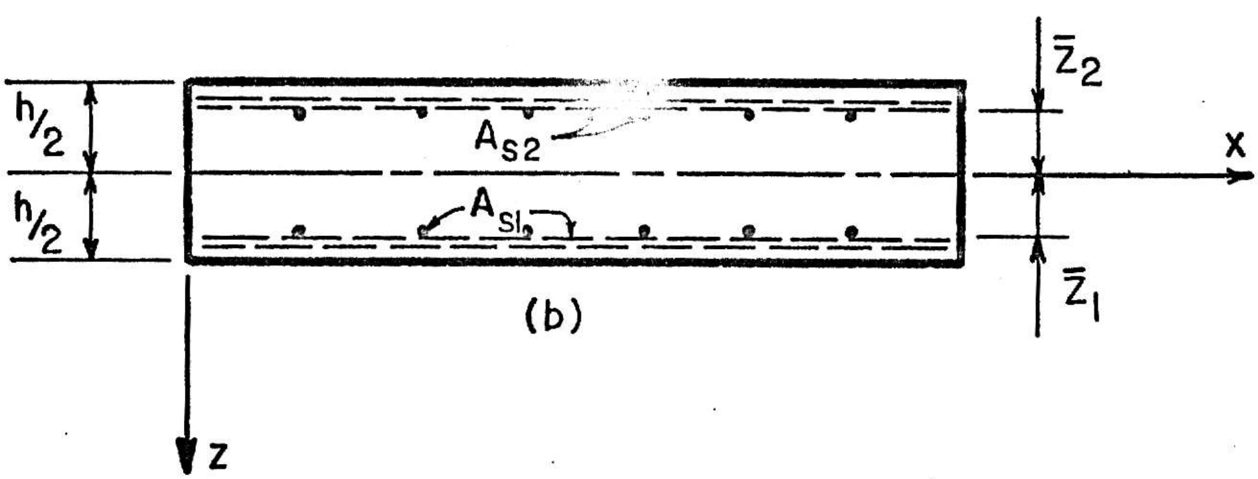


Fig. 1 a) Plate Subjected to Distributed In-plane Forces with Coordinate System and Finite Difference Grid

b) Cross-section showing Steel Reinforcing

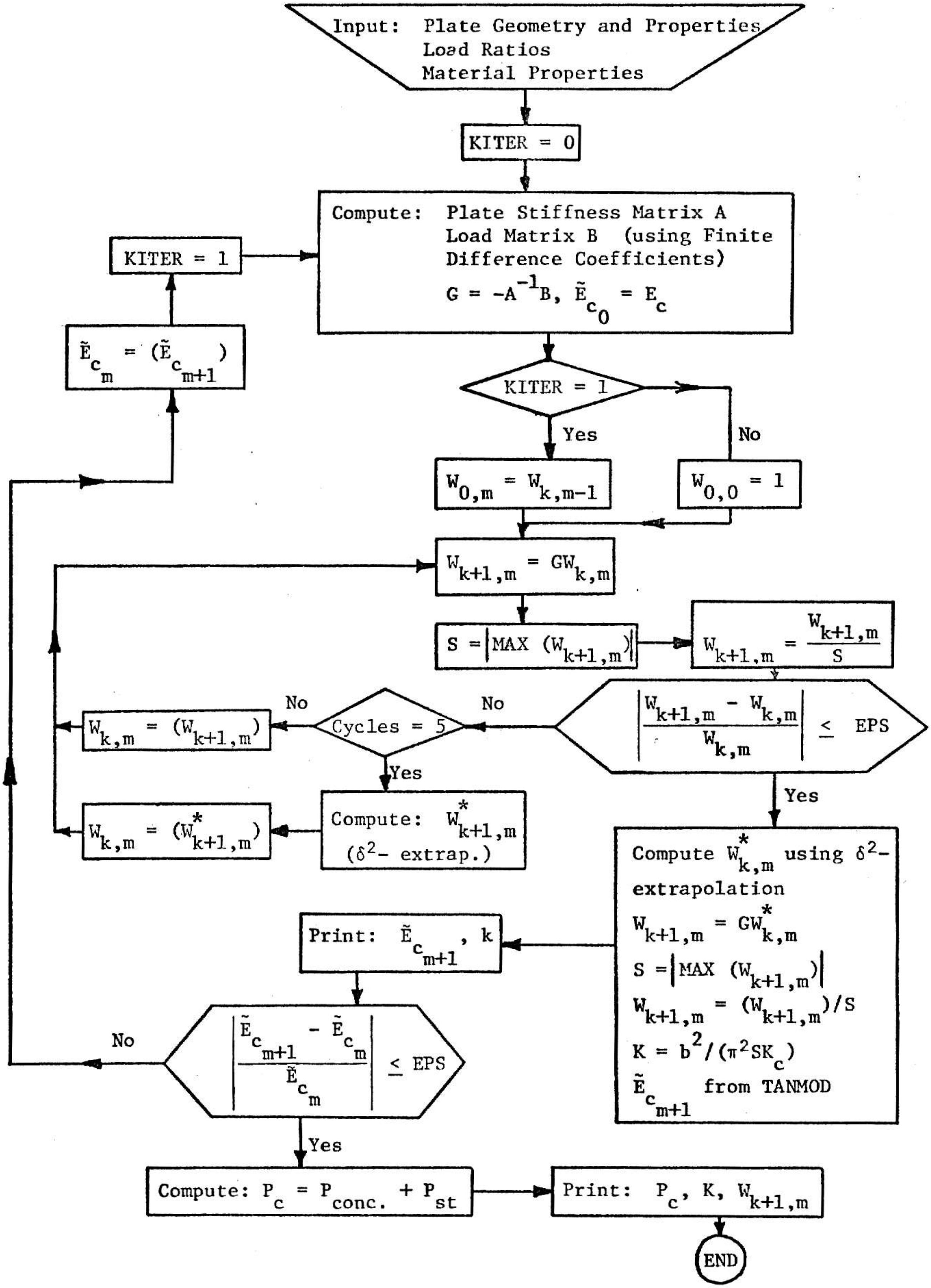


Fig. 2 Reinforced Concrete Plate Buckling Program Flow Chart

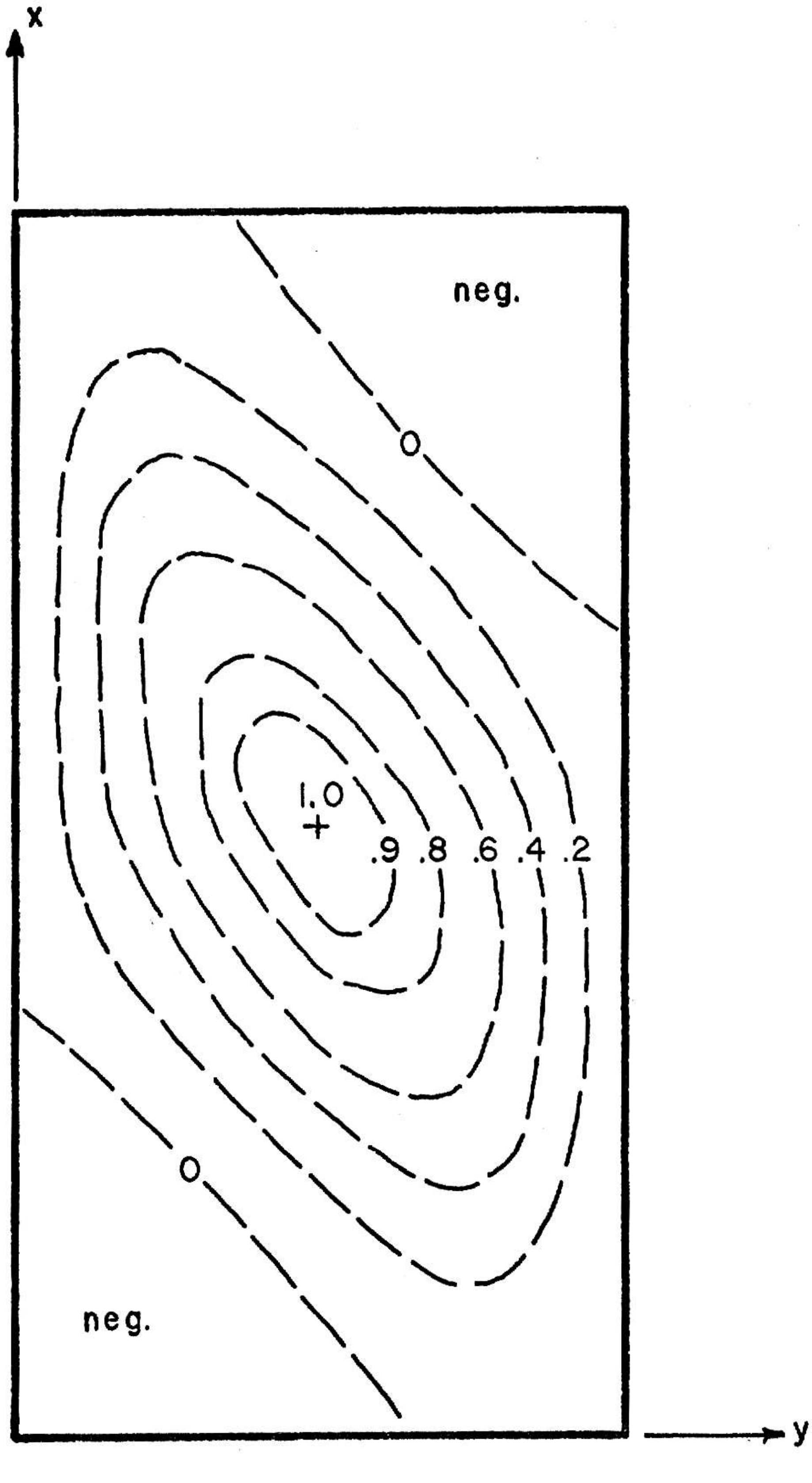


Fig. 3 Deflection Contours for a Plate with an $\frac{a}{b}$ ratio of 2 Subjected to Shearing Load only

