DESIGN OF REINFORCED CONCRETE COLUMNS UNDER BIAXIAL BENDING

A MASTER'S THESIS

in

Civil Engineering

Middle East Technical University

By

Farzad MARJANI

December 1989

Approval of the graduate school of Natural and Applied Sciences

Prof. Dr. Alpay ANKARA

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Doğan ALTINBİLEK

Chairman of the Department

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Uğur ERSOY

Supervisor

Examining Committee in Charge:

Prof. Dr. Uğur ERSOY ....

Prof. Dr. Ergin ATIMTAY ....

Assoc. Prof. Dr. Mehmet UTKU ....

Asst. Prof. Dr. Güney ÖZCEBE ....

Prof. Dr. Tuğrul TANKUT

Committee Chairman

A B S T R A C T

DESIGN OF REINFORCED CONCRETE COLUMNS UNDER BIAXIAL BENDING

MARJANI, Farzad
M.S. in Civil Eng.
Supervisor: Prof.Dr. Uğur ERSOY
December 1989, 172 Pages

Design of reinforced concrete columns subjected to axial load and biaxial bending is the subject of this study. Two different methods are used to represent the concrete stress-strain relationship. One is by using rectangular stress-block and the other is applicable to any type of stress-strain relationship. A computer program using QUICK BASIC language is developed and numerical solutions for some practical cases are presented. To verify accuracy of the program, results obtained using the methods presented by others are compared with the computer program results.

Key words: reinforced concrete, columns, biaxial bending.

Ö Z E T

BİLEŞİK EĞİLME ALTINDAKİ BETONARME KOLONLARIN DİZAYNI

MARJANI, Farzad
Yüksek Lisans Tezi, İnş. Müh. Bölümü
Tez Yöneticisi: Prof.Dr.Uğur ERSOY
Aralık 1989, 172 Sayfa

Bu çalışmamızın konusu, eksenel yük ve bileşik eğilme altındaki betonarme kolonların dizaynıdır. Beton basınç bloğu için iki ayrı model kullanılmıştır. Bunlardan biri eşdeğer dikdörtgen gerilme dağılımıdır.Diğerinde ise herhangi bir gerilme-birim deformasyon ilişkisi kullanılabilmektedir. QUICK BASIC programlama dilinde çalışmakta olan bir bilgisayar programı geliştirilmiş ve birçok uygulama için sayısal çözümler gösterilmiştir. Programın doğrulğunu saptamak için diğer metodlarla elde edilen sonuçlar bilgisayar programının çıktıları ile karşılaştırılmıştır.

Anahtar kelimeler: betonarme, kolonlor, bileşik eğilme.
Bilim kolu : 624.03.02

To my parents

Anna
Hekim

ACKNOWLEDGEMENTS

The author would like to express his sincere thanks and gratitude to thesis supervisor, Prof. Dr. Uğur Ersoy for his guidance , suggestions and careful supervision offered during the period of this study.

He is also grateful to Mr. Joseph KUBIN for his contributions and assistance in professional programming.

He would also like to thank all of his friends for their support and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

A : area

A_c : cross-sectional area of concrete

A_{cc} : area of compression zone under rectangular stress-block

A_{st} : total area of longitudinal steel bars.

a : depth of equivalent rectangular stress-block

b : width of the rectangular cross-section

c : depth of neutral axis
coefficient in eq. (29)

d_{int} : depth of integration strips

E : modulus of elasticity

E_c : modulus of elasticity for concrete

E_s : modulus of elasticity for steel

e_x : eccentricity of axial load from the x-axis

e_y : eccentricity of axial load from the y-axis

F : force

F_c : resultant concrete force

F_{ci} : resultant force of i'th integration piece

F_{si} : resultant force of i'th longitudinal steel bar

f_{cd} : design strength of concrete

f_{ck} : specified compresive cylinder strength of concrete

f_{yd} : design yield strength of steel

f_{yk} : characteristic yield strength of longitudinal reinforcing steel

f_c' : strength of concrete

f_c'' : 0.85 f'

h : overall depth of cross-section

k_1 : depth of rectangular stress block / depth of neutral axis

K_1 : correction factors obtained by eq. (22.a,22.b,23.a,23.b. ,24.a,24.b) or by graphics represents in Fig. 2.4.a.

K_2 : correction factors obtained by eq. (27) or by graphics represents in Fig. 2.4.b.

k : variable obtained by Eq. (8,10,12,14,16,18)

M : moment

M_e : equivalent moment obtained by Eq. (28)

M_x : moment around x-axis

M_y : moment around y-axis

M_{ux} : ultimate moment capacity about the x-axis for a section subjected to N_u and M_{uy}

M_{uy} : ultimate moment capacity about the y-axis for a section subjected to N_u and M_{ux}

M_{xd} : design moment with respect to x-axis

M_{yd} : design moment with respect to y-axis

M_{uxo} : Ultimate moment capacity about the x-axis for a section subjected to uniaxial bending

M_{uyo} : Ultimate moment capacity about the y-axis for a section subjected to uniaxial bending

m : exponent in Eq. (2)
variable in Eq. (20) and (30)

m_x : variable obtained by eq. (5)

m_y : variable obtained by eq. (6)

N : axial load

N_x : ultimate load capacity when only eccentricity ey is presented

N_y : ultimate load capacity when only eccentricity ey is presented

N_1 : Maximum axial load permitted by TS-500 ($0.6\ f_{ck}\ A_c$).

N_{or} : axial load capacity of a section subjected to axial load only

N_u : ultimate load under biaxial bending

n : exponent in Eq. (2)
variable obtained by Eq. (4)

α : variable in Eq. (26)

α_{ne} : angle of neutral axis

β : a variable in Eq. (3.b)

ϵ_{ci} : concrete strain level at i'th integration piece

ϵ_{cu} : ultimate strain at the extreme concrete compresion fibber

ϵ_{si} : strain at i'th steel bar

μ : variable obtained by eq. (7)

σ_{ci} : stress level at the middle of i'th strip

σ_{si} : stress level at the i'th steel bar

C H A P T E R 1

INTRODUCTION

1.1. General.

Columns are structural members used primarily to support compressive loads. However due to the monolithic nature of reinforced concrete construction, there is always biaxial bending moments caused by gravity loads or by lateral loads (wind or seismic excitation). In the case of reinforced concrete columns and shearwalls, wide-flange cross sections have been used to improve the structural strength of the members and L-shaped sections are usually located at the corners of buildings, C-shaped cross sections commonly are used as columns and enclosures of the elevator shafts, S- and X-shaped sections have been used for purely architectural reasons, and other irregular sections are used in the precast concrete industry. In concrete bridge pier construction, hollow box or round columns frequently are used. Hollow round cross sections also are used in piling and pole construction.

Many charts, graphs and tables have been developed to simplify the design of reinforced concrete columns under axial load and biaxial bending. A great majority of these design aids are for rectangular cross-sections. The designer who has to design a non-rectangular cross-section under biaxial bending and axial load has to make a detailed analysis, spending a lot of time. Therefore a computer program which enables the design of any arbitrary cross-section under axial load and biaxial bending would be extremely helpfull.

1.2. Object and Scope of the Study.

The design of reinforced concrete columns has been studied by many investigators. The problem presented by a column subjected to axial load and uniaxial bending has been satisfactorily solved. Columns subjected to biaxial bending in addition to axial load present a more complicated problem. Although several researchers have developed methods of solution, none of these can be considered as being general and applicable to any type of a cross-section. Since the solutions presented are very lengthy and many of these are not design oriented, some approximate solutions have been presented for design purposes. Most of these approximate methods lead to overdesign of the members.

The main objective of this study was to develop a computer program for the design of reinforced Concrete sections of any arbitrary shape subjected to biaxial bending and axial force. This program should be able to solve cross-section of any geometry with any steel configuration. Also it should be possible to use any kind of stress-strain relationship for the concrete.

In this study such a computer program was developed. The program is design oriented. It can be entered by defining the cross-sectional geometry, steel configuration and material properties (including stress-strain relationship for concrete). When axial force and moments are also specified, the program gives the amount of reinforcement needed. The program is simple and uses minimum computer time.

In this study it was also intended to investigate the feasibility of rectangular stress-block in designing reinforced concrete columns with arbitrary cross-sections subjected to biaxial bending. The efficiency of rectangular stress-block in reducing the computer time was also investigated.

CHAPTER 2

REVIEW OF PREVIOUS STUDIES

2.1. General.

Until recently, design and analysis of reinforced concrete columns were based on the working stress method, assuming linear elastic stress-strain relationship for both materials. The designer proceeded by selecting the dimensions of the section and the configuration and amount of the reinforcement. Then he/she would compute the capacity of the section (limiting the stresses by allowable values), making sure that it would be greater than or equal to that required value. Reinforced concrete columns were usually overdesigned by selecting a large cross-section and more reinforcing steel than necessary to overcome the bending stresses. With the developement of the ultimate strength method, the design was based on the actual inelastic properties of the concrete and steel. Based on the ultimate strength method, new charts , graphs and tables were developed to help the engineer in designing reinforced concrete columns subjected to axial loads and uniaxial bending. These design aids were also used in designing reinforced concrete columns subjected to axial loads and biaxial bending by approximating the relationship between uniaxial bending and the biaxial bending. Attempts were made to produce design charts for columns subjected to biaxial bending. However such attempts did not yield practical results, because numerous charts would be needed to meet the design requirements.

The general solution of the problem can be obtained by using; (a) equilibrium ,(b) compatibility and (c) force-deformation relationships. First, the position of the neutral axis is assumed. In biaxial bending the position of the neutral axis is defined by two variables (depth & inclination). Assuming linear strain distribution and knowing the strain capacity of concrete, steel strains can be determined from compatibility. Then using the stress-strain relationships, forces in steel bars can be determined. Concrete resultant force can be found by using a suitable stress block. If the equilibrium is not satisfied, position of the neutral axis is altered. When force equilibrium is satisfied, moments can be computed.

2.2. Methods Based on Approximation of Failure Surface.

In case of members subjected to axial load and biaxial bending, the failure envelope becomes a surface instead of curve obtained in uniaxial bending, Fig.2.1.a. Several simplified methods have been developed by approximating this surface. These methods are based on section control. i.e. each time the capacity of the section should be checked to see whether the section can resist the load effects or not.It is obvious that by these formulations the required steel area can not be obtained directly. Therefore such methods are not very suitable for design purposes.

In 1960 Bresler [8] in his historical paper has suggested two expressions for the strength of a biaxially loaded column. One was an equation inspired from the Russian code and the related publications (Method A). The equation is obtained by approximating the interaction surface, See Fig. 2.1.(a)

$$\frac{1}{N_u} = \frac{1}{N_x} + \frac{1}{N_y} - \frac{1}{N_{or}} \qquad \ldots\ldots\ldots(1)$$

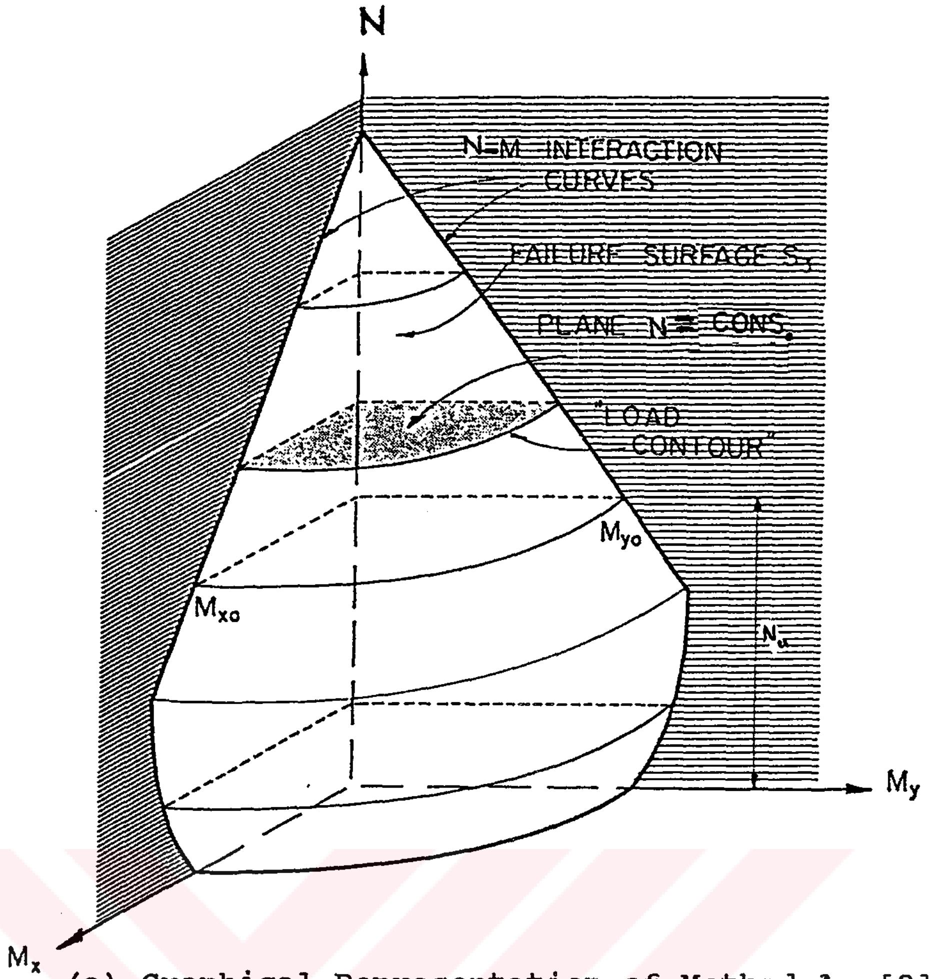
Where:

N_x = Ultimate load capacity when only eccentricity e_y is presented.

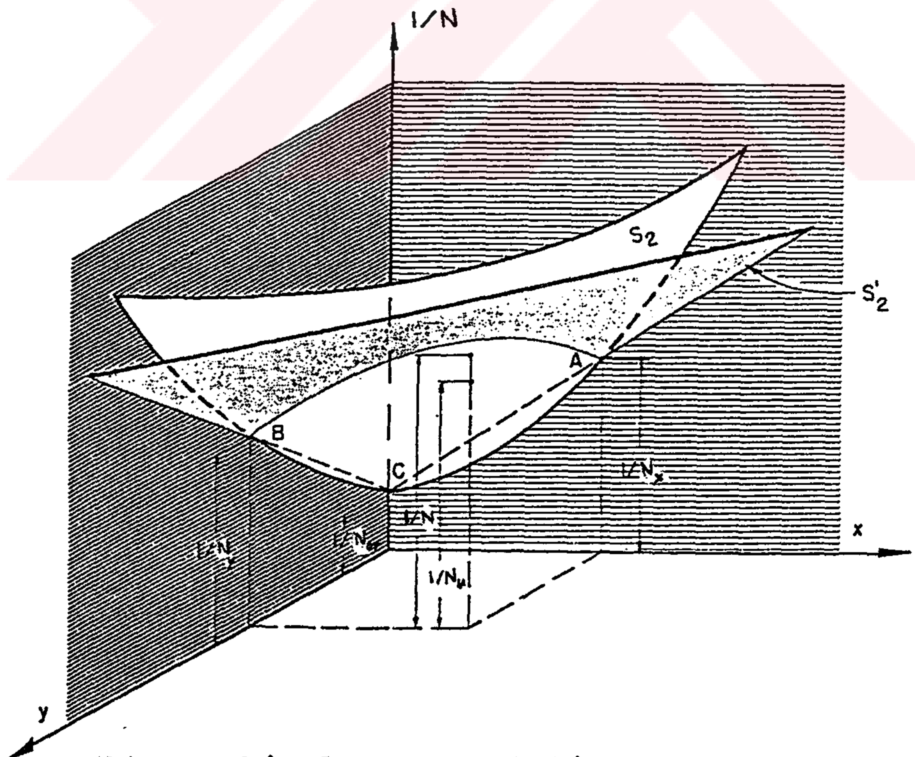
N_y = Ultimate load capacity when only eccentricity e_x is presented.

N_{or}= Ultimate load capacity when there is no eccentricity.

This expression, which is the most commonly used approximate method, is recommended in the Turkish code (TS-500). It is said to yield reasonable results for loads above 0.1 N_{or}. Bresler compared the results obtained by this equation with the results obtained using more exact methods and test results. He claimed that the maximum deviation from exact solution and test results is 9% and 16% respectively. E. Çokça [15] claimed that when a more realistic σ-ϵ curve is used, maximum deviation from test results drops to 9% (capacity comparison).