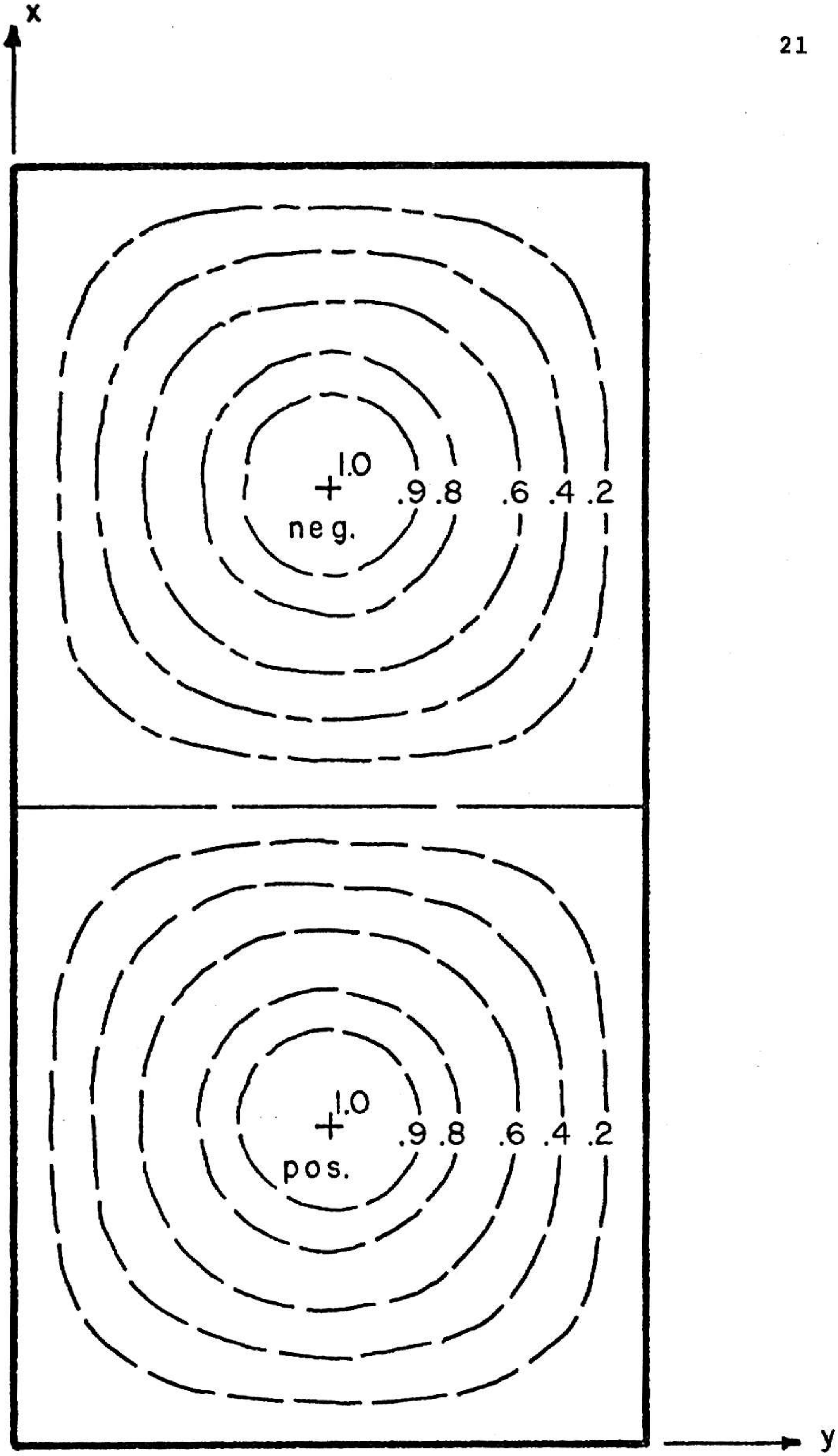


Fig. 4 Deflection Contours for a Plate with an $\frac{a}{b}$ ratio of 2 in Uni-axial Compression

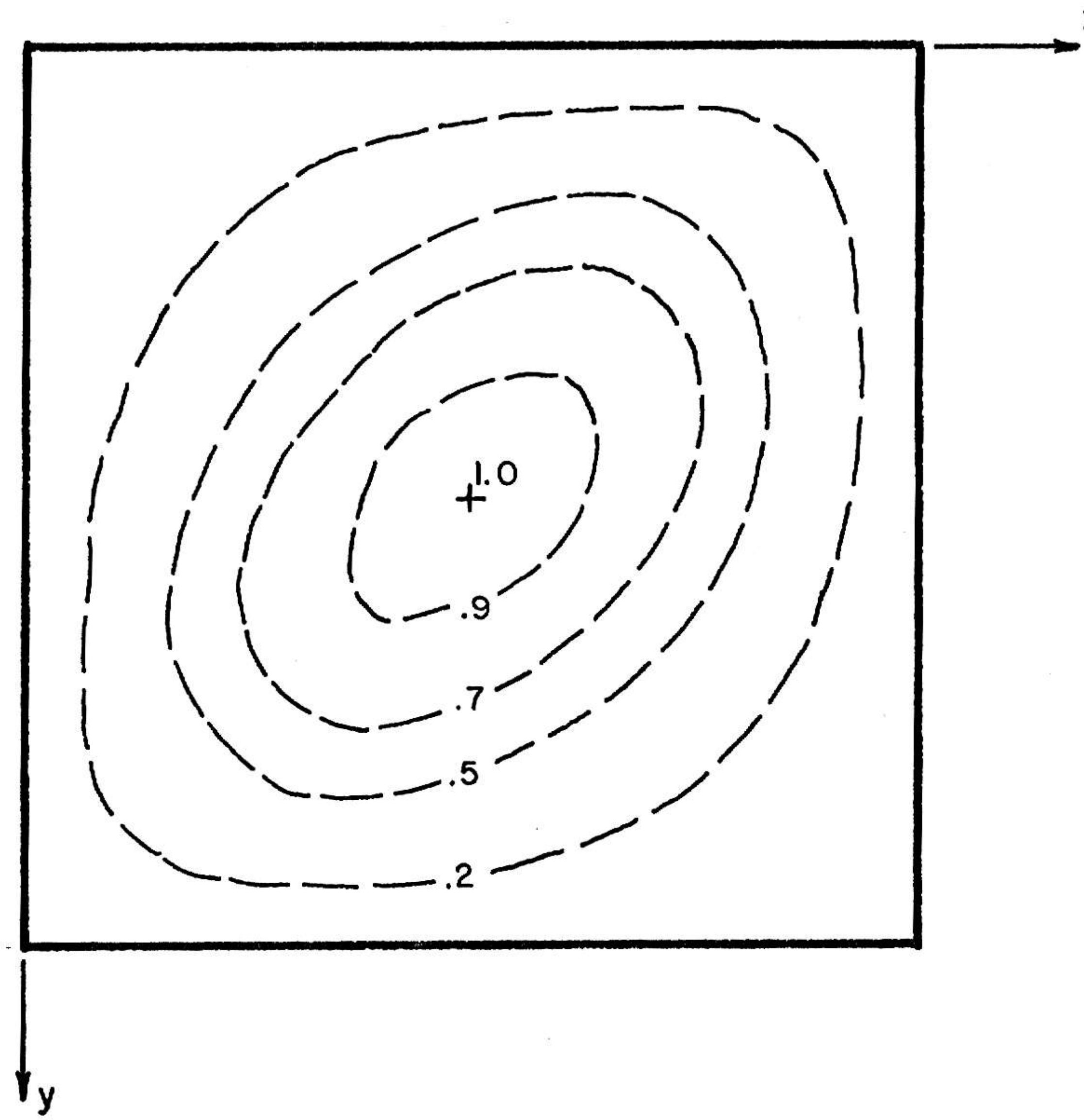


Fig. 5 Deflection Contours for a Square Plate Subjected to Bi-axial Compression plus Shear:

$\frac{\tau_{xy}}{\sigma_x} = 4, \quad \frac{\sigma_y}{\sigma_x} = 1$

APPENDIX C: FINITE DIFFERENCE DEVELOPMENT

The following assumptions are made in the derivation of the finite difference equations:

1) the function for the deflections has continous partial derivatives

2) the function can be represented by a Taylor's series

3) the nodal points are separated by a distance Δ in both directions (See Fig. 1a)

4) the truncation error is small compared to the main terms

Neglecting the error of truncating, the equations for the needed partial derivatives are (1):

$$\frac{\partial^2 w}{\partial x^2} = \frac{1}{\Delta^2}(w_{i-1,j} - 2w_{i,j} + w_{i+1,j}) \quad \text{---------------[C-1]}$$

$$\frac{\partial^2 w}{\partial y^2} = \frac{1}{\Delta^2}(w_{i,j-1} - 2w_{i,j} + w_{i,j+1}) \quad \text{---------------[C-2]}$$

$$\frac{\partial^2 w}{\partial x \partial y} = \frac{1}{\Delta^2}(4w_{i+1,j+1} - 4w_{i+1,j-1} + 4w_{i-1,j-1} - 4w_{i-1,j+1}) \quad \text{[C-3}$$

$$\begin{aligned}\nabla^4 w = \frac{1}{\Delta^4}(&w_{i,j-2} + 2w_{i-1,j-1} - 8w_{i,j-1} + 2w_{i+1,j-1} \\ &+ w_{i-2,j} - 8w_{i-1,j} + 20w_{i,j} - 8w_{i+1,j} + w_{i+2,j} \\ &+ 2w_{i-1,j+1} - 8w_{i,j+1} + 2w_{i+1,j+1} + w_{i,j+2}) \quad \text{----[C-4]}\end{aligned}$$

Starting from the governing equation of equilibrium, equation [1]: $\nabla^4 w = -\frac{1}{D} N_x\left(\frac{\partial^2 w}{\partial x^2} + 2\xi \frac{\partial^2 w}{\partial x \partial y} + \zeta \frac{\partial^2 w}{\partial y^2}\right)$

and substituting the appropriate expression for the partial derivatives results in the following relationship:

$$\begin{aligned}\frac{1}{\Delta^4}(&w_{i,j-2} + 2w_{i-1,j-1} - 8w_{i,j-1} + 2w_{i+1,j-1} + w_{i-2,j} \\ &- 8w_{i-1,j} + 20w_{i,j} - 8w_{i+1,j} + w_{i+2,j} + 2w_{i-1,j+1} \\ &- 8w_{i,j+1} + 2w_{i+1,j+1} + w_{i,j+2}) =\end{aligned}$$

$$- \frac{1}{D} N_x (\frac{1}{\Delta^2}) [w_{i-1,j} - 2w_{i,j} + w_{i+1,j} + \frac{\xi}{2} (w_{i+1,j-1} - w_{i+1,j+1} + w_{i-1,j+1} - w_{i-1,j-1}) + \zeta (w_{i,j-1} - 2w_{i,j} + w_{i,j+1})] \quad \text{------------------------------------[C-5]}$$

Multipling eq. [C-5] through by Δ^4, and rearranging terms gives equation [2]:

$$w_{i,j-2} + 2w_{i-1,j-1} - 8w_{i,j-1} + 2w_{i+1,j-1} + w_{i-2,j} - 8w_{i-1,j} + 20w_{i,j} - 8w_{i+1,j} + w_{i+2,j} + 2w_{i-1,j+1} - 8w_{i,j+1} + 2w_{i+1,j+1} + w_{i,j+2}) + \frac{\Delta^2}{D} N_x [w_{i-1,j} - 2w_{i,j} + w_{i+1,j} + \frac{\xi}{2} (w_{i+1,j-1} - w_{i+1,j+1} + w_{i-1,j+1} - w_{i-1,j-1}) + \zeta (w_{i,j-1} - 2w_{i,j} + w_{i,j+1})] = 0 \quad \text{-----[2]}$$

APPENDIX D: δ^2-EXTRAPOLATION APPLICATION

For some modes of buckling, the iteration on W converges rapidly. For others, the converging process is too slow. The case of pure shear, for example, oscillates between two modes which very gradually converge together (after 100 iteration cycles, the process was not close to converged). Uniaxial compression, on the other hand, converges quickly (less than 20 cycles). To handle all cases and decrease the number of iterations needed, the Aitken's δ^2-extrapolation (1) is used.

The δ^2-extrapolation is based on the assumption that W_k is composed of a true value with one error component:

$$w_{\alpha,k} = t_\alpha + c(b)^k \quad \text{[D-1]}$$

where b and c are constants. The values for t, b, and c are unknown. Substituting three successive iterations, W_{k-2}, W_{k-1}, and W_k, into eq. [D-1], the resulting three simultaneous equations, when solved for T, yield:

$$t_\alpha = \frac{w_{\alpha,k-2}w_{\alpha,k} - w^2_{\alpha,k-1}}{w_{\alpha,k-2} - 2w_{\alpha,k-1} + w_{\alpha,k}} \quad \text{[D-2]}$$

If one mode predominates, eq. [D-2] gives a considerable advance towards the true values. In any case, eq. [D-2] gives a better approximation to the true values than the previous iterations. Since the process may or may not give the exact value due to the unknown qualities of the modal values, the extrapolation is used to generate a new starting matrix for another series of iteration. For this application, equation [D-2] is reformulated as:

$$w^{*}_{\alpha,k+1} = \frac{w_{\alpha,k-2}w_{\alpha,k} - w^{2}_{\alpha,k-1}}{w_{\alpha,k-2} - 2w_{\alpha,k-1} + w_{\alpha,k}} \quad \text{------------------[D-3]}$$

where k is an integer multiple of 5. The (*) merely indicates that this value is obtained from the extrapolation process.

This method reduces the total number of iteration cycles to an acceptable maximum of about 60 cycles. In addition, once the iterative process has converged, the technique is used to obtain an improved final vector. The closer the iterated values are to the true ones, the better the results will be using the δ^2-extrapolation procedure.

APPENDIX E: THE RELATIONSHIP BETWEEN N_{x_c} AND $\tilde{E}_c$

The three equations used to develop the relationship between N_{x_c} and $\tilde{E}_c$, equation [12], are:

$$\tilde{E}_c = \left.\frac{df_e}{d\varepsilon_e}\right|_{\varepsilon_{e_c}} \text{------------------------------------[E-1]}$$

$$f_{e_c} = \sigma_{x_c}(1 + 3\xi^2 + \zeta^2)^{\frac{1}{2}} \text{----------------------------[E-2]}$$

$$f_e = 0.85f'_c(2e_e - e_e^2) \text{------------------------------[E-3]}$$

where $e_e = \dfrac{\varepsilon_e}{\varepsilon_0}$.

The first equation is the definition of the tangent modulus, the second equation comes from applying the octahedral shear theory (1), and the last one is the Madrid parabolic stress-strain law modified for use in flexure (4).

Taking the derivative of eq. [E-3] with respect to ε_e:

$$\frac{df_e}{d\varepsilon_e} = \frac{0.85f'_c}{\varepsilon_0}(2 - 2e_e) = \frac{1.7f'_c}{\varepsilon_0}(1 - e_e)$$

Therefore: $$\tilde{E}_c = \frac{1.7f'_c}{\varepsilon_0}(1 - e_e) \text{----------------------[E-4]}$$

Solving eq. [E-3] for e_e leads to:

$$e_e = \frac{2 - \sqrt{4 - \dfrac{4f_e}{0.85f'_c}}}{2} = 1 - (1 - \frac{f_e}{0.85f'_c})^{\frac{1}{2}}$$

At the strain level e_{e_c} this becomes:

$$1 - e_{e_c} = (1 - \frac{f_{e_c}}{0.85f'_c})^{\frac{1}{2}} \text{-----------------------[E-5]}$$

Substituting eq. [E-5] into eq. [E-4] yields:

$$\tilde{E}_c = \frac{1.7f'_c}{\varepsilon_0}\left(1 - \frac{f_{e_c}}{0.85f'_c}\right)^{\frac{1}{2}} \text{------------------------------[E-6]}$$

Taking the relationship for f_{e_c} from eq. [E-2] and noting that $\sigma_{x_c} = \frac{N_{x_c}}{h}$, equation [E-6] becomes eq. [12]:

$$\tilde{E}_c = \frac{1.7f'_c}{\varepsilon_0}\left(1 - \frac{N_{x_c}\sqrt{1 + 3\xi^2 + \zeta^2}}{0.85f'_c h}\right)^{\frac{1}{2}} \text{-----------------[12]}$$

To determine the total axial load, P_c, ε_{e_c} must be found. Since N_{x_c} is known, ε_{e_c} can be computed by:

$$\varepsilon_{e_c} = \varepsilon_0\left[1 - \left(1 - \frac{N_{x_c}\sqrt{1 + 3\xi^2 + \zeta^2}}{0.85f'_c h}\right)^{\frac{1}{2}}\right] \text{--------------[E-7]}$$

For the case of pure shear, N_{x_c} is replaced by N_{xy_c} and the quantity under the $\sqrt{\ }$ sign is replaced by $\sqrt{3}$.