(a) Graphical Representation of Method A. [8]



(b) Graphical Representation of Method B. [8]

Fig.2.1. Graphical Representation of Bresler's Method.

The other equation proposed by Bresler represents sections of the interaction surface corresponding to various levels of axial load N (Method B), Fig. 2.1.(b)

$$\text{i.e.} \quad \left(\frac{M_{ux}}{M_{uxo}} \right)^{m} + \left(\frac{M_{uy}}{M_{uyo}} \right)^{n} = 1 \qquad \text{......(2)}$$

Where:

M_{ux} = $N_u \cdot e_y$ = Moment about X-axis.

M_{uy} = $N_u \cdot e_x$ = Moment about Y-axis.

M_{uxo} = Uniaxial flexural strength about x-axis for the axial load under consideration.

M_{uyo} = Uniaxial flexural strength about y-axis for the axial load under consideration.

m,n = Exponents depending on column properties.

This method has been revised by various researches by making different proposals for m and n. In the British Code (CP110-72), m=n=1 is suggested for low axial loads. These parameters are linearly increased up to 2.0 for high levels of axial loads.

Meek [32] in 1963 approximated the curved interaction relationship between moments corresponding to a given axial load by two straight lines as shown in Fig. 2.2. Points A and C are uniaxial moment capacities and point B is the capacity at the diagonal. On the same figure values obtained from his experiments are also marked.

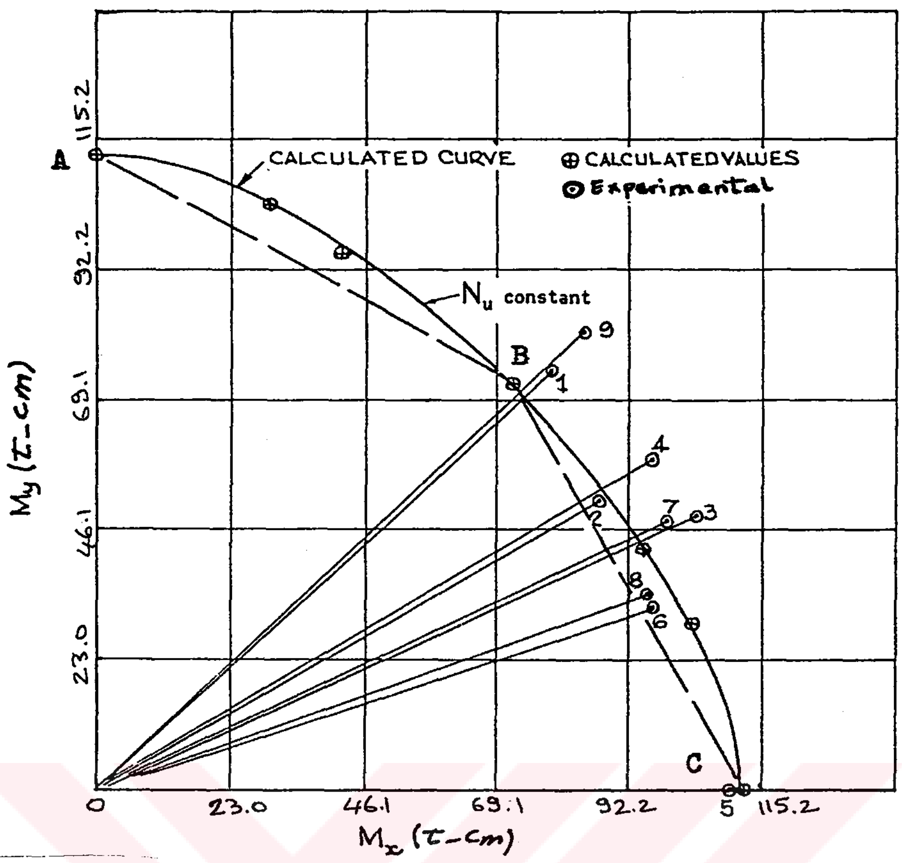


Fig. 2.2. Moment Interaction Under Constant Axial Load.[32]

A simplified ultimate strength design procedure was presented by Fleming and Werner [24] in 1965, and a set of design curves were given for the most commonly used concrete strengths and for different steel percentages. This method was also based on the approximation of the interaction surface.

Weber [43] in 1966 developed a method for approximating the failure surface based on linear interpolation between the capacity for bending on the diagonal and the capacity for bending on a principal axis. He produced design charts for columns with equal bending resistance on both principal axes. These charts are limited in scope and are not intended to cover the full range of parameters encountered in normal design. According to Weber's study the maximum error in these charts was found to be about 5.3% .

Parme, Nieves and Gouwens [38] in 1966 suggested that the response of a column to biaxial bending must be related to the uniaxial resistance of that section. Based on this assumption, they suggested design equations similar to Bresler's. In these equations bending moments about each axis are related by a single though variable exponent.

$$\left(\frac{M_x}{M_{ux}}\right)^n + \left(\frac{M_y}{M_{uy}}\right)^n = 1 \qquad \text{...........(3.a)}$$

or:

$$\left(\frac{M_x}{M_{ux}}\right)^{\log .5/\log \beta} + \left(\frac{M_y}{M_{uy}}\right)^{\log .5/\log \beta} = 1 \qquad \text{...........(3.b)}$$

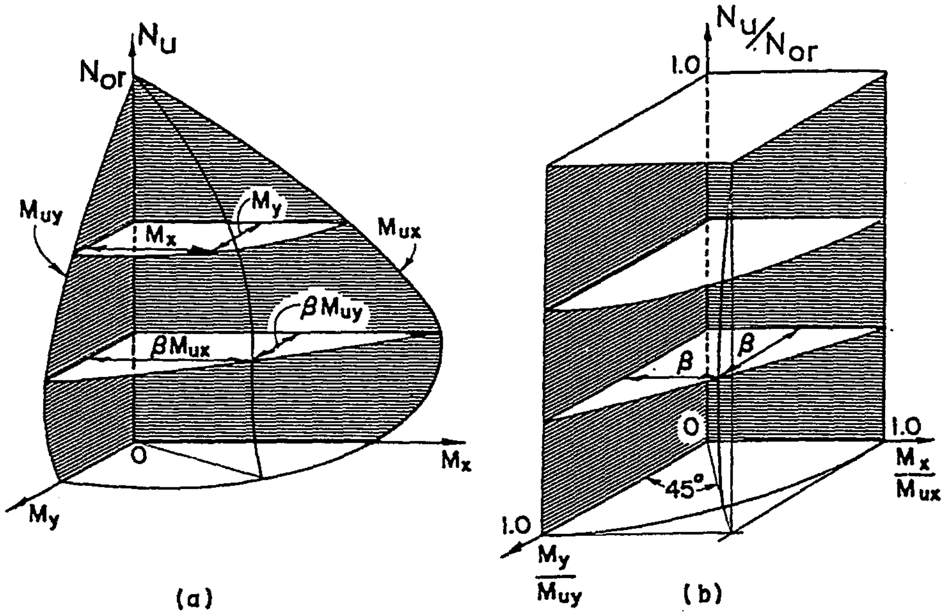
Where:

M_{ux} = The uniaxial flexural strength about the X-axis.

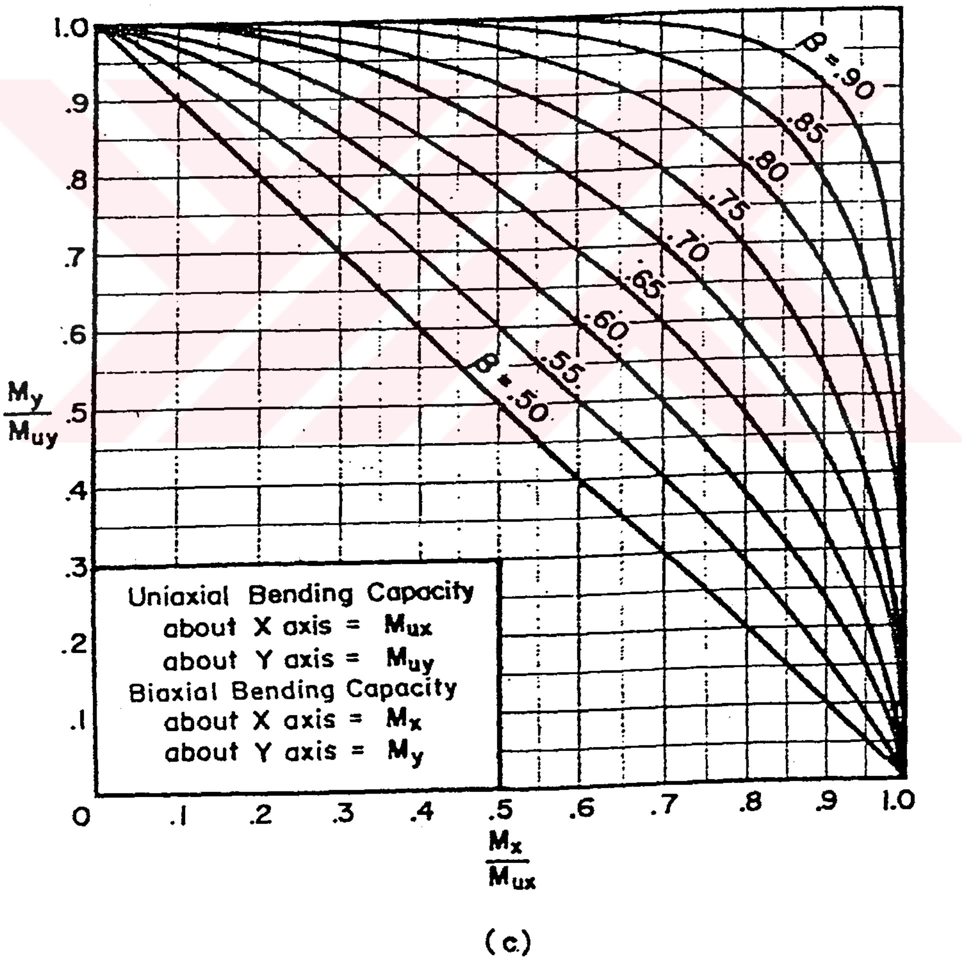
M_{uy} = The uniaxial flexural strength about the Y-axis.

Þ = The ordinate of the contours at the point at which the relative moments are equal.Fig. 2.3.(b)

With known value of β , the bending resistance in a direction of an eccentrically loaded column can be obtained. For design convenience, plots of curves generated for different values of β for a rectangular column section are given in Fig. 2.3.(c) It can be concluded that the biaxial bending resistance of an



Ultimate Capacity Surfaces. [38]



Biaxial Moment Relationship. [38]

Fig.2.3. Interaction Relationship. [38]

axially loaded column can be represented graphically as a surface formed by a series of interaction curves drawn radially from the ultimate load N_u axis. Fig. 2.3.(a) When the bending resistance is plotted in terms of the dimensionless parameters N_u / N_{or} , M_y / M_{uy} , M_x / M_{ux} , with the latter two terms designated as the relative moments, the failure surface generated assumes the shape shown in Fig. 2.3.(b) .

N_u = the ultimate load under biaxial bending.

N_{or} = the ultimate load when there is no eccentricity.

Gouwens [27] in 1977 approximated the interaction between M_{ux} and M_{uy} for a constant N by two straight lines similar to Meek's suggestion. The method was more conservative and simple equations were used instead of charts.

Shanmughasundaram [40] in 1977 proposed a method to obtain the exponent n suggested by Parme et al. without calculating β . But this method has not the same simplicity.

Investigations by Hsu [21,22,23] showed that the formula proposed by Bresler for load contour simulation (2) is also applicable to the arbitrary cross sections such as L, channel and T-sections. He made some experimental and analytical (discrete area approach) studies on these sections. According to his studies, for practical design purposes using exponent of m=n=1.5 in the formula gives satisfactory results. It should be noted that the same exponent is normally used for rectangular sections.

2.3. Approximate Design Formulations.

Solanki [41] proposed a method in his paper published in 1980 to provide a simple estimate for the steel area to be used in a more detailed biaxial analysis. Two formulas were suggested for a section with uniformly spaced steel bars. The method was found to be satisfactory when compared with Bresler's and Parme's methods.

Çakıroğlu and Özer [14] in 1983 proposed simple formulas for the ultimate strength design of rectangular sections. In this method the required steel area for a given section or the section dimensions for a given steel ratio can be obtained directly . The method is very suitable for design purposes. Formulas were derived for the case of rectangular sections with symmetrically placed steel at four corners and concrete cover ratio of 0.1 ,and reinforcing steel grades of BÇ-I and BÇ-III. The formulas suggested can also be used for analysis to calculate the ultimate load carrying capacity of a given section.

The non-dimensional parameters were specified as:

$$n = \frac{N}{b\ h\ f_{ck}} \qquad \text{.............(4)}$$

$$m_x = \frac{M_x}{b\ h^2\ f_{ck}} \quad \text{.............................(5)}$$

$$m_y = \frac{M_y}{b^2\ h\ f_{ck}} \quad \text{.............................(6)}$$

$$\mu = \frac{A_{st}}{b\ h} \cdot \frac{f_{yk}}{f_{ck}} \quad \text{........................(7)}$$

Where:

b = Depth of rectangular cross-section.(dimension in x-direction)

h = Hight of sectangular cross-section.(dimension in y-direction)

The formulas are:

- For BÇ-I (S220) Steel Grade,

if $n < 0.2$

Then: $k = (3-6.65\ n)\ m_y + 2.65\ n + 0.1$ ....(8)

$\mu = 2.86\ m + 2.92\ n^2 - 1.48\ n$ ....(9)

if $0.2 \leq n \leq 0.3$

Then: $k = (-1.33 + 15\ n)\ m_y - 1.4\ n + 0.91$ ....(10)

$\mu = 2.92\ m - 0.192$ ....(11)

if $n > 0.3$

then: $k = (3.67 - 1.67\ n)\ m_y + 0.85 - 1.2\ n$ ..(12)

$\mu = 2.92\ m + 0.62\ n^2 + 0.254\ n - 0.33$ ..(13)

- For BÇ-III (S420) Steel Grade,

if $n < 0.2$

then: $k = (2 - 6.65\ n)\ m_y + 2.65\ n + 0.2$ ....(14)

$\mu = 2.86\ m + 2.92\ n^2 - 1.48\ n$ ....(15)

if $0.2 \leq n \leq 0.3$

then: $k = (-4.33 + 25\ n)\ m_y - 2.05\ n + 1.14$ ..(16)

$\mu = 2.93\ m - 0.192$ ....(17)

if $n > 0.3$

then: $k = (3.67 - 1.67\ n)\ m_y + 0.75 - 0.75\ n$ ..(18)

$\mu = 2.92\ m + 0.62\ n^2 + 0.254\ n - 0.32$ ...(19)

Where:

$m = m_x + k\ m_y \quad (m_y \leq m_x)$ ....(20)

Total steel area is obtained by the following formulation.

$$A_{st} = K_1 \cdot K_2 \cdot \mu \cdot \frac{f_{ck}}{f_{yk}} \cdot b \cdot h \quad ....(21)$$

Where K_1 and K_2 are the correction coefficients. These coefficients can be obtained either by the following formulation or from the graphs presented in Fig. 2.4.

K_1 is related to the distribution of reinforcing bars. The cases are:

a) Uniformly distributed reinforcement along four sides,

for $n \leq 1$

$K_1 = 1.137 + 1.223\ n - 3.445\ n^2 + 3.083\ n^3 - 0.87\ n^4$

.......(22.a)

for $n > 1$ then $K_1 = 1.12$ ....(22.b)

b) Reinforcement lumped at eigth points; four at the corners and four between the corners.

for $n \leq 1$

$$K_1 = 1.07 + 1.349\,n - 3.99\,n^2 + 4.062\,n^3 - 1.406\,n^4 \quad \text{.......(23.a)}$$

for $n > 1$ then $K_1 = 1.08$ ....(23.b)

c) $(3/16)A_{st}$ is located at the corners and $(1/16)A_{st}$ is located between the corners.

for $n \leq 1$

$$K_1 = 1.028 + 0.633\,n - 1.868\,n^2 + 1.816\,n^3 - 0.584\,n^4 \quad \text{......(24.a)}$$

for $n > 1$ then $K_1 = 1.025$ ....(24.b)

d) Reinforcement is equally and uniformly distributed along two opposite sides.

if $m_y / m_x \leq 0.4$ then:

for BÇ-I $K_1 = 1 + (m_y/m_x)^2$ ....(25.a)

for BÇ-III $K_1 = 1 + .5(m_y/m_x)^2$ ....(25.b)

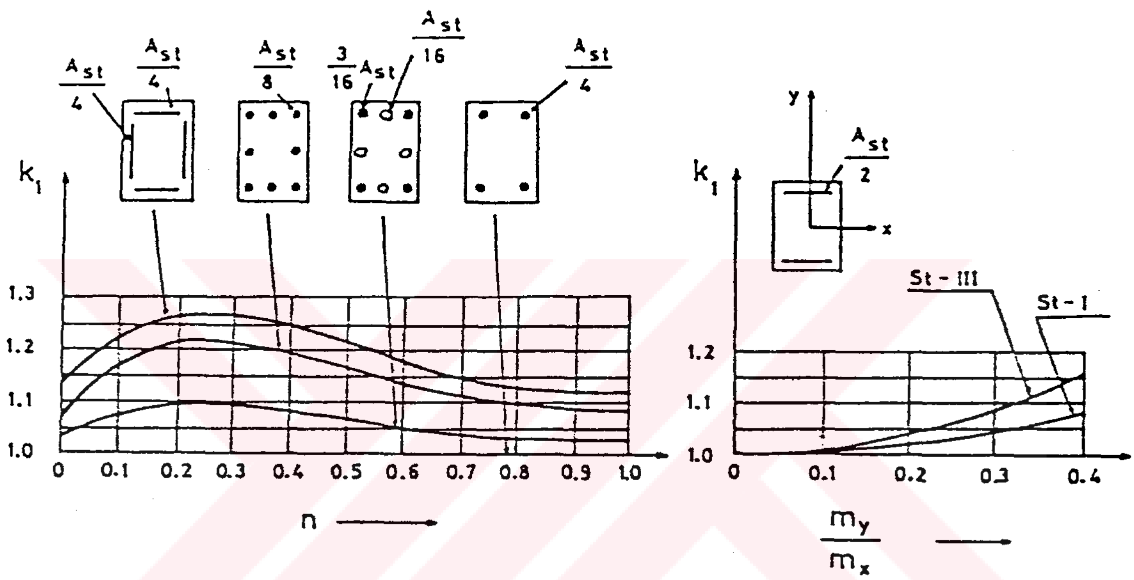
K_2 is related to the concrete cover ratio and can be obtained from following formulas or graphs shown in Fig.2.4.(b).

$$\alpha = \frac{h - 2h'}{h} \quad \text{.............................(26)}$$

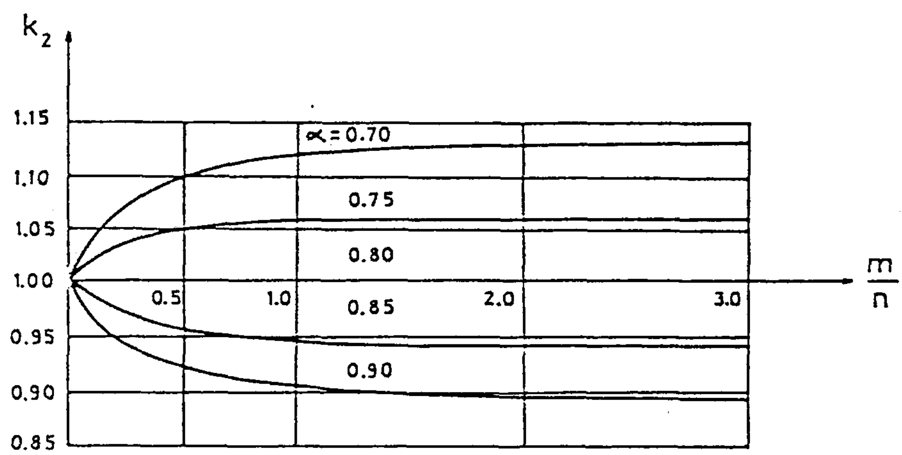
and

$$K_2 = \frac{(m/n)\,(0.8/a) + 0.2}{(m/n) + 0.2} \quad \text{...................(27)}$$

The limitations of this method are; (a) it is for rectangular sections only and (b) CEB stress block has to be used. Also some error is introduced when the failure surface is represented by simple approximate equations.



(a) K_1 Coefficient.[14]



(b) K_2 Coefficient.[14]

Fig. 2.4. K_1 and K_2 Coefficients.

2.4. Discrete Area Approach for Ultimate Capacity.

First study on the moment curvature relations for reinforced concrete compresion members under biaxial bending was made by Warner [42] in 1969. A finite element method was used in which the concrete and steel areas in the cross-section were broken into many small discrete areas. Biaxial moments were found by summing the moments of the elemental forces. This approach proved to be very advantegeous in that stress-strain relations for concrete and steel of almost any desired form could be used and irregularities in the shape of the cross-section and in the placement of steel reinforcement could both be taken into account.

Lachance [31] in 1980 studied the ultimate biaxial bending strength of rectangular and L-shaped sections using discretized area approach. He used Newton-Raphson method of iteration to satisfy the equilibrium conditions. He also used different concrete stress-block and showed that the factor which influenced the ultimate biaxial strength most, was the compressive strength of concrete.

Al-Noury and Chen [3] in 1982 partitioned the cross-section into a number of small elementary areas of steel and concrete, and obtained the Moment-Curvature-Thrust relationships by Step-by-Step application of the force-deformation equilibrium equations. They also extended and verified contour interaction equation for its applicability to composite sections.

2.5. Studies on Ultimate Strength Capacity.

Ultimate strength of square Columns with equal amounts of reinforcement in each of the four faces or only in opposite faces, under biaxially eccentric loads was investigated by Furlong [26] in 1961. He provide and interaction diagrams for different steel and concrete strengths.

Aas-Jacobsen [1] in 1964 proposed to replace the moments with the biaxial eccentricities e_x and e_y in a main plane of symmetry and with an equivalent moment M_e.

$$M_e = (N_u\ e_x + C\ M_1)\ m \qquad \text{.............(28)}$$

Where:

$$C = \frac{e_y\ b}{e_x\ h} \qquad \text{..........................(29)}$$

$$m = (\ 1 + C^2\ )^{.5} \qquad \text{....................(30)}$$

These formulas were for bending about the Y-axis. For bending about the X-axis, e_y and b should be replaced by e_x and h respectively. M_1 is a small additional moment which depends on the mode of failure, the section shape, dimensionless load $\frac{N_u}{f_{ck}\ b\ h}$ and relative reinforcement index $\frac{f_{yk}\ A_{st}}{f_{ck}\ b\ h}$.

In most cases $M_1 = 0$.

Farah and Huggins [20] in 1969 presented an integration method which could be applied to the case of regularly shaped columns such as rectangular or square. According to their study, using a continuous fourth-order polynominal fitted by the least squares method to stress-strain data obtained from uniaxial compresion tests of concrete ,the following relationship exists. see Fig. 2.5.(a)

$\sigma_c = f_{ck} \Sigma K_i \epsilon^i$ = Stress in concrete. ....(31)

Where:

f_{ck}= Compressive strength of concrete.(Psi)

$i = 1 , 2 , 3 , 4$ and K_i is constant.

$K_1 = 0.958 \times 10^3$

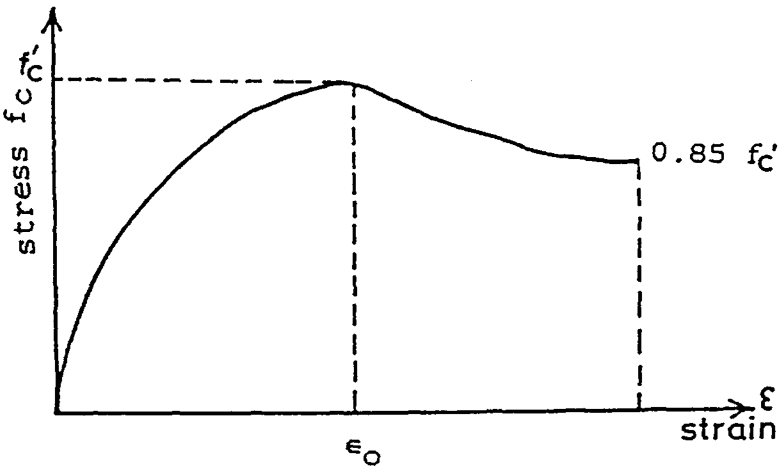
$K_2 =-0.312 \times 10^6$

$K_3 = 0.306 \times 10^8$

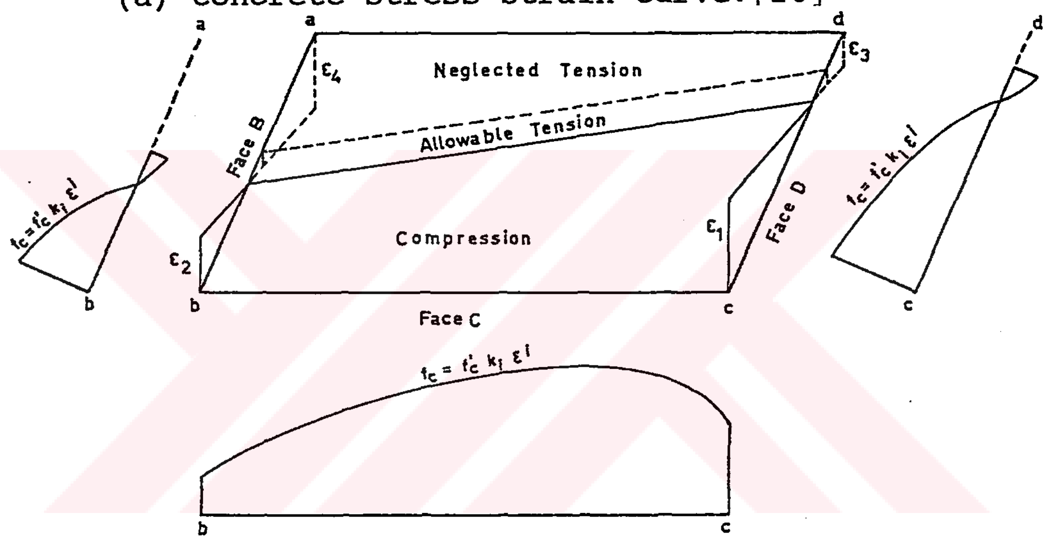
$K_4 =-0.257 \times 10^9$.

The limiting concrete compressive strain is taken as 0.004 and the stress corresponding to this is assumed to be 0.85 f_{ck} The section strain distribution is determined by the three corner strains, namely ϵ_1 , ϵ_2 , ϵ_3 , Fig. 2.5.(b).

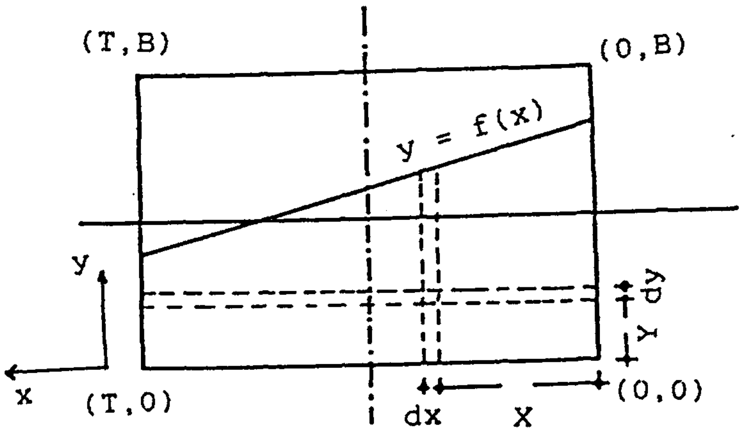
The origin O(0,0) is taken as the corner subjected to the highest compressive strain, with the coordinate axes as shown in Fig.2.5.c.



(a) Concrete Stress-Strain Curve.[20]



(b) Stress and Strain Distribution in Section.[20]



(c) Area and Limits of Integration.[20]

Fig. 2.5. Method Proposed by Farah and Huggins.

$$\epsilon(x,y) = \epsilon_1 - (\epsilon_1-\epsilon_3)\,\frac{x}{T} - (E_1-\epsilon_2)\,\frac{y}{T} \quad \text{.........(32)}$$

and the corresponding stress:

$$\sigma_c(x,y) = f_{ck}\,\Sigma\,K_i\,\epsilon^i(x,y) \quad \text{....................(33)}$$

Where:

T = The width of the section.

B = The depth of the section.

Considering the elementary area dxdy .Fig.2.5.c, the force N on the column is the summation of concrete forces on the elementary areas, plus the steel force.

$$N = \int_0^T \int_0^{f(x)} \sigma_c(x,y)\,dxdy + N_{st} \quad \text{.............(34)}$$

Where:

N_{st} = Force of the steel.

The elementary moment about x-axis is dM_x Where:

$$dM_x = \sigma_c(x,y)\,dxdy\,\left(\frac{B}{2} - y\right) \quad \text{..............(35)}$$

And the total moment M_x is the sum of the elmentary moments.

$$M_x = \int_0^T \int_0^{f(x)} \sigma_c(x,y)\,\left(\frac{B}{2} - y\right) dydx + Mx_{st} \quad \text{...(36)}$$

Where:

Mx_{st} = The moment of steel

Similarly:

$$dM_y = \sigma_c(x,y)\ dxdy\ (\frac{T}{2} - x) \qquad \ldots\ldots\ldots\ldots\ldots\ldots(37)$$

$$M_y = {}_o\int^{T} {}_o\int^{f(x)} \sigma_c(x,y)\ (\frac{T}{2} - x)\ dydx + My_{st} \qquad \ldots\ldots\ldots(38)$$

Solution of these equations was obtained by assuming the values for ϵ_1 , ϵ_2 and ϵ_3 and iterating these values until the equations for N , M_x , M_y were satisfied.

Dinsmore [16] in 1982 developed a computer program based on the criteria established by the ACI Commitee 318. Biaxial bending problems that the designer faces were solved directly without requiring design charts or tables. Program solved the moment capacity M_{ux} and M_{uy} and drew the interaction diagram for the specified input "N_u" value. To determine the capacity of the section from the output, the values could be interpolated between interaction plot points, which would always be conservative between points and more accurate at the point itself. To enter the program, the designer had to have location and amount of reinforcement.

Brondom-Nielsen [9] in 1982 developed a method, by writing moment and force equilibrium with respect to the axes through the normal force N_u and parallel to arbitrary orthogonal X- and Y-axes.Using this approach the following relationships can be obtained. (see Fig.2.6.)

$$M_x = \Sigma (y_i - e) A_i \sigma_{si} - f_{cd} \int (y - e)\, dA_c = 0 \quad \ldots(39)$$

$$M_y = \Sigma (x_i - n) A_i \sigma_{si} - f_{cd} \int (x - n)\, dA_c = 0 \quad \ldots(40)$$

$$N_u = f_{cd} A_c - \Sigma A_i \sigma_{si}$$

Where:

M_x = Moment about X-axis.

M_y = Moment about Y-axis.

A_i = area of the i'th steel bar.

σ_{si}= Normal stress of i'th steel bar.

f_{cd}= Concrete Compressive strength.

A_c = Concrete area.

N_u = The normal force.

x_i, y_i = Coordinates of i'th steel bar.