APPENDIX F: THE ELASTIC BUCKLING COEFFICIENT

The following assumptions are made in developing the elastic buckling coefficient, K, for concrete plates:

1) $w = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} a_{mn} \sin \frac{m\pi x}{a} \sin \frac{n\pi y}{b}$

2) the plate is simply supported

3) $\eta = \frac{\tilde{E}_c}{E_c}$, or $K_c = \eta D$

For uni-axial compression, the governing equation is:

$$K_c \nabla^4 w + K_s \left(\frac{\partial^4 w}{\partial x^4} + \frac{\partial^4 w}{\partial y^4}\right) + N_{x_c} \frac{\partial^2 w}{\partial x^2} = 0 \text{ ------------[F-1]}$$

Since $\nabla^4 w = \left[\left(\frac{m\pi}{a}\right)^4 + 2\frac{m^2 n^2 \pi^2}{a^2 b^2} + \left(\frac{n\pi}{b}\right)^4\right] a_{mn}^4 \sum\sum \sin \frac{m\pi x}{a} \sin \frac{n\pi y}{b}$

$$\frac{\partial^4 w}{\partial x^4} + \frac{\partial^4 w}{\partial y^4} = \left[\left(\frac{m\pi}{a}\right)^4 + \left(\frac{n\pi}{b}\right)^4\right] a_{mn}^4 \sum\sum \sin \frac{m\pi x}{a} \sin \frac{n\pi y}{b}$$

$$\frac{\partial^2 w}{\partial x^2} = -\left(\frac{m\pi}{a}\right)^2 a_{mn}^2 \sum\sum \sin \frac{m\pi x}{a} \sin \frac{n\pi y}{b}$$

equation [F-1] becomes:

$$K_c\left[\left(\frac{m\pi}{a}\right)^4 + 2\frac{m^2 n^2 \pi^2}{a^2 b^2} + \left(\frac{n\pi}{b}\right)^4\right] a_{mn}^2 + K_s\left[\left(\frac{m\pi}{a}\right)^4 + \left(\frac{n\pi}{b}\right)^4\right] a_{mn}^2$$
$$= N_{x_c}\left(\frac{m\pi}{a}\right)^2 \text{ --------------------------------[F-2]}$$

For N_{x_c} to be a minimum, take n equal to 1, and all the a_{mn} equal to zero, except for a_{11}, which is taken equal to 1. Equation [F-2] then becomes:

$$N_{x_c} = \frac{\pi^2}{b^2} K_c \left(\frac{mb}{a} + \frac{a}{mb}\right)^2 + \frac{\pi^2}{b^2} K_s \left[\left(\frac{mb}{a}\right)^2 + \frac{1}{b^2}\right] \text{ ---------[F-3]}$$

With typical plates the term $1/_b 2$ is small compared with $\left(\frac{mb}{a}\right)^2$; without appreciable error, equation [F-3] is rewritten as:

$$N_{x_c} = \frac{\pi^2}{b^2} K_c \left(\frac{mb}{a} + \frac{a}{mb}\right)^2 + \frac{\pi^2}{b^2} K_s \left(\frac{mb}{a}\right)^2 \text{ -----------------[F-4]}$$

For plates with no steel reinforcing, or with the center of the steel at the center of the plate ($K_s = 0$), eq. [F-4] becomes:

$$N_{x_c} = K \frac{\pi^2}{b^2} \eta D \text{ ------------------------------------[F-4a]}$$

where $K = \left(\frac{mb}{a} + \frac{a}{mb}\right)^2$ and is called the elastic buckling coefficient. Analogous expressions may be developed for other loading and support conditions.

A square plate will be used as an illustrative example. $\frac{a}{b} = 1$; for N_{x_c} to be a minimum m must be chosen equal to 1. Therefore $\frac{mb}{a} = \frac{a}{mb} = 1$, and eq. [F-4] becomes:

$N_{x_c} = \frac{\pi^2}{b^2} (4K_c + K_s)$. For typical design steel ratios and steel placing, the values for K_s will lie in the range $0 \leq K_s \leq K_c$, with most of the values between $\frac{1}{4}$ to $\frac{1}{2} K_c$. With K_s equal to K_c, the error involved in neglecting K_s is 25%; for $K_s = \frac{1}{2} K_c$, the error is 13%. If the plate contains no steel, there is no error, of course.

As a computing aid, the error involved in this approximation is acceptable, since iterating on the values given by eq. [15] will always converge rapidly; the alternative is iterating using eq. [12]. Due to the nature of the relationship between N_{x_c} and $\tilde{E}_c$, using eq. [12] will seldom converge as rapidly as using eq. [15].