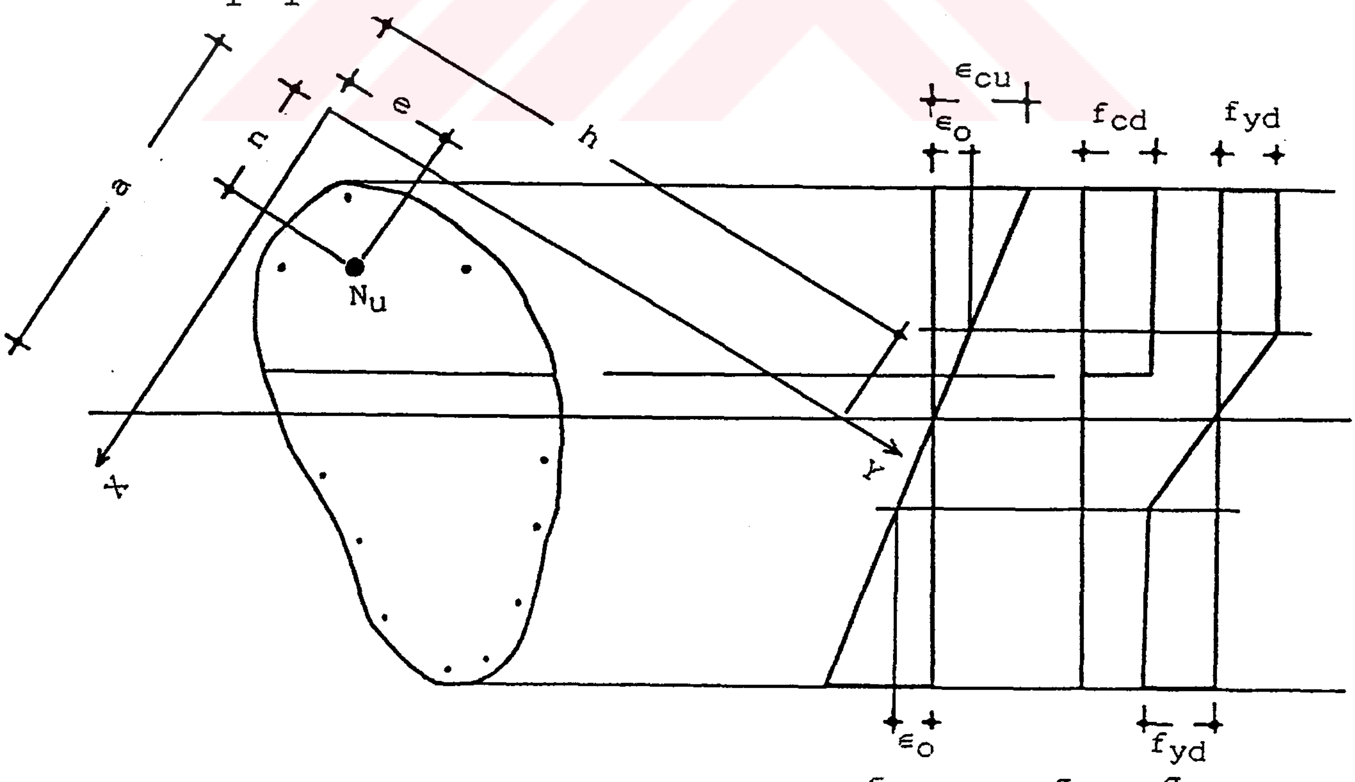


Fig. 2.6. Arbitrary Cross-Section Subjected to Biaxial Bending.[9]

The nonlinear equations generated by this technique were solved by a two-dimensional Newton-Raphson iteration using finite differences in lieu of the partial derivatives. Although this technique is suitable for arbitrary cross-sections, the program can only shift automatically to triangular, trapezoidal and pentagonal compression shapes. This difficulty was overcomed by the author later. In the method presented in one of his papers [10] published in 1985, he used polygonal coordinate system to generate arbitrary cross-sectional shapes. It was even applicable to prestressed sections.

Kawakami et al. [29] in 1985 developed a computer program to study ultimate limit strength of reinforced concrete sections of arbitrary shape. This method was also adaptable for predicting cracking strength of prestressed sections . Equilibrium of axial forces and biaxial bending were derived analytically by using Gauss' integral. The data for cross-sections were given by a polygonal system.

2.6. Experimental Research.

The first experimental research related to columns subjected to biaxial bending was made by Meek [32] in 1963. He tested nine square column under small eccentricities and found the interaction line to be nearly circular in shape, which is only true for small eccentricities. In determining the ultimate bending moments, second order effects were not taken into account.

Ramamarthy [39] in 1966 presented an experimental work on the biaxial bending resistance of square and rectangular sections with eight bars evenly distributed on each face.Lateral deflections of the columns under biaxial loading were recorded and ultimate load of all columns were determined on the basis of final eccentricities at the time of failure. Different levels of axial load and eccentricities were applied to the specimens.

Heimdahl and Binachini [28] in 1975 tested 48 corner-reinforced tied columns with square cross-sections at various magnitudes and angles of eccentricities. They modified nominal eccentricities by considering the lateral deflections measured at ultimate load in each test. They concluded that equivalent rectangular block gave reasonably good results in prediction of ultimate strength of eccentrically loaded columns.

Furlong [26] in 1979 tested nine columns having rectangular cross-sections with the identical dimensions. They were all reinforced with 10 longitudinal bars. The test was under different levels of axial load and different skew angles. It has been found that the inclination of the neutral axis increased as the skew angle of loading departed from the minor axis. The experimental results of Furlong [26] will be discussed and compared with the results obtained from Author's computer program in chapter 6.

Hsu had some experimental research on the ultimate strength of reinforced concrete columns with arbitrary cross-sections such as L [21], Channel [22] and T shapes [23]. The experimental results of Hsu will be discussed and compared with the outputs of Author's computer program results in chapter 6.

Recently, several experimental studies have been reported on columns subjected to cyclic loads causing biaxial moments.

C H A P T E R 3

ULTIMATE STRENGTH OF REINFORCED CONCRETE COLUMNS

3.1. General.

Reinforced concrete columns are classified into three categories, (a) tied columns, (b) spiral columns and (c) composite columns. The behaviour of tied and spiral columns are identical up to the level of ultimate strength. Under uniaxial loading the ultimate strength in both types is reached when concrete reaches the crushing strength. At this stage the steel has also yielded. Beyond this stage (after first peak) the behaviour depends on the strength and ductility of the core. Strength and ductility of the core is a function of the confinement provided by the lateral reinforcement. If adequate spiral is provided, the spiral columns can reach a second peak after some excessive deformation. Since ties are not as effective as spirals, the presence of ties generally increase the ductility and does not affect the strength unless excessive amount of ties are provided.

In reinforced concrete columns classical buckling is not a problem. However second order moments due to deflections can magnify the moments significantly. In design practice second order effects are usually added to the frame moments found or these moments are magnified by multiplying the moments obtained from frame analysis by a magnification factor. Therefore the " design moment " is the magnified moment which includes the second order effects.

3.2. Columns Subjected to Axial Load Only.

Although uniaxially loaded compression members are not permitted in design codes,the strength of such columns should be known since they represent a limit. The ultimate strength of an axially loaded column can be expressed as the sum of the yield strength of steel, plus the crushing strength of concrete. Since concrete in the column is not as good as the concrete in the cylinder, concrete strength is usually multiplied by a reduction factor, 0.85 .

$$N_{or} = 0.85 f_{ck} (A_c - A_{st}) + A_{st} f_{yk} \quad ....(42)$$

Where:

f_{ck} = characteristic compressive Strength of Concrete.

f_{yk} = characteristic yield strength of the longitudinal Steel.

A_c = cross sectional area of the member.

A_{st} = total area of longitudinal steel bars.

This equation represents the strength of both tied and spiral columns with respect to the first peak.(before shell crushing). In practice or even in the labratory it is not possible to have pure axial load. Due to the monolithic nature of concrete construction, imperfection in geometry and unisotropy of concrete, there is always some eccentricity.

3.3. Columns Subjected to Axial Load and Uniaxial Bending.

The design moment and design axial force can be replaced by an axial load applied with an eccentricity of $e = M_d / N_d$. A reinforced concrete section subjected to an axial force and uniaxial bending is shown in Fig. 3.1. As shown, the cross-section has a symmetry about the plane of loading. This is a must for a uniaxial loading, and under this condition the neutral axis is perpendicular to the plane of loading. Note that the plane of loading is perpendicular to section and passes through the both plastic centroid and point of eccentricity.Ultimate strength for such cases can be obtained by writing proper equilibrium, compatibility and force-deformation equations.

Equilibrium:

$$N = F_c + \sum_{i=1}^{n} A_{si}\, \sigma_{si} \qquad \text{......................(43)}$$

Moment about the centroid.

$$M = N \cdot e = F_c\,(X_p - \bar{X}) + \sum_{i=1}^{n} A_{si}\, \sigma_{si}\, X_i \qquad \text{.......(44)}$$

Compatibility:

From the strain triangles:

$$\frac{c}{\epsilon_{cu}} = \frac{X_p - c - X_i}{-\epsilon_{si}} \qquad \text{.............(45)}$$

Force Deformation:

$$\sigma_{si} = \epsilon_{si}\, E_s \leq f_{yk} \qquad \text{.................(46)}$$

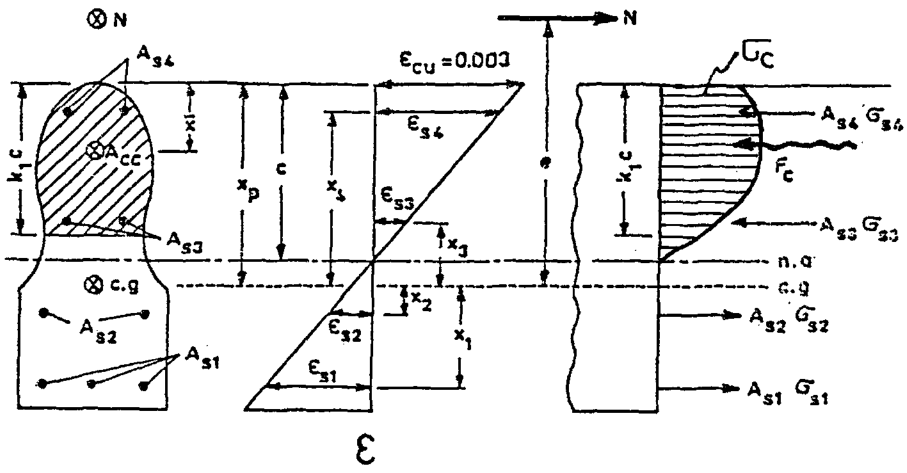


Fig.3.1. Section Subjected to Axial Load and Uniaxial Bending.[19]

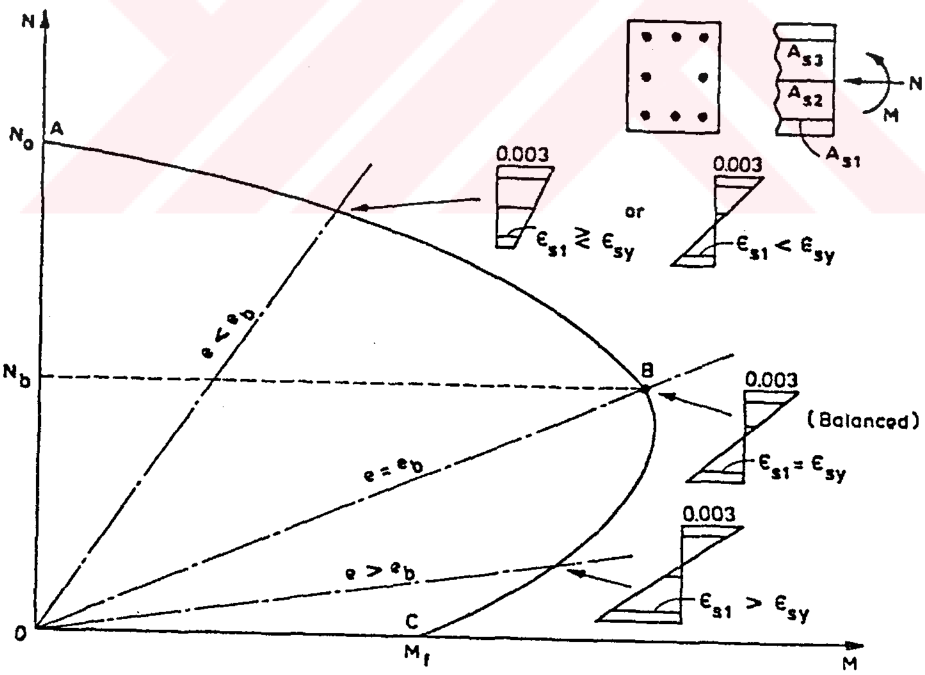


Fig. 3.2. Interaction Diagram of Sections Under Axial Load and Uniaxial Bending.[19]

Or by substituting Eq.(45) into Eq.(46),

$$\sigma_{si} = \epsilon_{cu} E_s \left(1 + \frac{x_i - x_p}{c} \right) \leq f_{yk} \quad \ldots\ldots\ldots(47)$$

Where:

F_c = resultant force in concrete

A_{si} = steel area at i'th level

σ_{si} = steel stress at i'th level

x_p = distance to the plastic centroid

$\bar{x}$ = distance from extreme compression fiber to the resultant force in concrete

x_i = distance from plastic centroid to the steel at i'th level

c = depth of neutral axis

ϵ_{cu} = maximum strain permitted for concrete. (TS-500 , ϵ_{cu}=0.003)

ϵ_{si} = strain in steel at i'th level

E_s = modulus of elasticity of steel

It should be noted that no restrictions have been placed on the geometry of the cross-section and on the arrangement of steel, except symmetry about plane of loading. In design, design strengths are used instead of characteristic strengths, i.e. f_{cd} and f_{yd} in place of f_{ck} and f_{yk} . The equations are applicable all the way from $e = 0$ (axial Compression) to $e = \infty$ (pure Bending). In other words, these three simple equations i.e. Eq.(43), (44) and (47) represent the ultimate capacity of a reinforced cross-section subjected to axial load and flexure in all ranges of eccentricity. For a given cross-section, the plot of N versus M obtained from these equations, will represent the strength envelope. Strength envelopes are usually called "Interaction diagrams" or "Interaction curves", Fig. 3.2. Two types of failures are possible, " Compression " and " Tension " failures. In between there is a transition point which is called " Balanced failure ". Portion AB of interaction curve represents compression failure. Two types of strain distributions are possible for this zone. One representing no tension on the section and the other one with tension. This type of failure is brittle, because when the extreme fiber in compression reaches the crushing strain, the nearest steel to the tension face has not yielded yet. Therefore failure is governed by the characteristics of concrete. In case of tension failure, which is represented by portion BC , the nearest steel to the tension face has already yielded when concrete at the extreme compression fiber reaches the crushing strain. This is a ductile type of failure because it is governed by the characteristics of steel. Point B represents the transition point or the balanced failure. The nearest Steel to the tension face reaches the yield strain simultaneously with concrete reaching the crushing strain. The type of failure can be determined by comparing either axial loads or eccentricities as shown below.

Compression failure:

if $e < e_b$

or if $N > N_b$

Tension failure:

if $e > e_b$

or if $N < N_b$

3.4. Columns Subjected to Axial Load and Biaxial Bending.

In practice a great majority of the columns are subjected to biaxial bending in adition to axial loads. In many cases the smaller moment along one axis is neglected and such columns are designed for uniaxial bending. However, some columns, especially the corner columns, are subjected to significant bending in both directions and should be designed or analyzed considering the biaxial moments. In general the neutral axis in these columns is not perpendicular to the resultant eccentricity. For members subjected to axial load and biaxial bending, the interaction diagram is replaced by an interaction surface (Failure Surface), Fig. 3.4.(a). Each point on this surface represents a set of axial load N and moments about the major axes (M_x,M_y). Any horizontal section taken through the interaction surface, will be a curve which represents possible M_x , M_y combinations under a given load N, Fig.3.4.(b). These curves are interaction curves for M_x and M_y.

For a given cross-section with specified material strengths, each interaction surface corresponds to a given amount of steel, and will be referred to as an "isobaric failure surface". For convenience interaction volumes can be defined as the set of isobaric failure surfaces obtained by varying the amount of reinforcement proportionally in a given cross-section. It may be useful to think of the interaction volumes as successive hulls surrounding a core, as in an onion where the hulls are layers of isobaric failure surfaces, and the core represents what unreinforced concrete resists. The upper layer corresponds to the maximum reinforcement that can be permitted by codes.

The ultimate strength of a reinforced concrete cross-section subjected axial load and biaxial bending can be obtained by writing proper equilibrium, compatibility and force-deformation equations, similar to uniaxial bending. The main differences are; two moment equilibrium equations are written for biaxial bending and the position of the neutral axis is defined by two variables instead of one, Fig. 3.3.

In writing the equilibrium and force-deformation equations one has to use proper models to express the stress-strain relations for concrete and steel. These will be discussed in next chapter.

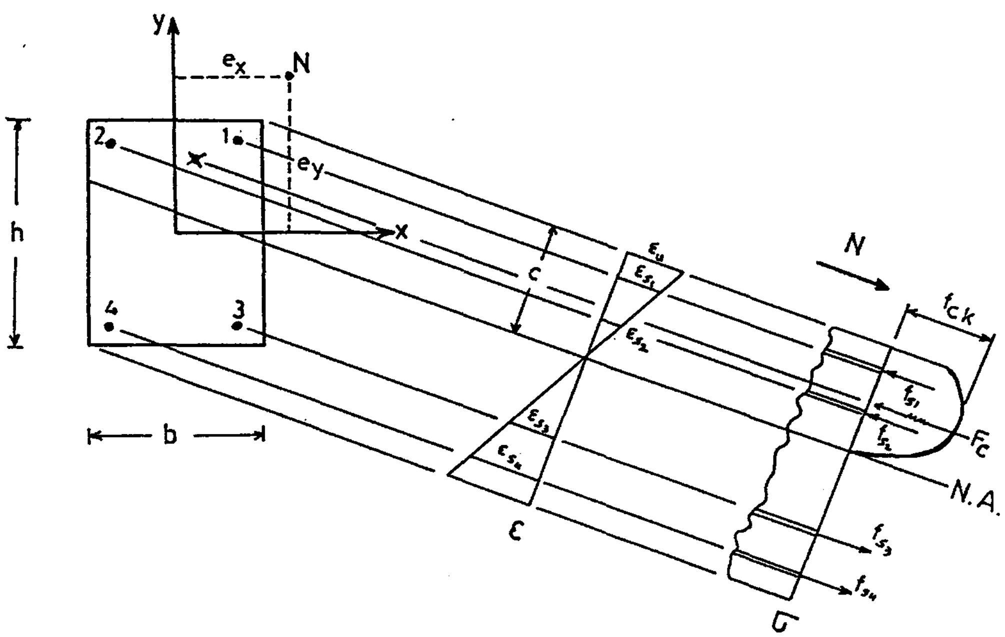
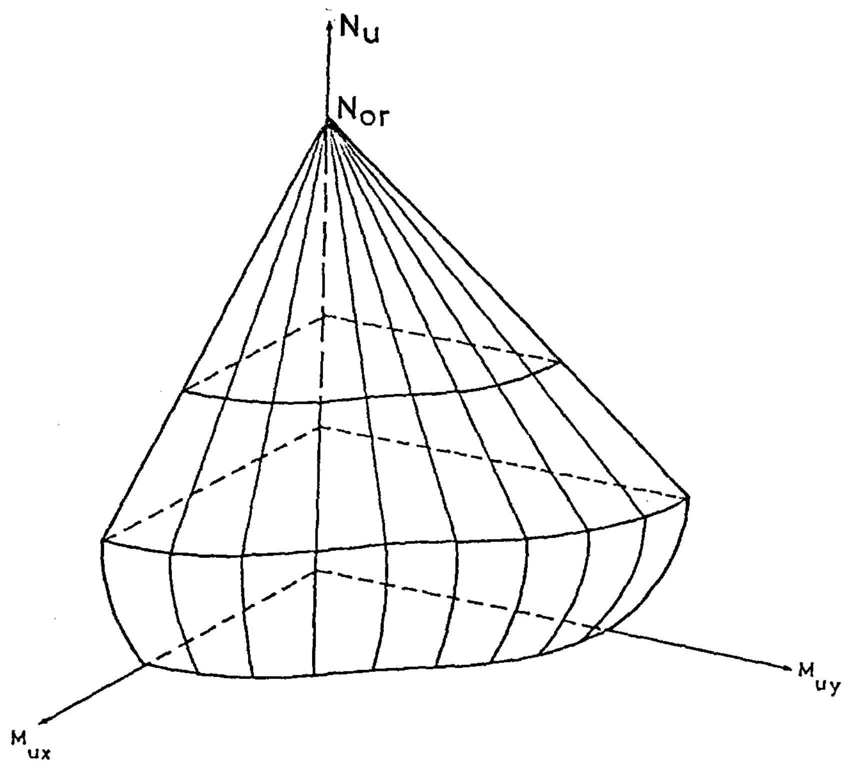
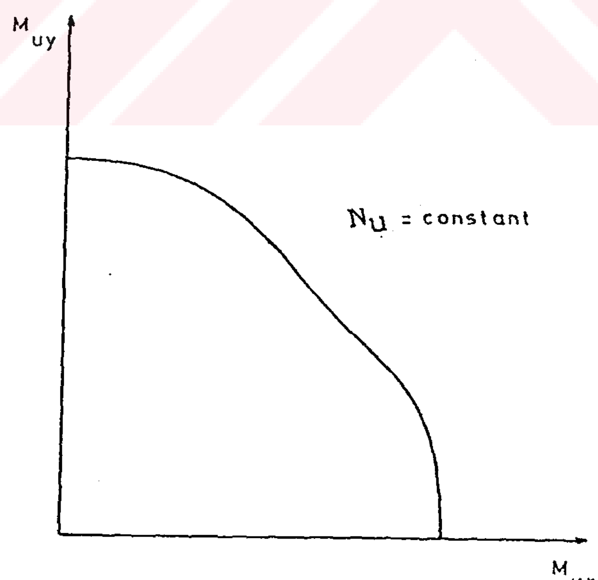


Fig.3.3. Column Section Under Axial Load and Biaxial Bending.



(a) Interaction Surface of Sections Under Axial Load and Biaxial Bending.[15]



(b) Moment Interaction Curve Under a Constant Load.[15]

Fig. 3.4. Interaction Surface and Moment Interaction.

CHAPTER 4

MATERIAL MODELS

4.1. General.

In this chapter various alternatives for concrete stress-strain relationship are presented. The relationship used for steel is also briefly discussed. Since the design problem is investigated, the design values instead of characteristic values will be used in expressing stress-strain relations.

In tables 4.1 and 4.2 the characteristic and design strength values for different concrete and steel classes as described in TS-500 are presented.

4.2. Stress-Strain Relationship For Concrete.

The compressive strength and stress-strain relationship of concrete are usually obtained from tests on standard cylinders (150 x 300 mm). The cylinders are longitudinally loaded at a slow strain rate to reach the maximum stress in 2 or 3 minutes. The stress-strain relationship of concrete is not unique and depends on many variables. Therefore, no stress-strain relation presented as a mathematical model can be excact.

Table 4.1. Material Properties of Concrete According to TS-500.

Concrete Class		k1	Characteristic Strength N/mm^2 (kg/cm^2)		Design Strength N/mm^2 (kg/cm^2)		Equivalent Cube Strength	Modulus of Elasticity
			fck (comp.)	fckt (tens.)	fcd (comp.)	fctd (tens.)	N/mm^2 (kg/cm^2)	kN/mm^2 (t/cm^2)
Normal concrete	BS14 (C14)	0.85	14 (140)	1.3 (13)	9.5 (95)	0.85 (8.5)	16 (160)	26.15 (261.5)
	BS16 (C16)	0.85	16 (160)	1.4 (14)	11 (110)	0.90 (9.0)	20 (200)	27.00 (270.0)
	BS20 (C20)	0.85	20 (200)	1.6 (16)	13 (130)	1.00 (10.0)	25 (250)	28.50 (285.0)
	BS25 (C25)	0.85	25 (250)	1.8 (18)	17 (170)	1.15 (11.5)	30 (300)	30.25 (302.5)
High strength concrete	BS30 (C30)	0.82	30 (300)	1.9 (19)	20 (200)	1.25 (12.5)	35 (350)	31.80 (318.0)
	BS35 (C35)	0.79	35 (350)	2.1 (21)	23 (230)	1.35 (13.5)	40 (400)	33.20 (332.0)
	BS40 (C40)	0.76	40 (400)	2.2 (22)	27 (270)	1.45 (14.5)	45 (450)	34.55 (345.5)
	BS45 (C45)	0.73	45 (450)	2.3 (23)	30 (300)	1.55 (15.5)	50 (500)	35.80 (358.0)
	BS50 (C50)	0.70	50 (500)	2.5 (25)	33 (330)	1.65 (16.5)	55 (550)	36.95 (369.5)

Table 4.2. Mechanical Properties of Reinforcing Steel According to TS-500.

Steel Grade	Yield Strength f_{yk} N/mm² (kg/cm²)	Design Strength f_{yd} N/mm² (kg/cm²)	Ult. Strength f_{su} N/mm² (kg/cm²)	Manuf. Proc.	Surface Geometry
BÇ I	220 (2200)	191 (1910)	340 (3400)	Hot rolled	Plain
BÇ III a	420 (4200)	365 (3650)	500 (5000)	Hot rolled	Deformed
BÇ III b	420 (4200)	365 (3650)	500 (5000)	Cold worked	Deformed
BÇ IV a	500 (5000)	435 (4350)	550 (5500)	Hot rolled	Deformed or Plain
BÇ IV b	500 (5000)	435 (4350)	550 (5500)	Cold worked	

Note : a) $E_s = 2 \times 10^5$ N/mm² (2×10^6 kg/cm²)

b) For Steel grades, instead of BÇI, BÇIII etc. one can use:

BÇ-I S220
BÇ-III S420
BÇ-IV S500

Various models have been presented for the stress-strain relationship of concrete. Here only the most commonly used two, Kent and Park and Hognestad models will be discussed.

In design, usually simplified stress blocks are used instead of stress-strain models. Most commonly used simplified blocks are, rectangular block and CEB rectangular-parabola. Here, these simplified block will be discussed first.

Basic properties of the stress blocks or stress-strain relations are summarized in table 4.3. Design values used by the author in the computer program developed during this study are given in table 4.4.

4.2.1. Equivalent Rectangular Stress-Block.

This stress-block is recommended by both ACI-318 and TS-500 codes. The maximum concrete compresive strain is taken as 0.003 , and a uniform Stress of $0.85\ f_{cd}$ in TS-500 and $0.85\ f_{ck}$ in ACI is assumed to exist. In TS-500 the depth of uniform stress block is specified as $a = k_1 * c$ where c is the depth of neutral axis. The k_1 coefficient is specified for each concrete class in a table. In the ACI code depth of the block is taken as $\beta_1 * c$ and β_1 is expressed as a function of concrete strength, Fig. 4.1.

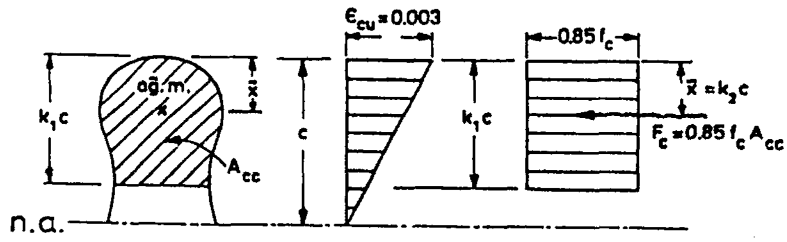
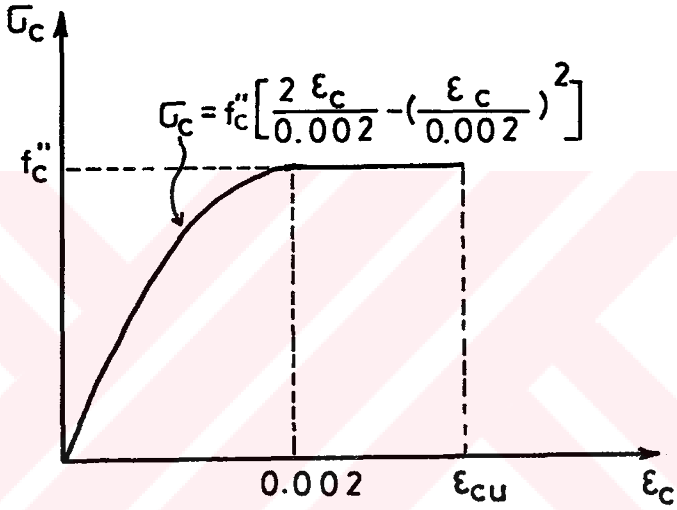
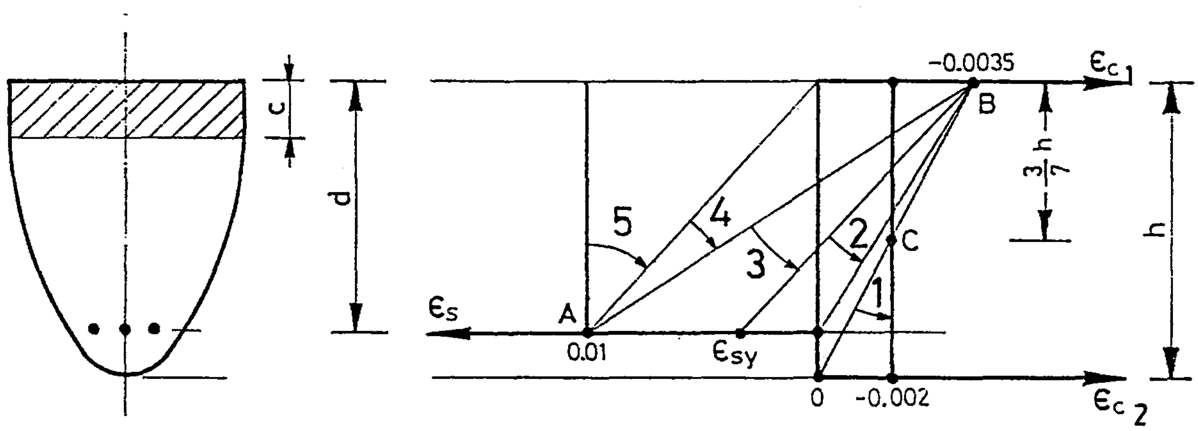


Fig.4.1. Rectangular Stress-Block.



(a) CEB Stress-Block.



(b) Strain Diagrams in CEB .

Fig. 4.2. CEB Stress-Block and Limiting Strains.

$$\text{Depth of Stress-Block} = \beta_1 \cdot c \qquad \ldots\ldots\ldots\ldots\ldots(48)$$

Where:

$$\beta_1 = 0.85 \qquad \text{for} \quad f_{ck} \leq 4000 \text{ psi}$$

$$\beta_1 = 0.85 - \frac{(f_{ck}-4000)}{1000} * 0.05 \qquad \text{for} \quad f_{ck} > 4000 \text{ psi}$$

c = depth of the neutral axis measured from the top compressive fiber.

4.2.2. CEB Stress-Block.

In the CEB model code , stress-block is approximated by a parabola-rectangle. The first portion which ends at a strain value of 0.002 is a second degree parabola. The second portion is linear (rectangle) where the stress remains constant up to the maximum strain, Fig. 4.2.(a). Maximum strain is 0.0035 for pure flexure and decreses with the level of axial load down to 0.002 under high levels of axial load as shown in Fig. 4.2.(b).

4.2.3. Kent & Park Stress-Strain Relationship.

Kent and Park tried to represent the stress-strain relationship of concrete by a parabola and a straight line. The ascending part is represented by a second degree parabola and the descending part is represented by a linear curve. The ultimate strain of concrete is not specified. This model can also represent the confined concrete. To do this the slope of the descending portion is changed, Fig. 4.3.

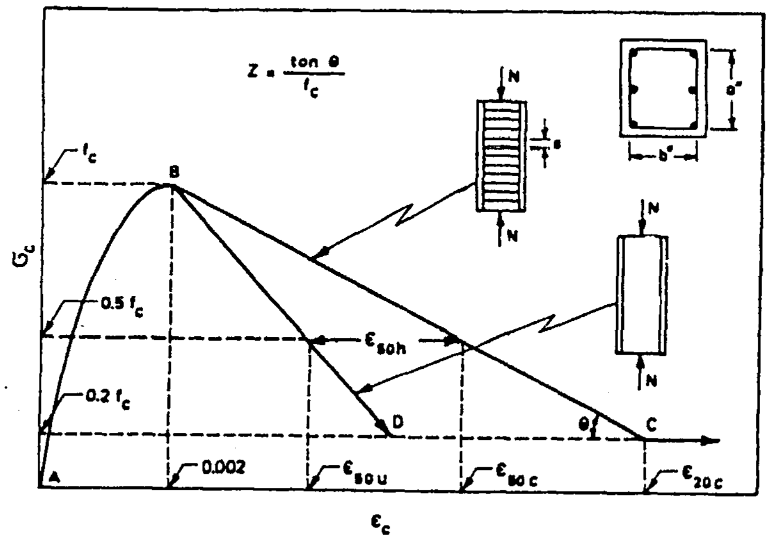


Fig. 4.3. Kent & Park Stress-Strain Relationship.