APPENDIX G: PROGRAM LISTING

THIS BOOK CONTAINS NUMEROUS PAGES THAT ARE CUT OFF

THIS IS AS RECEIVED FROM THE CUSTOMER

```
C      PROGRAM FOR SHEAR-COMPRESSION BUCKLING OF PLATES
C      KC=E*H**3/12./(1.-NU**2)
C      XS=-XSI/2.---XSI=NXY/NX
C      ZETA= NYC/NXC
C      DELS=DEL*DEL---DEL=B/(N+1)
C      BETA=WARPING PARAMETER
C      M=NO.OF POINTS X-DIRECTION
C      N=NO.OF POINTS Y-DIRECTION
C      MAX. NO. OF ITERATIONS =100
C      MAXIMUM M*N=121
C      IB=0 SHEAR ONLY,IB=1 SHEAR+COMPRESSION
C      FOR PURE SHEAR, INPUT XSI = 1.0
C             KBC IS THE BOUND. COND. CODE.  KBC=0 FOR ALL EDGES SIMPLY
C      SUPPORTED; =1 FOR ALL EDGES FIXED; AND =2 FOR SOME EDGES FIXED,
C      AND SOME SIMPLY SUPPORTED.      IF KBC=2, E--- =-1. FOR A SIMPLE
C      SUPPORT AND =+1. FOR A FIXED EDGE.  IF KBC=2 READ ETOP,ERT,EBOT,EL
C      KSTL = THE NUMBER OF STEEL LAYERS IN THE PLATE.
C
C
C
       DIMENSION A(121,121),B(121,122),W(121),W1(121),RHO(5),ZBAR(5)
       COMMON A,B,KC,KS,DELS,XS,BETA,ZETA,M,N,IB,ETOP,ERT,EBOT,ELT
       COMMON/BLOCK1/W,L
       COMMON/BLOCK2/ENEW,BKCOEF,CONST1,CONST3,EPS,E,KSTL
       COMMON/BLOCK4/CONST2,EPC,FPRIME,EST,BL,PCR,RHOT
       REAL KS,KC,NU
       KASE =0
 1000 KASE =KASE+1
C
       READ(5,2,END=1001) TH,M,N,DEL,KBC,KSTL,EST
     2 FORMAT(F10.4,2I5,F10.4,2I5,F20.0)
C
       WRITE(6,1) KASE
     1 FORMAT(1H1,32X,'INPUT DATA FOR CASE ',I3)
C
       IF(KBC.EQ.2) READ(5,3) ETOP,ERT,EBOT,ELT
     3 FORMAT(4F5.1)
C
       READ(5,12) XSI,ZETA,BETA,IB
    12 FORMAT(3F10.4,I5)
C
       PI=3.1415927
       AL=DEL*(FLOAT(M)+1.)
       BL=DEL*(FLOAT(N)+1.)
       ASPECT=AL/BL
       WRITE(6,13) TH,M,N,DEL,AL,BL,KBC,KSTL
    13 FORMAT(1H0,'PLATE GEOMETRY AND PROPERTIES',/ ,11X,'PLATE THICKNESS
      1 =',F6.2,'; M =',I3,'; N =',I3,'; DELTA =',F8.4,1H;,/ ,11X,'LENGTH
      2, A =',F7.2,'; WIDTH, B =',F7.2,'; KBC =',I2,'; KSTL =',I2)
       WRITE(6,14)
    14 FORMAT(1H0,'BOUNDARY CONDITIONS')
       L=M*N
       IF(KBC.NE.0) GO TO 30
       WRITE(6,5)
     5 FORMAT(1H ,10X,'ALL EDGES SIMPLY SUPPORTED.')
       ETOP=-1.
       ERT =-1.
       EBOT=-1.
       ELT =-1.
    30 IF(KBC.NE.1) GO TO 40
```

```
      WRITE(6,8)
    8 FCRMAT(1H ,10X,'ALL EDGES CLAMFEC.')
      ETOP=1.
      ERT =1.
      EBCT=1.
      ELT =1.
   40 IF(KBC.EC.2) WRITE(6,17) ETCP,ERT,EECT,ELT
   17 FORMAT(1H ,10X,'TOP EDGE=',F4.C,', RIGHT ECGE=',F4.0,', BOTTCM EDG
     1E=',F4.0,', LEFT EDGE=',F4.0)
      WRITE(6,15) XSI,ZETA,BETA,IB
   15 FORMAT(1H0,'LOAD FACTORS',/ ,11X,'NXY/NX, XSI =',F10.4,'; NYC/NX,
     1ZETA =',F10.4,'; BETA =',F6.2,'; IB =',I2)
C
      READ(5,18) E,NU,FPRIME,EPC
   18 FCRMAT(F20.0,3F1C.5)
C
      WRITE(6,19) E,NU,FPRIME,EPO
   19 FCRMAT(1HC,'MATERIAL PRCPERTIES',// ,6X,'CCNCRETE',12(1H*),'INITIA
     1L MODULUS =',1PE11.4,'; NU =',CPF8.4,7H; F'C =,F6.0,'; STRAIN, EPC
     2 =',1PE10.3)
      KS=C.
      IF(KSTL.EQ.0) GO TO 550
      WRITE(6,20) EST
   20 FORMAT(1H ,5X,'STEEL',15(1H*),'ELASTIC MCDULUS =',1PE11.4)
      SUM=0.0
      RHCT = C.C
C
      READ(5,21) (RHO(I),ZBAR(I),I=1,KSTL)
   21 FCRMAT(8F10.5)
C
      DC 50 I=1,KSTL
      WRITE(6,22) I,RHC(I),ZBAR(I)
   22 FORMAT(1H ,25X,'LAYER',I2,5X,'RHC =',F7.4,', ZBAR =',F6.3)
      RHCT = RHCT + RHC(I)
   50 SUM=SUM+RHC(I)*ZBAR(I)**2
      KS=EST*TH*SUM
  550 CCNST1=(EPC/(1.7*FPRIME))**2
      CCNST2=SQRT(1.0+3.0*XSI**2+ZETA**2)
      IF(IB.EQ.C) CCNST2=SQRT(3.0)
      CCNST3=TH**2*PI**2*CCNST2/(12.*(1.-NU**2)*C.85*FPRIME*BL**2)
      CCNST4=TH**3/(12.0*(1.0-NU**2))
      KITER=C
  600 KC=E*CCNST4
      XS=-XSI/2.
      DELS=DEL**2
      IF(KITER.EQ.1) GO TO 99
      DO 98 I=1,L
   98 W(I)=1.
   99 DC100 I=1,L
  1C0 W1(I)=C.
      DO101 I=1,L
      DO101 J=1,L
      A(I,J)=0.
  1C1 B(I,J)=0.
      IPASS=C
      CALL MATAB(IPASS)
      IPASS=1
      CALL MATINV(A,L,CARN,B)
      IF(CARN)222,222,223
  222 WRITE(6,6)
```

```
    6 FORMAT(1HC,56X,'MATRIX IS SINGLLAR')
      GC TC 1000
  223 JM=L+1
      DC50C I=1,L
      DC500 J=1,JM
  500 B(I,J)=C.
      CALL MATAB(IPASS)
      DO401 I=1,L
      DO402 K=1,L
      DO402 J=1,L
  402 W1(K)=-A(I,J)*B(J,K)+W1(K)
      DO401 K=1,L
      A(I,K)=W1(K)
  401 W1(K)=C.
      ITER=C
      KCUNT=C
      KFINAL=C
  200 IF(ITER.EC.100) GO TC 201
      DO102 I=1,L
  102 W1(I)=C.
      DC103 I=1,L
      DO103 J=1,L
  103 W1(I)=A(I,J)*W(J)+W1(I)
      TEMP=C.
  301 DO104 I=1,L
      IF(ABS(W1(I))-TEMP)1C4,1C4,105
  105 TEMP=ABS(W1(I))
      JJ=I
  104 CONTINUE
      S2=TEMP
      IF(S2.EQ.C.C) WRITE(6,23)
   23 FORMAT(1HO,' S2 EQUALS ZERC.  RESET TO 1.0 AND CONTINUE')
      IF(S2.EQ.C.0) S2=1.0
      EPS=1.E-2
      ITER=ITER+1
      DO1C8 I=1,L
      W(I)=W1(I)/S2
      B(ITER,I)=W(I)
  1C8 CONTINUE
      B(100,ITER) = S2
      IF(KFINAL.EC.1) GC TC 7CC
      KCUNT=KCUNT+1
      IF(KCUNT.NE.5)GC TC 120
      CALL DELSQ(ITER,B)
      KCUNT = C
  120 IF(ITER.EQ.1) GC TC 200
      DO 1C6 II=1,L
      T1= B(ITER-1,II)
      IF(T1.EQ.C.0) T1=0.0001
      T2= B(ITER,II)
      IF(ABS(T2).LT.EPS) GC TO 1C6
      S3= ABS((T2-T1)/T1)
      IF(S3.GT.EPS) GO TC 20C
  1C6 CONTINUE
      GO TC 1C7
  201 WRITE(6,1C)
   10 FORMAT(1HC,47X,'MAXIMUM NUMBER CF ITERATIONS REACHED')
      GO TO 1CCC
  1C7 IF(KCUNT.LT.2) GC TO 110
      CALL DELSQ(ITER,B)
```

```
  110 KFINAL=1
      GC TC 200
  7C0 XNXC=1.0/S2
      SIGMAC=XNXC/TH
      BKCOEF=XNXC*BL**2/(PI**2*KC)
      IPASS=C
      CALL TANMCD(KEC,IPASS)
      WRITE(6,16) ENEW,ITER
   16 FORMAT(1HC,5X,'THE TANGENT MODULLS IS ',1PE10.4,', NC. CF ITERATIC
     1NS IS',CPI4)
      KITER=1
      IF(KEC.EQ.0) GO TO 6C0
      IPASS = 1
      CONST5= 1.C -(XNXC*CCNST2/(C.85*FPRIME*TH))
      CALL TANMCD(KEC,IPASS)
      IF(KEC.EC.C) GC TC 6C0
      IF(KS.EQ.C.C) XNXC=BKCCEF*PI**2*E*CCNST4/BL**2
      CALL PCRIT(XNXC,TH)
      WRITE(6,25), PCR
   25 FORMAT(1H0,5X,'THE TCCAL CRITICAL LCAD IN THE X DIRECTICN IS ',
     1  F9.2)
      WRITE(6,11) XNXC,BKCCEF,ASPECT
   11 FORMAT(1H0,44X,'THE BUCKLING LCAD IS NXC=',E15.8,/ ,5X,'THE PARAME
     1TER K IS',F8.4,' FCR THIS PLATE WITH AN ASPECT RATIC CF',F8.4)
      WRITE(6,9) ITER,S2
    9 FORMAT(1HC,28X,'NEW DEFLECTICN VALUES FCR ITER=',I5,2X,'BUCKLING C
     *OEFFICIENT=',E15.8)
      WRITE(6,4) (W (I),I=1,L)
    4 FCRMAT(2X,12F9.4)
      GO TO 100C
 1001 WRITE(6,7)
    7 FORMAT(1H1)
      STCP
      END
C
      SUBROUTINE MATAB(IPASS)
      DIMENSICN A(121,121),B(121,122)
      COMMCN A,B,KC,KS,DELS,XS,BETA,ZETA,M,N,IB,ETCP,ERT,EBCT,ELT
      REAL KS,KC
      C1=(KS+KC)/DELS
      C2=2.*BETA*KC/DELS
      C3=C1*(12.+ETOP+ELT) + 4.*C2
      C4=-4.*C1-2.*C2
      C5=C1*(12.+ETOP) + 4.*C2
      C6=12.*C1+4.*C2
      C9=C1*(12.+ELT) + 4.*C2
      C10=C1*(12.+ERT) + 4.*C2
      C11=C1*(12.+EBCT+ELT) + 4.*C2
      C12=C1*(12.+EBOT) + 4.*C2
      C13=C1*(12.+EBCT+ERT) + 4.*C2
      C14=C1*(12.+ETCP+ERT) + 4.*C2
      IF(IB)2CC,2CC,2C1
  2C0 C7=0.
      C8=0.
      GO TC 202
  2C1 C7=1.
      C8=-2.-2.*ZETA
  2C2 IF(IPASS)204,203,204
  203 A(1,1)=C3
      A(1,2)=C4
```

```
      A(1,3)=C1
      A(1,M+1)=C4
      A(1,M+2)=C2
      A(1,2*M+1)=C1
      GO TO 205
  204 B(1,1)=C8
      B(1,2)=C7
      B(1,M+1)=ZETA
      B(1,M+2)=-XS
  205 IF(IPASS)207,206,207
  206 A(2,1)=C4
      A(2,2)=C5
      A(2,3)=C4
      A(2,4)=C1
      A(2,M+1)=C2
      A(2,M+2)=C4
      A(2,M+3)=C2
      A(2,2*M+2)=C1
      GO TO 208
  207 B(2,1)=C7
      B(2,2)=C8
      B(2,3)=C7
      B(2,M+1)=XS
      B(2,M+2)=ZETA
      B(2,M+3)=-XS
  208 MM=M-2
      DO 100 KK=3,MM
      IF(IPASS)209,210,209
  210 A(KK,KK-2)=C1
      A(KK,KK-1)=C4
      A(KK,KK)=C5
      A(KK,KK+1)=C4
      A(KK,KK+2)=C1
      A(KK,M+KK-1)=C2
      A(KK,M+KK)=C4
      A(KK,M+KK+1)=C2
      A(KK,2*M+KK)=C1
      GO TO 100
  209 B(KK,KK-1)=C7
      B(KK,KK)=C8
      B(KK,KK+1)=C7
      B(KK,M+KK-1)=XS
      B(KK,M+KK)=ZETA
      B(KK,M+KK+1)=-XS
  100 CONTINUE
      KK=M-1
      IF(IPASS)212,211,212
  211 A(KK,KK-2)=C1
      A(KK,KK-1)=C4
      A(KK,KK)=C5
      A(KK,KK+1)=C4
      A(KK,M+KK)=C4
      A(KK,M+KK-1)=C2
      A(KK,M+KK+1)=C2
      A(KK,2*M+KK)=C1
      GO TO 213
  212 B(KK,KK-1)=C7
      B(KK,KK)=C8
      B(KK,KK+1)=C7
      B(KK,M+KK-1)=XS
```

```
      B(KK,M+KK)=ZETA
      B(KK,M+KK+1)=-XS
  213 IF(IPASS)215,214,215
  214 A(M,M-2)=C1
      A(M,M-1)=C4
      A(M,M)=C14
      A(M,2*M)=C4
      A(M,3*M)=C1
      A(M,2*M-1)=C2
      GO TO 216
  215 B(M,M-1)=C7
      B(M,M)=C8
      B(M,2*M-1)=XS
      B(M,2*M)=ZETA
  216 IF(IPASS)218,217,218
  217 A(M+1,M+1)=C9
      A(M+1,M+2)=C4
      A(M+1,M+3)=C1
      A(M+1,1)=C4
      A(M+1,2*M+1)=C4
      A(M+1,3*M+1)=C1
      A(M+1,2)=C2
      A(M+1,2*M+2)=C2
      GO TO 219
  218 B(M+1,2)=XS
      B(M+1,1)=ZETA
      B(M+1,M+1)=C8
      B(M+1,M+2)=C7
      B(M+1,2*M+1)=ZETA
      B(M+1,2*M+2)=-XS
  219 IF(IPASS)221,220,221
  220 A(M+2,M+1)=C4
      A(M+2,M+2)=C6
      A(M+2,M+3)=C4
      A(M+2,M+4)=C1
      A(M+2,2)=C4
      A(M+2,2*M+2)=C4
      A(M+2,3*M+2)=C1
      A(M+2,1)=C2
      A(M+2,3)=C2
      A(M+2,2*M+1)=C2
      A(M+2,2*M+3)=C2
      GO TO 222
  221 B(M+2,1)=-XS
      B(M+2,2)=ZETA
      B(M+2,3)=XS
      B(M+2,M+1)=C7
      B(M+2,M+2)=C8
      B(M+2,M+3)=C7
      B(M+2,2*M+1)=XS
      B(M+2,2*M+2)=ZETA
      B(M+2,2*M+3)=-XS
  222 LC=M+3
      LU=2*M-2
      DO 101 KK=LC,LU
      IF(IPASS)224,223,224
  223 A(KK,KK-2)=C1
      A(KK,KK-1)=C4
      A(KK,KK)=C6
      A(KK,KK+1)=C4
```

```
      A(KK,KK+2)=C1
      A(KK,KK-M)=C4
      A(KK,M+KK)=C4
      A(KK,2*M+KK)=C1
      A(KK,KK-M-1)=C2
      A(KK,KK-M+1)=C2
      A(KK,M+KK-1)=C2
      A(KK,M+KK+1)=C2
      GO TO 101
  224 B(KK,KK-M-1)=-XS
      B(KK,KK-M)=ZETA
      B(KK,KK-M+1)=XS
      B(KK,KK-1)=C7
      B(KK,KK)=C8
      B(KK,KK+1)=C7
      B(KK,M+KK-1)=XS
      B(KK,M+KK)=ZETA
      B(KK,KK+M+1)=-XS
  101 CONTINUE
      KK=2*M-1
      IF(IPASS)226,225,226
  225 A(KK,KK-2)=C1
      A(KK,KK-1)=C4
      A(KK,KK)=C6
      A(KK,KK+1)=C4
      A(KK,KK-M)=C4
      A(KK,M+KK)=C4
      A(KK,2*M+KK)=C1
      A(KK,KK-M-1)=C2
      A(KK,KK-M+1)=C2
      A(KK,M+KK-1)=C2
      A(KK,M+KK+1)=C2
      GO TO 227
  226 B(KK,KK-M-1)=-XS
      B(KK,KK-M)=ZETA
      B(KK,KK-M+1)=XS
      B(KK,KK-1)=C7
      B(KK,KK)=C8
      B(KK,KK+1)=C7
      B(KK,M+KK-1)=XS
      B(KK,M+KK)=ZETA
      B(KK,M+KK+1)=-XS
  227 KK=2*M
      IF(IPASS)229,228,229
  228 A(KK,KK-2)=C1
      A(KK,KK-1)=C4
      A(KK,KK)=C10
      A(KK,KK-M)=C4
      A(KK,M+KK)=C4
      A(KK,2*M+KK)=C1
      A(KK,KK-M-1)=C2
      A(KK,M+KK-1)=C2
      GO TO 230
  229 B(KK,KK-M-1)=-XS
      B(KK,KK-M)=ZETA
      B(KK,KK-1)=C7
      B(KK,KK)=C8
      B(KK,M+KK-1)=XS
      B(KK,M+KK)=ZETA
  230 LO=2*M+1
```

```
      LU=(N-3)*M+1
      DO 102 KK=LO,LU,M
      IF(IPASS)232,231,232
  231 A(KK,KK)=C9
      A(KK,KK+1)=C4
      A(KK,KK+2)=C1
      A(KK,KK-2*M)=C1
      A(KK,KK-M)=C4
      A(KK,M+KK)=C4
      A(KK,2*M+KK)=C1
      A(KK,KK-M+1)=C2
      A(KK,M+KK+1)=C2
      GO TO 102
  232 B(KK,KK-M+1)=XS
      B(KK,KK-M)=ZETA
      B(KK,KK)=C8
      B(KK,KK+1)=C7
      B(KK,M+KK)=ZETA
      B(KK,M+KK+1)=-XS
  102 CONTINUE
      LO=2*M+2
      LU=(N-3)*M+2
      DO 103 KK=LO,LU,M
      IF(IPASS)234,233,234
  233 A(KK,KK-1)=C4
      A(KK,KK)=C6
      A(KK,KK+1)=C4
      A(KK,KK+2)=C1
      A(KK,KK-2*M)=C1
      A(KK,KK-M)=C4
      A(KK,M+KK)=C4
      A(KK,2*M+KK)=C1
      A(KK,KK-M-1)=C2
      A(KK,KK-M+1)=C2
      A(KK,M+KK-1)=C2
      A(KK,M+KK+1)=C2
      GO TO 103
  234 B(KK,KK-M-1)=-XS
      B(KK,KK-M)=ZETA
      B(KK,KK-M+1)=XS
      B(KK,KK-1)=C7
      B(KK,KK)=C8
      B(KK,KK+1)=C7
      B(KK,M+KK-1)=XS
      B(KK,M+KK)=ZETA
      B(KK,M+KK+1)=-XS
  103 CONTINUE
      LO=3*M-1
      LU=(N-2)*M-1
      DO 104 KK=LO,LU,M
      IF(IPASS)236,235,236
  235 A(KK,KK-2)=C1
      A(KK,KK-1)=C4
      A(KK,KK)=C6
      A(KK,KK+1)=C4
      A(KK,KK-2*M)=C1
      A(KK,KK-M)=C4
      A(KK,M+KK)=C4
      A(KK,2*M+KK)=C1
      A(KK,KK-M-1)=C2
```

```
      A(KK,KK-M+1)=C2
      A(KK,M+KK-1)=C2
      A(KK,M+KK+1)=C2
      GO TO 104
  236 B(KK,KK-M-1)=-XS
      B(KK,KK-M)=ZETA
      B(KK,KK-M+1)=XS
      B(KK,KK-1)=C7
      B(KK,KK)=C8
      B(KK,KK+1)=C7
      B(KK,M+KK-1)=XS
      B(KK,M+KK)=ZETA
      B(KK,M+KK+1)=-XS
  104 CONTINUE
      LO=3*M
      LU=(N-2)*M
      DO 105 KK=LO,LU,M
      IF(IPASS)238,237,238
  237 A(KK,KK-2)=C1
      A(KK,KK-1)=C4
      A(KK,KK)=C10
      A(KK,KK-2*M)=C1
      A(KK,KK-M)=C4
      A(KK,M+KK)=C4
      A(KK,2*M+KK)=C1
      A(KK,KK-M-1)=C2
      A(KK,M+KK-1)=C2
      GO TO 105
  238 B(KK,KK-M-1)=-XS
      B(KK,KK-M)=ZETA
      B(KK,KK-1)=C7
      B(KK,KK)=C8
      B(KK,M+KK-1)= XS
      B(KK,M+KK)=ZETA
  105 CONTINUE
      LO=(N-2)*M+3
      LU=(N-1)*M-2
      DO 106 KK=LO,LU
      IF(IPASS)240,239,240
  239 A(KK,KK-2)=C1
      A(KK,KK-1)=C4
      A(KK,KK)=C6
      A(KK,KK+1)=C4
      A(KK,KK+2)=C1
      A(KK,KK-2*M)=C1
      A(KK,KK-M)=C4
      A(KK,M+KK)=C4
      A(KK,KK-M-1)=C2
      A(KK,KK-M+1)=C2
      A(KK,M+KK-1)=C2
      A(KK,M+KK+1)=C2
      GO TO 106
  240 B(KK,KK-M-1)=-XS
      B(KK,KK-M)=ZETA
      B(KK,KK-M+1)=XS
      B(KK,KK-1)=C7
      B(KK,KK)=C8
      B(KK,KK+1)=C7
      B(KK,M+KK-1)=XS
      B(KK,M+KK)=ZETA
```

```
      B(KK,M+KK+1)=-XS
  106 CONTINUE
      LO=(N-1)*M+3
      LU=N*M-2
      DO 107 KK=LO,LU
      IF(IPASS)242,241,242
  241 A(KK,KK-2)=C1
      A(KK,KK-1)=C4
      A(KK,KK)=C12
      A(KK,KK+1)=C4
      A(KK,KK+2)=C1
      A(KK,KK-2*M)=C1
      A(KK,KK-M)=C4
      A(KK,KK-M-1)=C2
      A(KK,KK-M+1)=C2
      GO TO 107
  242 B(KK,KK-M-1)=-XS
      B(KK,KK-M)=ZETA
      B(KK,KK-1)=C7
      B(KK,KK)=C8
      B(KK,  KK+1)=C7
      B(KK,KK-M+1)=XS
  107 CONTINUE
      KK=(N-2)*M+1
      IF(IPASS)244,243,244