Region A-B :

$$\sigma_c = f_c \left\{ \frac{2\,\epsilon_c}{0.002} - \left(\frac{\epsilon_c}{0.002}\right)^2 \right\} \quad \ldots\ldots\ldots(49)$$

Region B-C and B-D :

$$\sigma_c = f_c \{ 1 - Z (\epsilon_c - 0.002)^2 \} \quad \ldots\ldots\ldots(50)$$

$$\epsilon_{50u} = \frac{3 + 0.0285 f_c}{14.2 f_c - 1000} \quad \ldots\ldots\ldots(51)$$

$$\epsilon_{50h} = \frac{3}{4} \rho_s \frac{b''}{s} \quad \ldots\ldots\ldots(52)$$

$$Z = \frac{0.5}{\epsilon_{50u} + \epsilon_{50h} - 0.002} \quad \ldots\ldots\ldots(53)$$

$$\rho_s = \frac{A_o (a'' + b'') 2}{s (a'') (b'')} \quad \ldots\ldots\ldots(54)$$

Where:

σ_c = concrete stress

f_c = maximum concrete stress. ($f_c = f_{ck}$)

and the other parameters are as described in Fig. 4.3. In the above equations stress or strength should be written in Kgf/cm^2 ; dimensions in cm and areas in cm^2.

4.2.4. Hognestad Stress-Strain Relationship.

The Hognestad's stress-strain curve is also composed of two parts, an initial second degree parabola and a linear portion for the descending branch (See Fig. 4.4). The maximum stress is assumed to be 0.85 f_{ck} and a maximum strain in concrete is defined as 0.0038 .

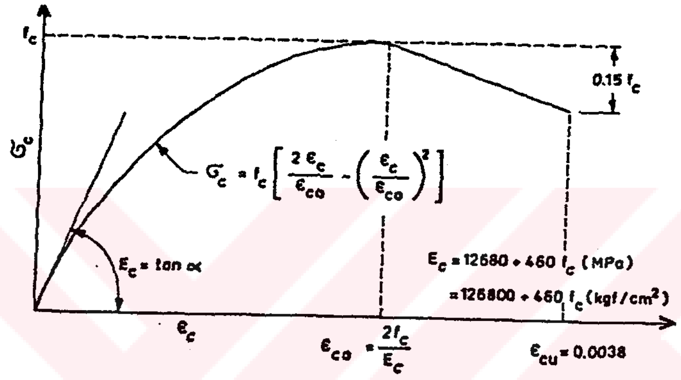
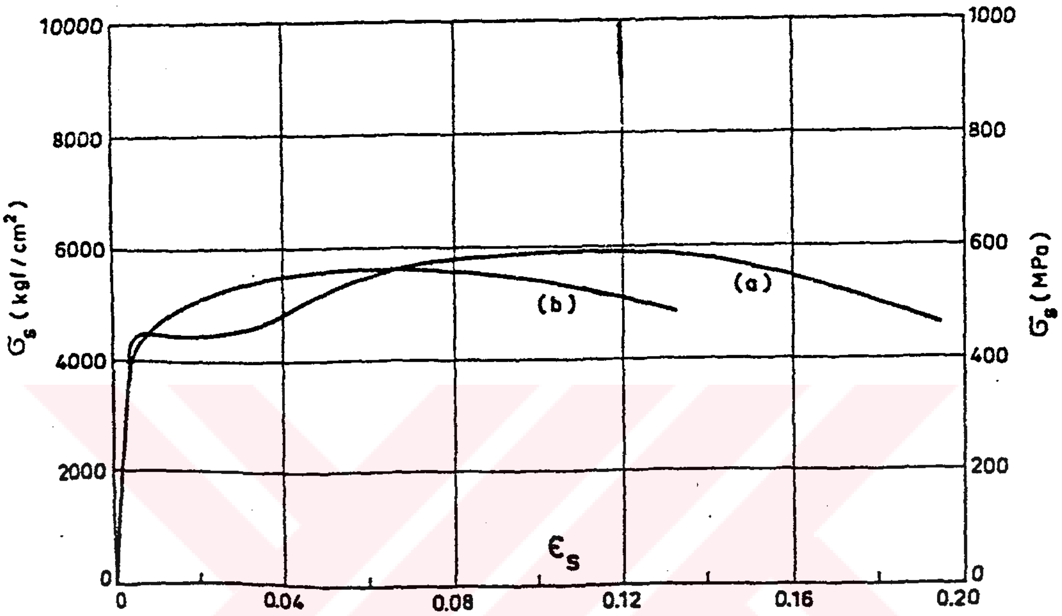


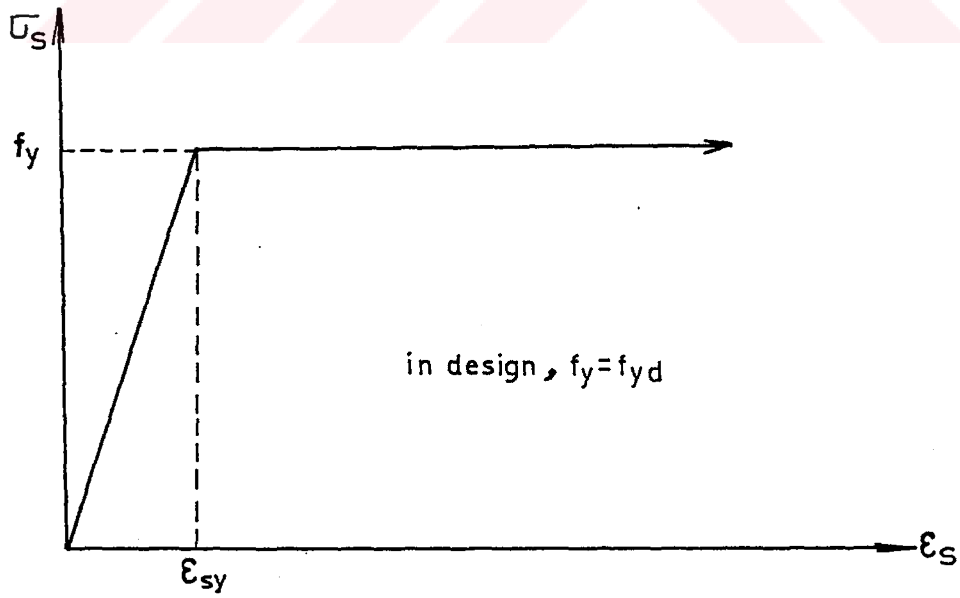
Fig. 4.4. Hognestad Stress-Strain Relationship.

4.3. Stress-Strain Relationship for Reinforcing Steel.

Stress-strain curves for reinforcing steel bars (Fig. 4.5.(a)), are obtained from tests on steel bars loaded in uniaxial tension. For hot rolled steel, the curves have an initial linear elastic portion, a clear yield point followed by a yield plateau. This yield plateau is followed by a strain hardening region, prior to rupturing of steel. For design purposes only the first two part, elastic region and yield plateau are usually taken into consideration. So, under monotonic loading stress-strain relationship is assumed to be elasto-plastic as shown in Fig.4.5.(b). This model simulates the $\sigma-\epsilon$ relationship of hot rolled steel very well. For cold worked steel which does not have any definite yield plateau ,the model is more approximate.



(a) Stress-Strain Curves for Reinforcing Steel.



(b) Elasto-Plastic Idealization of Reinforcing Steel.

Fig. 4.5. Stress-Strain Relationship for Steel.

TABLE 4.3. Comparison of Stress-Strain Models and Simplified Stress Blocks.

STRESS-STRAIN MODEL	Rectangular	C E B	Kent & Park	Hognestad
Maximum Stress	$.85\ f_{ck}$	$.85\ f_{ck}$	$1.0\ f_{ck}$	$.85\ f_{ck}$
Strain at Maximum Stress, ϵo	.002	.002	.002	$1.7\ f_{ck}/E_c$
Maximum Strain, ϵcu	.003	.002-.0035	---	.0038
Stress at Maximum Strain	$.85\ f_{ck}$	$.85\ f_{ck}$	——	$.723\ f_{ck}$

Note: All models in this table are for unconfined concrete.

TABLE 4.4. Design Values for Stress-Strain Models and Simplified Blocks Used in the Computer Program.

STRESS-STRAIN MODEL	Rectangular	C E B	Kent & Park	Hognestad
Maximum Stress	$.85\ f_{cd}$	$.85\ f_{cd}$	$1.0\ f_{cd}$	$.85\ f_{cd}$
Strain at Maximum Stress, ϵo	.002	.002	.002	$1.7\ f_{ck}/E_c$
Maximum Strain, ϵcu	.003	.002-.0035	___(*)	.0038
Stress at Maximum Strain	$.85\ f_{cd}$	$.85\ f_{cd}$	——	$.723\ f_{cd}$

(*) In the program any maximum strain can be input. However in the examples solved ϵ_{cu} = 0.003 was used.

CHAPTER 5

COMPUTER PROGRAM DEVELOPED

5.1. General.

The ultimate strength of reinforced concrete cross-sections subjected to biaxial bending moments in addition to axial force can be obtained by writing proper equilibrium, compatibility and force-deformation equations. In order to be able to write these equations the following assumptions have to be made:

1- Concrete can not carry any tension.

2- There is a perfect bond between steel and concrete.

3- Plane sections remain plane after bending (linear strain distribution)

4- Behavior of reinforcing steel is elastoplastic.
$\sigma_s = \epsilon_s E_s \leq f_{yd}$

5- Stress distribution in the compression zone is similar to the σ-ϵ relationship obtained from uniaxial tests. Any reasonable stress-strain curve or simplified equivalent block can be used.

6- Crushing strain for concrete has to be specified, ϵ_{cu}.

The first three assumptions need no discussion, they are even used in working stress design. The fourth assumptions about the behavior of steel was discussed in chapter 4. Here only the last two assumptions will be discussed.

Stress Block :

The effect of the shape of the stress-block was investigated by E. Çokça [15]. It is concluded that since the influence of concrete on the overall behavior is not very significant for low axial load levels, effect of different concrete distributions do not greatly alter the moment capacity prediction for axial loads below the balanced value. But as the axial load level reaches higher values, differences in the predicted moment capacity due to different concrete stress distributions are observed. Since for design purposes the only two important properties of the stress block are its area and centroid, any two blocks having identical area and centroid will yield identical results regardless of the shape of the block.

Limiting Concrete Strain :

Crushing or limiting strain for concrete depends on many variables, most important of which are; (a) shape of the cross-section, (b) rate of loading, (c) percentage of confining reinforcement and (d) strain gradient. E. Çokça studied the effect of (a), (c) and (d) by making a parametric study. Using a general stress-strain relationship (Kent & Park) the ultimate moments corresponding to different ultimate strain values assumed for sections subjected to axial load and biaxial bending are computed. By plotting these points (M versus ϵ_{cu}) the optimum ϵ_{cu} which would yield the maximum moment were predicted, Fig. 5.1.

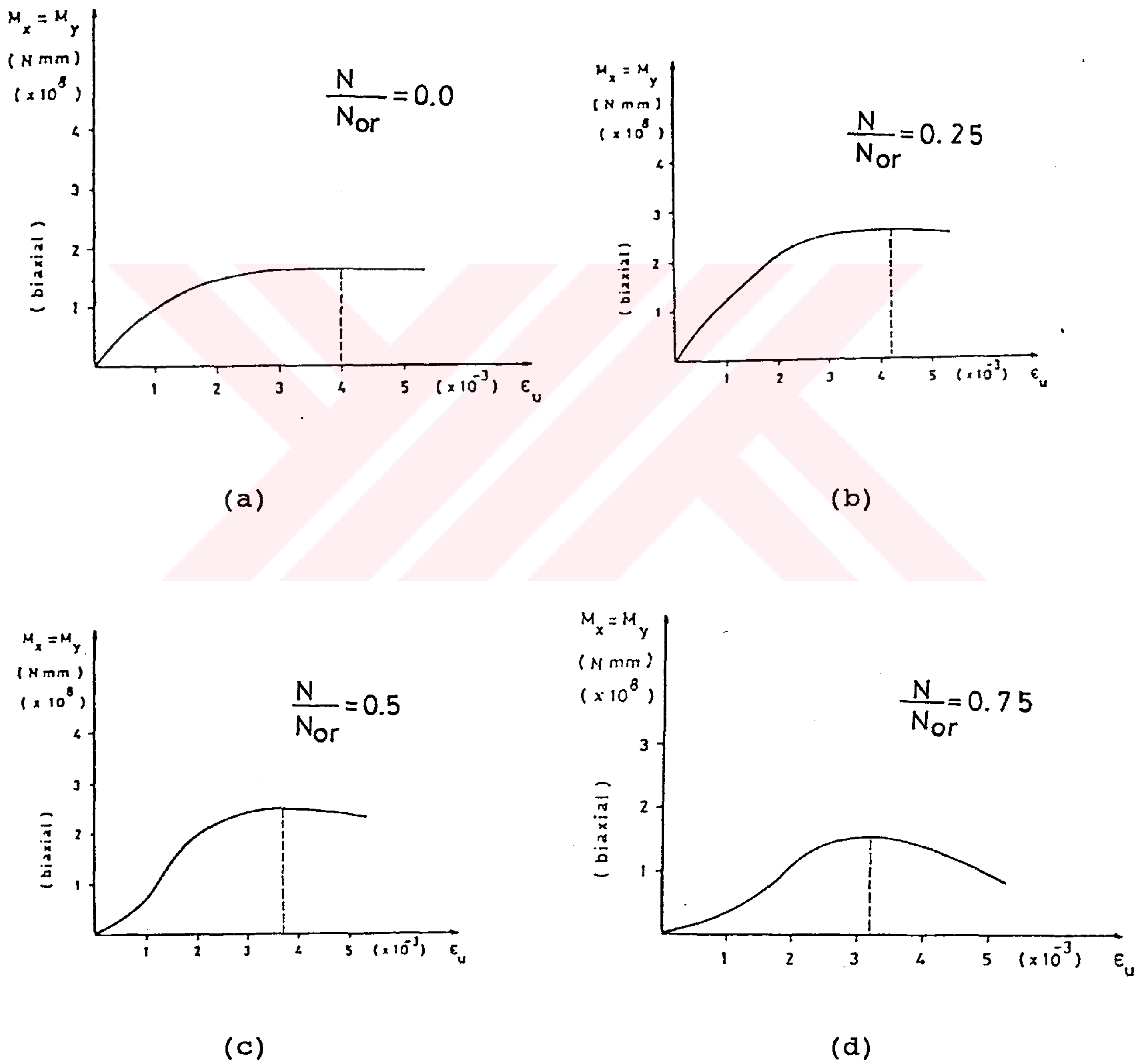


Fig. 5.1. Ultimate Strain - Maximum Diagonal Biaxial Bending Capacity Relationship (Rectangular Section).[15]

Since different levels of axial load were considered, the influence of strain gradient and shape of the cross-section were automatically included in calculations. Çokça made the same study for different cross-sectional shapes (T, triangle etc.).

As can be seen from Fig. 5.1 the strain corresponding to the peak value of the moments represent the crushing strain to be used. Although the optimum strain varied significantly with the variables considered, the difference in moment corresponding to 0.003 , 0.0035 or even 0.004 was not significant. Therefore it is concluded that for design purposes it is quite feasible to use ϵ_{cu} = 0.003 although the real value could be different.

5.2. Basic Principles.

In the design of columns subjected to biaxial moments in addition to axial force, the main objective is to find the steel area needed for a given geometry of cross-section and steel configuration. To do this, one has to know the position of the neutral axis. The parameters that are chosen to describe the position of the neutral axis are $\overline{Y}$ and α_{ne} as presented in Fig.5.2 $\overline{Y}$ is the depth of neutral axis, i.e. the distance from the point (x_{max}, y_{max}) to the intersection of the neutral axis and the line $y=y_{max}$. The angle α_{ne} is the angle that neutral axis makes with the line $Y=Y_{max}$. Therefore it can be said that internal forces N , M_x , M_y are the functions of position of neutral axis and position and amount of reinforcement. If the position and the ratio of each steel area to the total area of steel is known, the unknowns will be only ($\overline{Y}$, α_{ne} , A_{st}), where A_{st} is the total area of steel in that cross-section. From the equilibrium conditions, i.e. internal forces are equal to external loads, the following equations can be obtained:

$$F_1 = f_1 (\overline{Y} , \alpha_{ne} , A_{st}) - N_d = 0 \qquad ....(55)$$

$$F_2 = f_2 (\overline{Y} , \alpha_{ne} , A_{st}) - M_{xd} = 0 \qquad ....(56)$$

$$F_3 = f_3 (\overline{Y} , \alpha_{ne} , A_{st}) - M_{yd} = 0 \qquad ....(57)$$

The problem is reduced to the solution of these three simultaneous equations. The Newton-Raphson Method was used to solve these equations. It should be noted that in finding the force in each steel bar, compatibility equations are needed to find the strains and force-deformation relation is needed to find the corresponding stresses. In writing the compatibility equations one has to know ϵ_{cu}. In finding the concrete resultant force stress block should be known. Therefore one has to make assumptions about the concrete stress distribution and crushing strain of concrete.

5.3. Newtons Iterative Method.

Suppose it is required to obtain a first approximation to a certain root $X=\alpha$ of an equation of the form $f(x)=0$ and a more nearly accurate value is required. If the first approximation is denoted by X_o , then attempt to determine ΔX so that $f(X + \Delta X) = 0$. For small values of ΔX the Taylor series expansion can be written as:

$$f(X_o+\Delta X) = f(X_o) + \Delta X\, f'(X_o) + \frac{(\Delta X)^2}{2} f''(X_o) + \ldots \qquad (58)$$

If the initial approximation is sufficiently accurate and if the higher derivatives of f are not excessively large at X_o , the terms involving higher powers of ΔX in the equation $f(X_o+\Delta X) = 0$ can be neglected and hence obtain for ΔX_o the following approximation:

$$f(X_o) + \Delta X_o\, f'(X_o) = 0 \qquad (59)$$

Therefore:

$$\Delta X_o = - \frac{f(X_o)}{f'(x_o)} \qquad (60)$$

The next aproximation for x is then taken as:

$$X_1 = X_o + \Delta X_o \qquad (61)$$

And the process is repeated.

Geometrically, this procedure consists of approximating the curve representing $y = f(x)$ by its tangent line at $X = X_o$, and of determining the intersection of the tangent line with the X axis.(See Fig. 5.3.)

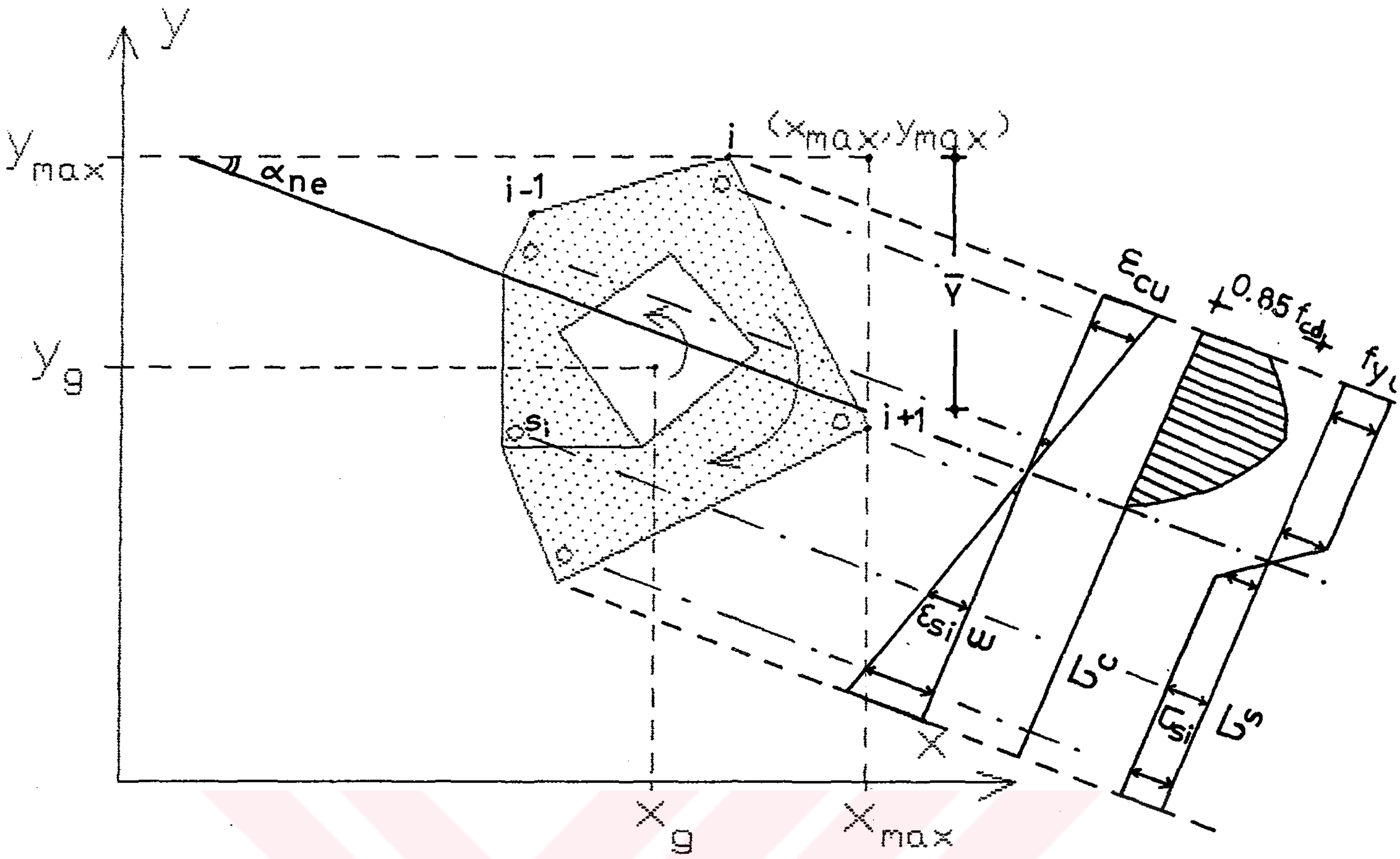


Fig. 5.2. Cracked Polygonal Reinforced Concrete Cross-Section.

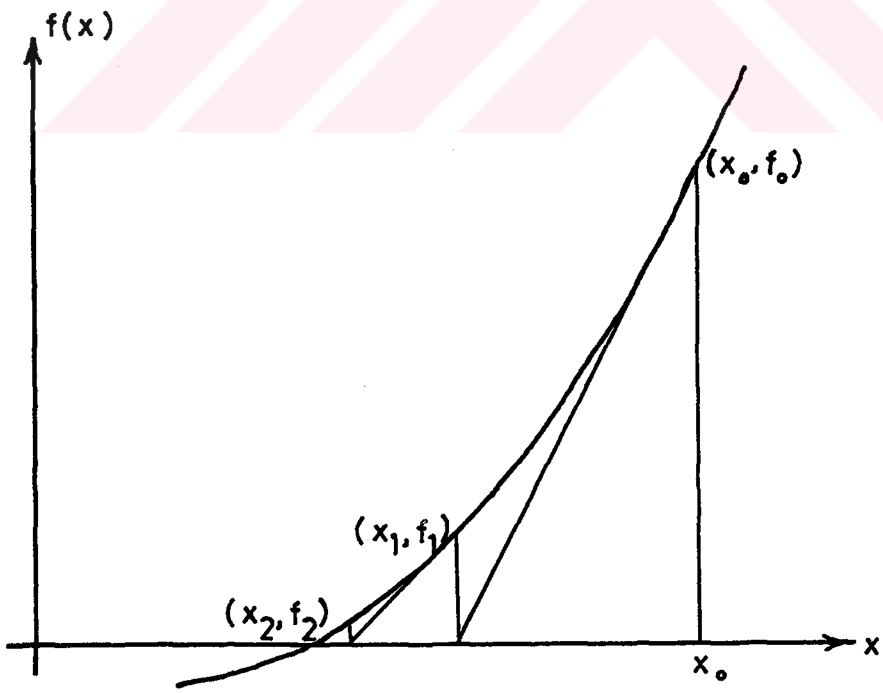


Fig. 5.3. Geometric Representation of Newton-Raphson Method.

5.3.1. Application of Newton-Raphson Method.

Equations (55,56,57) can be presented in the following form.

$$F(\overline{Y}, \alpha, A) = 0 \quad \ldots\ldots(62)$$

$$G(\overline{Y}, \alpha, A) = 0 \quad \ldots\ldots(63)$$

$$H(\overline{Y}, \alpha, A) = 0 \quad \ldots\ldots(64)$$

Where:

$$\alpha = \alpha_{ne}$$

and

$$A = A_{st}$$

Solution of these simultaneous equations is required. Suppose a reasonablly accurate initial approximation ($\overline{Y}_o$, α_o, A_o) has been obtained by some method. Next an attempt is made to determine values of $\Delta\overline{Y}$, $\Delta\alpha$, ΔA, such that the equations:

$$F(\overline{Y}_o+\Delta\overline{Y}, \alpha_o+\Delta\alpha, A_o+\Delta A) = 0 \quad \ldots\ldots(65)$$

$$G(\overline{Y}_o+\Delta\overline{Y}, \alpha_o+\Delta\alpha, A_o+\Delta A) = 0 \quad \ldots\ldots(66)$$

$$H(\overline{Y}_o+\Delta\overline{Y}, \alpha_o+\Delta\alpha, A_o+\Delta A) = 0 \quad \ldots\ldots(67)$$

are simultaneously satisfied. If the left-hand members are expanded in Taylor Series about the initial point and if only linear terms are retained, these equations become:

$$F_o + \Delta\overline{Y} \cdot F_{\overline{y}o} + \Delta\alpha \cdot F_{\alpha o} + \Delta A \cdot F_{Ao} = 0 \quad \ldots\ldots(68)$$

$$G_o + \Delta\overline{Y} \cdot G_{\overline{y}o} + \Delta\alpha \cdot G_{\alpha o} + \Delta A \cdot G_{Ao} = 0 \quad \ldots\ldots(69)$$

$$H_o + \Delta\overline{Y} \cdot H_{\overline{y}o} + \Delta\alpha \cdot H_{\alpha o} + \Delta A \cdot H_{Ao} = 0 \quad \ldots\ldots(70)$$

The zero subscripts indicate that the functions involved are evaluated at the point ($\bar{Y}_o$, α_o, A_o) and the $\bar{Y}$, α , A subscripts indicate that the function is derivative with respect to $\bar{Y}$, α , A. Thus the approximate corrections $\Delta\bar{Y}_o$, $\Delta\alpha_o$ and ΔA_o are given by the solution of these equations in the form:

$$\Delta\bar{Y}_o = - \frac{\begin{vmatrix} F_o & F_{\alpha o} & F_{Ao} \\ G_o & G_{\alpha o} & G_{Ao} \\ H_o & H_{\alpha o} & H_{Ao} \end{vmatrix}}{\dfrac{\partial(F,G,H)_o}{\partial(\bar{Y},\alpha,A)}} \qquad \text{......(71)}$$

$$\Delta\alpha_o = - \frac{\begin{vmatrix} F_{\bar{y}o} & F_o & F_{Ao} \\ G_{\bar{y}o} & G_o & G_{Ao} \\ H_{\bar{y}o} & H_o & H_{Ao} \end{vmatrix}}{\dfrac{\partial(F,G,H)_o}{\partial(\bar{Y},\alpha,A)}} \qquad \text{......(72)}$$

$$\Delta A_o = - \frac{\begin{vmatrix} F_{\bar{y}o} & F_{\alpha o} & F_o \\ G_{\bar{y}o} & G_{\alpha o} & G_o \\ H_{\bar{y}o} & H_{\alpha o} & H_o \end{vmatrix}}{\dfrac{\partial(F,G,H)_o}{\partial(\bar{Y},\alpha,A)}} \qquad \text{......(73)}$$

Where:

$$\frac{\partial(F,G,H)_o}{\partial(\bar{Y},\alpha,A)} \quad \text{is} \quad \begin{vmatrix} F_{\bar{y}o} & F_{\alpha o} & F_{Ao} \\ G_{\bar{y}o} & G_{\alpha o} & G_{Ao} \\ H_{\bar{y}o} & H_{\alpha o} & H_{Ao} \end{vmatrix} \quad \text{or the JACOBIAN matrix.} \qquad \text{.......(74)}$$

All of these simultaneous equations can be written in a matrix form.

$$\begin{bmatrix} F_{yo} & F_{ao} & F_{Ao} \\ G_{yo} & G_{\alpha o} & G_{Ao} \\ H_{yo} & H_{\alpha o} & H_{Ao} \end{bmatrix} \cdot \begin{bmatrix} \Delta \overline{Y}_o \\ \Delta \alpha_o \\ \Delta A_o \end{bmatrix} = - \begin{bmatrix} F_o \\ G_o \\ H_o \end{bmatrix} \qquad \text{.........(75)}$$

It is seen that the success of this method depends in part upon the magnitude of the jacobian determinant in the neighborhood of the desired solution.

$$\underline{J}_o \cdot \Delta \underline{X}_o = -\underline{F}_o \qquad \text{...................(76)}$$

For the i'th term the equation will be;

$$\underline{J}_i \cdot \Delta \underline{X}_i = -\underline{F}_i \qquad \text{...................(77)}$$

or:

$$\Delta \underline{X}_i = -\underline{J}_i^{-1} \cdot \underline{F}_i \qquad \text{...................(78)}$$

If the Jacobian matrix is shown in terms of original notations, it becomes:

$$\underline{J} = \begin{bmatrix} \frac{\partial F_1}{\partial Y} & \frac{\partial F_1}{\partial \alpha_{ne}} & \frac{\partial F_1}{\partial A_{st}} \\ \frac{\partial F_2}{\partial Y} & \frac{\partial F_2}{\partial \alpha_{ne}} & \frac{\partial F_2}{\partial A_{st}} \\ \frac{\partial F_3}{\partial Y} & \frac{\partial F_3}{\partial \alpha_{ne}} & \frac{\partial F_3}{\partial A_{st}} \end{bmatrix} \qquad \text{.......(79)}$$

To obtain this Jacobian matrix each derivative is calculated by first order numerical differentiation method. i.e.

$$\frac{\partial F}{\partial X} = \frac{F(X + \Delta X) - F (X)}{\Delta X} \qquad \text{........(80)}$$

Using equation (75) and other formulations, the algorithms of the computer program will be as the flow-chart shown in Fig. 5.4.

5.4. Computer Program.

The aim of the computer program developed is to find a position for the neutral axis and a value for the steel area which satisfy the equilibrium conditions under the given design loads. The position of the neutral axis is defined by two parameters. One is the depth and the other is the angle of the neutral axis. Therefore the unknowns are $\overline{Y}$, α_{ne} and A_{st}. The programs use the Newton-Raphson method to converge to the real roots (the values of unknowns that satisfy the equilibrium). As mentioned earlier the Newton-Raphson method uses the jacobian matrix for convergence. This jacobian matrix can be obtained numerically by secant method as shown before. The procedure of the program can be summaraized as the following.