  243 A(KK,KK)=C9
      A(KK,KK+1)=C4
      A(KK,KK+2)=C1
      A(KK,KK-2*M)=C1
      A(KK,KK-M)=C4
      A(KK,M+KK)=C4
      A(KK,KK-M+1)=C2
      A(KK,M+KK+1)=C2
      GO TO 245
  244 B(KK,KK-M+1)=XS
      B(KK,KK-M)=ZETA
      B(KK,KK)=C8
      B(KK,KK+1)=C7
      B(KK,M+KK)=ZETA
      B(KK,M+KK+1)=-XS
  245 KK=(N-1)*M+1
      IF(IPASS)247,246,247
  246 A(KK,KK)=C11
      A(KK,KK+1)=C4
      A(KK,KK+2)=C1
      A(KK,KK-2*M)=C1
      A(KK,KK-M)=C4
      A(KK,KK-M+1)=C2
      GO TO 248
  247 B(KK,KK-M+1)=XS
      B(KK,KK-M)=ZETA
      B(KK,KK)=C8
      B(KK,KK+1)=C7
  248 KK=(N-2)*M+2
      IF(IPASS)250,249,250
  249 A(KK,KK-1)=C4
      A(KK,KK)=C6
      A(KK,KK+1)=C4
      A(KK,KK+2)=C1
      A(KK,KK-2*M)=C1
```

```
      A(KK,KK-M)=C4
      A(KK,M+KK)=C4
      A(KK,KK-M-1)=C2
      A(KK,KK-M+1)=C2
      A(KK,M+KK-1)=C2
      A(KK,M+KK+1)=C2
      GO TO 251
  250 B(KK,KK-M-1)=-XS
      B(KK,KK-M)=ZETA
      B(KK,KK-M+1)=XS
      B(KK,KK-1)=C7
      B(KK,KK)=C8
      B(KK,KK+1)=C7
      B(KK,M+KK-1)=XS
      B(KK,M+KK)=ZETA
      B(KK,M+KK+1)=-XS
  251 KK=(N-1)*M+2
      IF(IPASS)253,252,253
  252 A(KK,KK-1)=C4
      A(KK,KK)=C12
      A(KK,KK+1)=C4
      A(KK,KK+2)=C1
      A(KK,KK-2*M)=C1
      A(KK,KK-M)=C4
      A(KK,KK-M-1)=C2
      A(KK,KK-M+1)=C2
      GO TO 254
  253 B(KK,KK-M-1)=-XS
      B(KK,KK-M)=ZETA
      B(KK,KK-M+1)=XS
      B(KK,KK-1)=C7
      B(KK,KK)=C8
      B(KK,KK+1)=C7
  254 KK=(N-1)*M-1
      IF(IPASS)256,255,256
  255 A(KK,KK-2)=C1
      A(KK,KK-1)=C4
      A(KK,KK)=C6
      A(KK,KK+1)=C4
      A(KK,KK-2*M)=C1
      A(KK,KK-M)=C4
      A(KK,M+KK)=C4
      A(KK,KK-M-1)=C2
      A(KK,KK-M+1)=C2
      A(KK,M+KK-1)=C2
      A(KK,M+KK+1)=C2
      GO TO 257
  256 B(KK,KK-M-1)=-XS
      B(KK,KK-M)=ZETA
      B(KK,KK-M+1)=XS
      B(KK,KK-1)=C7
      B(KK,KK)=C8
      B(KK,KK+1)=C7
      B(KK,M+KK-1)=XS
      B(KK,M+KK)=ZETA
      B(KK,M+KK+1)=-XS
  257 KK=N*M-1
      IF(IPASS)259,258,259
  258 A(KK,KK-2)=C1
      A(KK,KK-1)=C4
```

```
      A(KK,KK)=C12
      A(KK,KK+1)=C4
      A(KK,KK-2*M)=C1
      A(KK,KK-M)=C4
      A(KK,KK-M-1)=C2
      A(KK,KK-M+1)=C2
      GO TO 260
  259 B(KK,KK-M-1)=-XS
      B(KK,KK-M)=ZETA
      B(KK,KK-M+1)=XS
      B(KK,KK-1)=C7
      B(KK,KK)=C8
      B(KK,KK+1)=C7
  260 KK=(N-1)*M
      IF(IPASS)262,261,262
  261 A(KK,KK-2)=C1
      A(KK,KK-1)=C4
      A(KK,KK)=C10
      A(KK,KK-2*M)=C1
      A(KK,KK-M)=C4
      A(KK,M+KK)=C4
      A(KK,KK-M-1)=C2
      A(KK,M+KK-1)=C2
      GO TO 263
  262 B(KK,KK-M-1)=-XS
      B(KK,KK-M)=ZETA
      B(KK,KK-1)=C7
      B(KK,KK)=C8
      B(KK,M+KK-1)=XS
      B(KK,M+KK)=ZETA
  263 KK=M*N
      IF(IPASS)265,264,265
  264 A(KK,KK-2)=C1
      A(KK,KK-1)=C4
      A(KK,KK)=C13
      A(KK,KK-2*M)=C1
      A(KK,KK-M)=C4
      A(KK,KK-M-1)=C2
      GO TO 266
  265 B(KK,KK-M-1)=-XS
      B(KK,KK-M)=ZETA
      B(KK,KK-1)=C7
      B(KK,KK)=C8
  266 LC=2*M+3
      LLU=(N-3)*M+3
      DO 108 JJ=LC,LLU,M
      LU=JJ+M-5
      DO 108 KK=JJ,LU
      IF(IPASS)268,267,268
  267 A(KK,KK-2)=C1
      A(KK,KK-1)=C4
      A(KK,KK)=C6
      A(KK,KK+1)=C4
      A(KK,KK+2)=C1
      A(KK,KK-2*M)=C1
      A(KK,KK-M)=C4
      A(KK,M+KK)=C4
      A(KK,2*M+KK)=C1
      A(KK,KK-M-1)=C2
      A(KK,KK-M+1)=C2
```

```
      A(KK,KK+M-1)=C2
      A(KK,M+KK+1)=C2
      GO TO 108
  268 B(KK,KK-M-1)=-XS
      B(KK,KK-M)=ZETA
      B(KK,KK-M+1)=XS
      B(KK,KK-1)=C7
      B(KK,KK)=C8
      B(KK,KK+1)=C7
      B(KK,M+KK-1)=XS
      B(KK,M+KK)=ZETA
      B(KK,KK+M+1)=-XS
  108 CONTINUE
      RETURN
      END
C
C     SUBROUTINE FOR MATRIX INVERSION
      SUBROUTINE MATINV(XKM,NWID,CARN,A)
      DIMENSION XKM(121,121),A(121,122),I1(122),I2(122),I3(122)
      DO 101 I=1,NWID
      DO101 J=1,NWID
  101 A(I,J)=XKM(I,J)
      DO102 I=1,NWID
  102 A(I,NWID+1)=I
      NI=NWID
      NJ=NI+1
      DO2 L=1,NJ
      I1(L)=L
    2 I2(L)=L
      DET=1.
      DO3 K=1,NI
      C=0.
      DO4 L=K,NI
      IL=I1(L)
      DO4 M=K,NI
      JM=I2(M)
      CC= ABS(A(IL,JM))
      IF(CC-C)4,5,5
    5 KI=IL
      KJ=JM
      KL=L
      KM=M
      C= ABS(A(KI,KJ))
    4 CONTINUE
      IF(K-NI)6,7,6
    6 I1(KL)=I1(K)
      I1(K)=KI
      I2(KM)=I2(K)
      I2(K)=KJ
    7 AKK=A(KI,KJ)
      A(KI,KJ)=1.
      DET=DET*AKK
      CC= ABS(DET)
      IF(CC-1.E10)141,41,41
  141 IF(CC-1.E-70)109,109,9
  109 CARN=0.
      GO TO 500
   41 DET=DET/1.E10
    9 CARN=1.
      DO10 M=1,NJ
```

```
   10 A(KI,M)=A(KI,M)/AKK
      DO3 L=1,NI
      IF(L-K)8,3,8
    8 IL=I1(L)
      AIK=A(IL,KJ)
      A(IL,KJ)=0.
      DO12 M=1,NJ
      JM=I2(M)
   12 A(IL,JM)=A(IL,JM)-AIK*A(KI,JM)
    3 CONTINUE
      DO13 L=1,NI
   13 I3(L)=I1(L)
      DO14 M=1,NI
      DO15 L=M,NI
      IF(I3(L)-I2(M))15,16,15
   15 CONTINUE
   16 IF(L-M)17,14,17
   17 JM=I3(M)
      IL=I3(L)
      I3(L)=JM
      I3(M)=IL
      DET=-DET
      DO18 J=1,NJ
      T=A(JM,J)
      A(JM,J)=A(IL,J)
   18 A(IL,J)=T
   14 CONTINUE
      DO19 M=1,NI
      DO20 L=M,NI
      IF(I2(L)-I1(M))20,21,20
   20 CONTINUE
   21 IF(L-M)22,19,22
   22 JM=I2(M)
      IL=I2(L)
      I2(L)=JM
      I2(M)=IL
      DO23 J=1,NI
      T=A(J,JM)
      A(J,JM)=A(J,IL)
   23 A(J,IL)=T
   19 CONTINUE
      DO103 I=1,NWID
      DO103 J=1,NWID
  103 XKM(I,J)=A(I,J)
  500 RETURN
      END
C
      SUBROUTINE DELSQ(ITER,B)
C        DELSQ APPLIES THE AITKEN'S DELTA-SQUARED EXTRAPOLATION.
C     THIS IS BASED ON THE ASSUMPTION THAT THE W'S HAVE A SINGLE ERROR
C     COMPONENT AND A TRUE VALUE.
      DIMENSION B(121,122)
      COMMON/BLOCK1/W(121),L
      J=ITER-2
      K=ITER-1
      KK=ITER+1
      DO 10 I=1,L
      IF(B(J,I).EQ.W(I).AND.B(K,I).EQ.W(I)) GO TO 10
      W(I)=(B(J,I)*W(I)-B(K,I)**2)/(B(J,I)+W(I)-2.*B(K,I))
   10 B(KK,I)=W(I)
```

```
   30 ITER=ITER+1
      RETURN
      END
C
      SUBROUTINE TANMOD(KEC,IPASS)
C        TANMOD IS BASED ON THE OCTAHEDRAL SHEAR THEORY, TANGENT MODULUS
C     THEORY, AND THE MADRID PARABOLIC STRESS-STRAIN LAW FOR CONCRETE.
      COMMON/BLOCK2/ENEW,BKCOEF,CONST1,CONST3,EPS,E,KSTL,CONST5
      IF(IPASS.EQ.1) GO TO 10
      C=BKCOEF*CONST3
      ENEW=(SQRT(C**2+4.*CONST1)-C)/(2.*CONST1)
      GO TO 20
   10 ENEW = SQRT(CONST5)
   20 TEMP=ABS(ENEW-E)/E
      E = ENEW
      KEC=0
      IF(KSTL.EQ.0) KEC=1
      IF(TEMP.LE.EPS) KEC=1
      RETURN
      END
C
      SUBROUTINE PCRIT(XNXC,TH)
C        THIS SUBROUTINE COMPUTES THE TOTAL AXIAL LOAD ON THE PLATE.  IT
C     INCLUDES THE EFFECT OF THE STEEL ON THE AXIAL STIFFNESS OF THE
C     PLATE IN BOTH THE X AND Y DIRECTIONS.  IT NEGLECTS ANY SHEARING
C     CONTRIBUTIONS BY THE STEEL REINFORCEMENT.
      COMMON/BLOCK4/CONST2,EPO,FPRIME,EST,BL,PCR,RHOT
      TEMP = CONST2*XNXC/(0.85*FPRIME*TH)
      ESUBE = 1.0-SQRT(1.0-TEMP)
      EPSE = ESUBE*EPO
      TEMP = EST*EPSE*RHOT*TH
      PCR =BL*(XNXC*(1.0-RHOT) + TEMP)
      RETURN
      END
```

ACKNOWLEDGMENTS

The author expresses appreciation for the guidance and advice of Dr. S. E. Swartz, Assistant Professor, Department of Civil Engineering, and for his assistance in carrying out the study.

Hearty thanks is also due to the other members of the committee for their efforts in editing and reviewing the thesis: Dr. T. O. Hodges, and Dr. Peter B. Cooper.

BIBLIOGRAPHY

1. Bleich, F., Buckling Strength of Metal Structures, McGraw-Hill Book Co., Inc., New York, 1952

2. Crandal, S. H., Engineering Analysis, McGraw-Hill Book Co., New York, 1956

3. Timoshenko, S. P. and Gere, J. M., Theory of Elastic Stability, 2nd ed., McGraw-Hill Book Co., Inc., New York, 1961

4. Swartz, S. E. and Rosebraugh, V. H., "Buckling of Reinforced Concrete Plates," Journal of the Structural Division, ASCE, Vol. 100, No. ST1, Proc. Paper 10280, January, 1974

A FINITE DIFFERENCE APPROACH

TO BUCKLING OF CONCRETE PLATES

by

FRANCIS ALAN WILEY

B. S. Civil Engineering, 1974

Kansas State University

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

1976

ABSTRACT

This thesis presents the results of a computer program for predicting the elastic buckling loads of plates with various loading and support conditions. The program was modified to predict buckling in reinforced concrete plates.

The method of solution was to approximate the governing differential equation by finite difference coefficients, generate a matrix equation, and solve the matrix equation by iterating on the column vector containing the lateral displacements. Convergence problems were encountered; to overcome this, the Aitken's δ^2-extrapolation was applied.

For elastic buckling several sample problems were run. The maximum error between a program value and a value in the literature for the elastic buckling coefficient was less than 4%. The maximum execution time was 95 seconds.

The modification for reinforced concrete utilized the octahedral shear theory, the tangent modulus theory, and the Madrid parabolic stress-strain law. The effect of steel reinforcing on both the buckling behavior and the axial stiffness was considered. Good agreement between the program results and values given in the literature was obtained. The mesh size had little effect on accuracy. Coarser grids gave acceptable solutions with a minimum of execution time.

With the maximum allowed tension in the plate set equal to $7.5\sqrt{f'_c}$, results indicated that typical shear walls would fail in diagonal tension, not buckling. The limiting $\frac{b}{h}$ ratio was computed to be 240; plates with typical dimensions will have ratios of 60, or less.

The buckled shape of the plate is given, and could be used to create an analytical expression for solving the buckling problem directly.