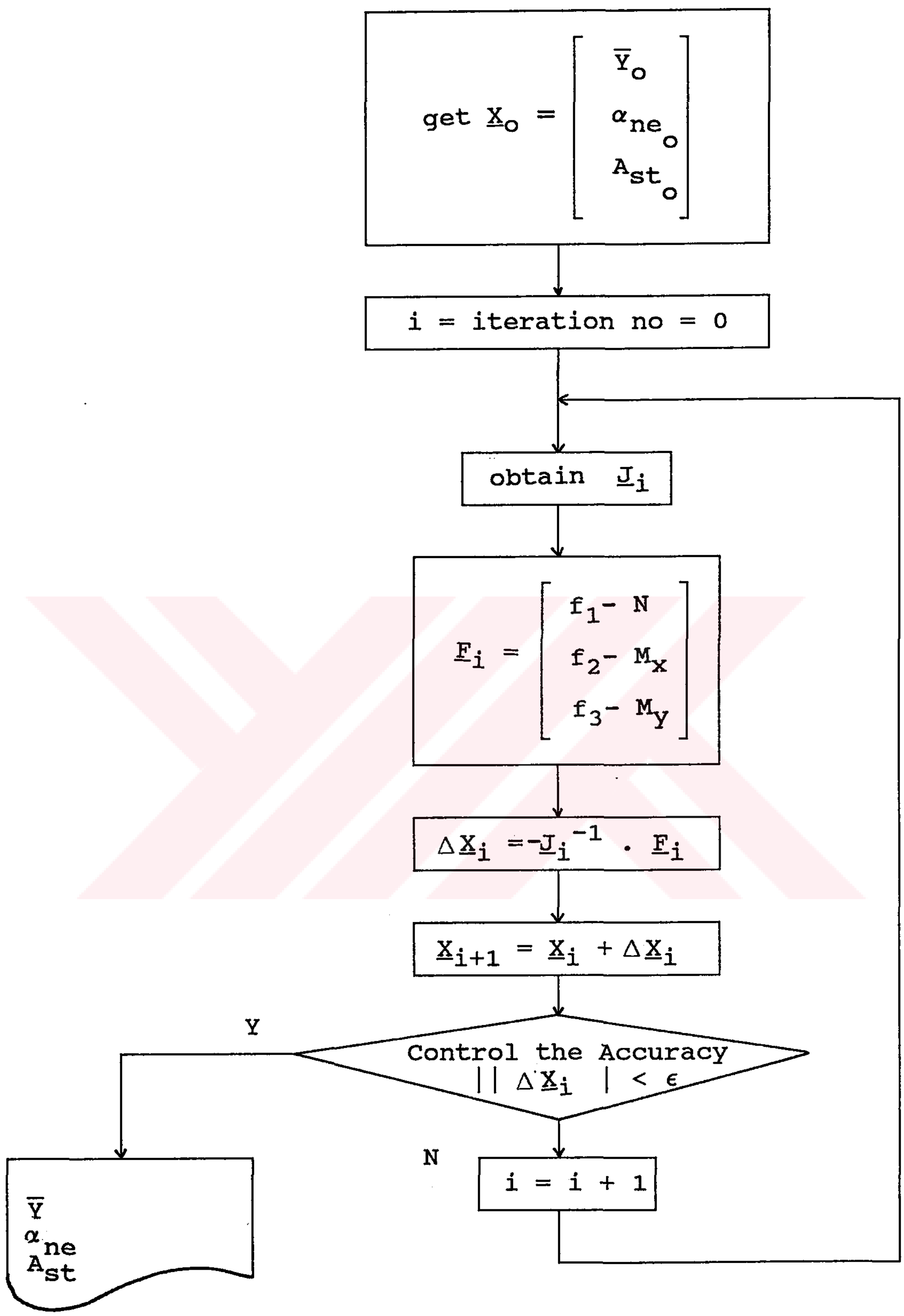


Fig. 5.4. Program Flow-Chart.

1. The program first assumes arbitrary values for the unknowns. For example it gives:

$$Y = Y_o \quad \text{.........................(81)}$$

$$\alpha_{ne} = \alpha_{neo} \quad \text{.........................(82)}$$

$$A_{st} = A_{sto} \quad \text{.........................(83)}$$

2. The program goes to a subroutine that can obtain the axial load and the moments around X- and Y-axis for the assumed steel area and neutral axis position.

$$N_o = N(Y_o, \alpha_{neo}, A_{sto}) \quad \text{..................(84)}$$

$$M_{xo} = M_x(Y_o, \alpha_{neo}, A_{sto}) \quad \text{..................(85)}$$

$$M_{yo} = M_y(Y_o, \alpha_{neo}, A_{sto}) \quad \text{..................(86)}$$

3. After this program adds a certain value (ΔY) to the Y_o and calculate the internal forces N, M_x, M_y again so the first column of Jacobian matrix can be obtained numerically.

$$\frac{\partial N}{\partial Y} = \frac{N(Y_o+\Delta Y, \alpha_{neo}, A_{sto}) - N(Y_o, \alpha_{neo}, A_{sto})}{\Delta Y} \quad \text{....(87)}$$

$$\frac{\partial M_x}{\partial Y} = \frac{M_x(Y_o+\Delta Y, \alpha_{neo}, A_{sto}) - M_x(Y_o, \alpha_{neo}, A_{sto})}{\Delta Y} \quad \text{....(88)}$$

$$\frac{\partial M_y}{\partial Y} = \frac{M_y(Y_o+\Delta Y, \alpha_{neo}, A_{sto}) - M_y(Y_o, \alpha_{neo}, A_{sto})}{\Delta Y} \quad \text{....(89)}$$

4. The program adds a certain value ($\Delta\alpha_{ne}$) to the α_{neo} and calculates the internal forces again so the second column of the jacobian matrix can obtained numericaly.

$$\frac{\partial N}{\partial \alpha_{ne}} = \frac{N(Y_o, \alpha_{neo}+\Delta\alpha_{ne}, A_{sto}) - N(Y_o, \alpha_{neo}, A_{sto})}{\Delta \alpha_{ne}} \quad \text{..(90)}$$

$$\frac{\partial M_x}{\partial \alpha_{ne}} = \frac{M_x(Y_o,\alpha_{neo}+\Delta\alpha_{ne},A_{sto}) - M_x(Y_o,\alpha_{neo},A_{sto})}{\Delta\alpha_{ne}} \quad ..(91)$$

$$\frac{\partial M_y}{\partial \alpha_{ne}} = \frac{M_y(Y_o, neo+\Delta a_{ne},A_{sto}) - M_y(Y_o,\alpha_{neo},A_{sto})}{\Delta\alpha_{ne}} \quad ..(92)$$

5. The program adds (ΔA_{st}) to the A_{sto} value and calculates the internal forces again. Using these forces, third column of the Jacobian matrix is again obtained nummerically.

$$\frac{\partial N}{\partial A_{st}} = \frac{N(Y_o,\alpha_{neo},A_{sto}+\Delta A_{st}) - N(Y_o,\alpha_{neo},A_{sto})}{\Delta A_{st}} \quad ...(93)$$

$$\frac{\partial M_x}{\partial A_{st}} = \frac{M_x(Y_o,\alpha_{neo},A_{sto}+\Delta A_{st}) - M_x(Y_o,\alpha_{neo},A_{sto})}{\Delta A_{st}} \quad ...(94)$$

$$\frac{\partial M_y}{\partial A_{st}} = \frac{M_y(Y_o,\alpha_{neo},A_{sto}+\Delta A_{st}) - M_y(Y_o,\alpha_{neo},A_{sto})}{\Delta A_{st}} \quad ...(95)$$

5. The changes that should be done in the assumed values for the unknowns are found by multiplying the Jacobian matrix by the unbalanced internal forces.

$$\begin{bmatrix} \Delta\bar{Y}_o \\ \Delta\alpha_{neo} \\ \Delta A_{sto} \end{bmatrix} = \begin{bmatrix} \frac{\partial N}{\partial Y} & \frac{\partial N}{\partial \alpha_{ne}} & \frac{\partial N}{\partial A_{st}} \\ \frac{\partial M_x}{\partial Y} & \frac{\partial M_x}{\partial \alpha_{ne}} & \frac{\partial M_x}{\partial A_{st}} \\ \frac{\partial M_y}{\partial Y} & \frac{\partial M_y}{\partial \alpha_{ne}} & \frac{\partial M_y}{\partial A_{st}} \end{bmatrix} \cdot \begin{bmatrix} N_o - N_d \\ M_{xo}-M_{xd} \\ M_{yo}-M_{yd} \end{bmatrix} \quad ....(96)$$

7. The new values for the unknowns are obtained by:

$$Y_1 = Y_o + \Delta Y_o \quad ....(97)$$

$$\alpha_{ne1} = \alpha_{neo} + \Delta\alpha_{ne} \quad ....(98)$$

$$A_{st1} = A_{sto} + \Delta A_{sto} \quad ....(99)$$

8. At this stage the values obtained are closer to the real roots. This iteration procedure is continued until changes in unknowns become very small and negligible.

To obtain the concrete internal forces, two different approaches are used in the computer program. First approach is called "Area Method" and the other is called "Integration of Area Method". Note that the unknown $\overline{Y}$ is different in each approach. In Area Method $\overline{Y}$ is the depth of compression line, whereas in Integration of Area Method it is the depth of neutral axis.

5.4.1. Area Method.

This method is used only for the rectangular stress-block. Since in rectangular stress-block there is no change in the stress value, the concrete internal force will be equal to the area of the compresion zone multiplied by the intensity of the stress-block. Therefore the problem reduces to finding the compression area and the centroid of the compresion zone, Fig. 5.5. The internal forces in concrete are:

$$F_{cc} = A_{cc}\ 0.85\ f_{cd} \qquad \text{.......................(100)}$$

$$Mx_{cc} = F_{cc}\ (\ Y_{cc} - Y_g\ ) \qquad \text{....................(101)}$$

$$My_{cc} = F_{cc}\ (\ X_{cc} - X_g\ ) \qquad \text{....................(102)}$$

And:

$$A_{cc} = \sum_{i=1}^{n} \frac{1}{2} \cdot (x_{i+1} - x_i) \cdot (y_{i+1} + y_i) \qquad \text{..............(103)}$$

$$X_{cc} = - \frac{1}{6\ A_{cc}} \sum_{i=1}^{n} (y_{i+1} - y_i) \cdot (x_i \cdot (x_i - X_{i+1}) + x_{i+1} \cdot x_{i+1}) \qquad \text{........(104)}$$

$$Y_{cc} = \frac{1}{6\,A_{cc}} \sum_{i=1}^{n} (x_{i+1} - x_i) \cdot (y_i \cdot (y_i - y_{i+1}) + y_{i+1} \cdot y_{i+1}) \quad \ldots\ldots\ldots(105)$$

Where:

X_g, Y_g = X and Y coordinates of center of gravity of cross section.

F_{cc} = Resultant force in concrete. (compression area multiplied by stress intensity)

Mx_{cc} = Moment of concrete resultant about x axis.

My_{cc} = Moment of concrete resultant about y axis.

n = Number of nodes which define the compression zone.

A_{cc} = Area of compression zone.

X_{cc},Y_{cc}= X and Y-coordinates of the center of gravity of compression zone.

5.4.2. Integration of Area Method.

In this method any stress-block or any stress-strain model can be used. The internal forces in concrete are calculated by dividing the cross-section into small strips parallel to the neutral axis. The depth of each strip is "d_{int}" and the stress along this depth is assumed to be constant. The stress is obtained from stress-strain relationship using the strain level at the middepth of the strip, Fig. 5.6. Concrete internal forces in the compression zone are:

$$N_c = \sum_{i=1}^{n} dF_i \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(106)$$

$$Mx_c = \sum_{i=1}^{n} dF_i \cdot (Y_i - Y_g) \quad \ldots\ldots\ldots\ldots(107)$$

$$My_c = \sum_{i=1}^{n} dF_i \cdot (X_i - X_g) \quad \ldots\ldots\ldots\ldots(108)$$

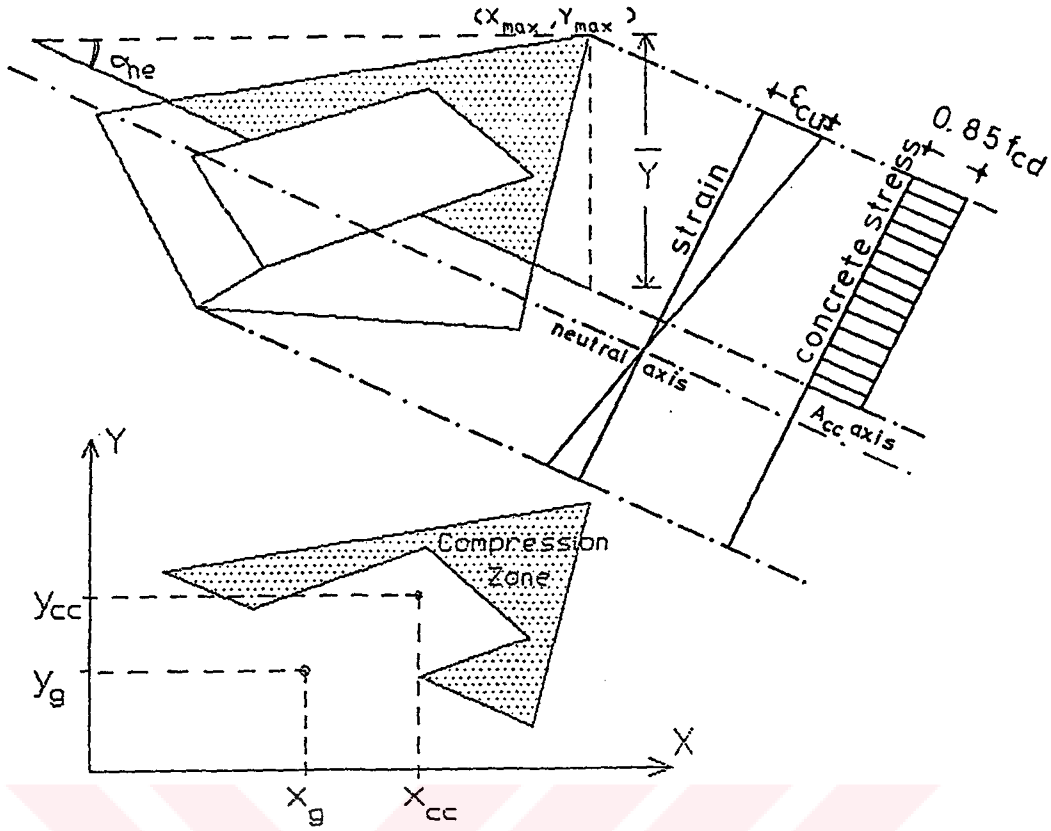


Fig. 5.5. Area Method.

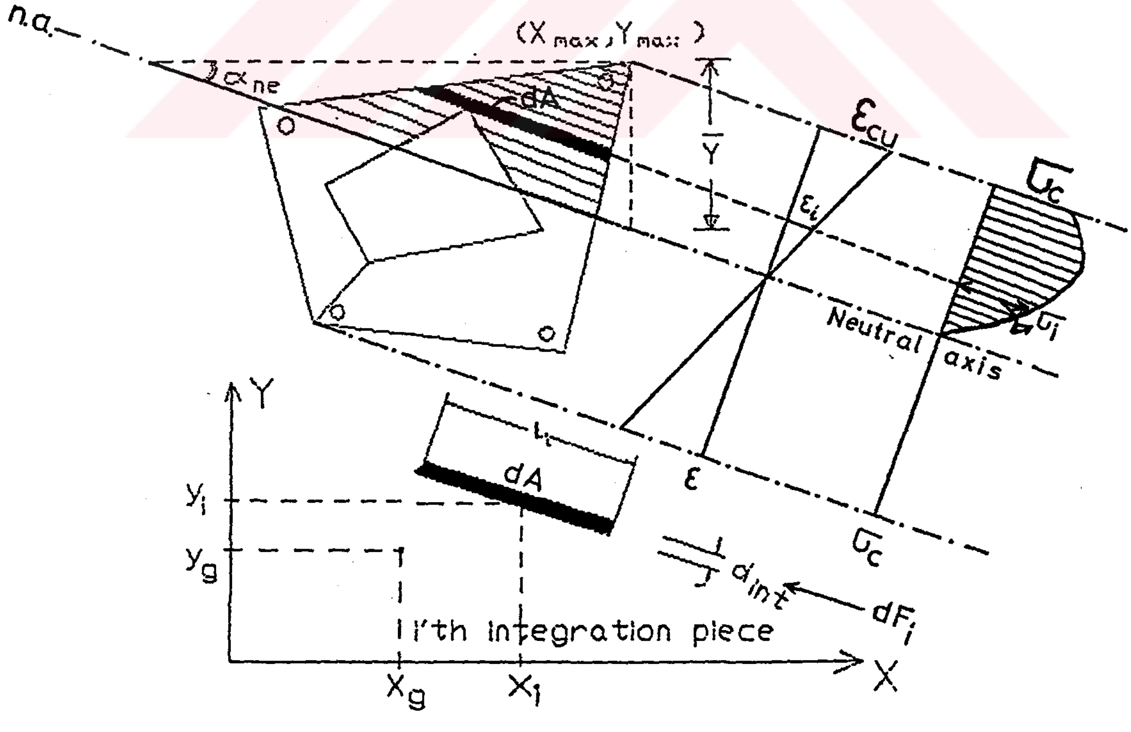


Fig. 5.6. Integration of Area Method.

Where:

N_c , Mx_c , My_c = Internal forces created by concrete.

n = Number of integration strips.

X_i , Y_i = x and y-Coordinates of center of gravity for i'th integration strip.

dA_i = $d_{int} \cdot L_i$

dF_i = $dA_i \cdot \sigma_i$

d_{int} = Depth of integration strip.

L_i = Length of i'th integration strip, parallel to the n.a.

dA_i = Area of i'th integration strip.($d_{int} \cdot L_i$)

dF_i = Normal force in i'th integration strip.

σ_i = Mean stress value in i'th integration strip.

5.5 Advantages of the program developed.

1- The program is very efficient for the design of reinforced concrete columns subjected to axial load and biaxial bending. It can locate the real position of neutral axis and determine required steel area for the given axial load and biaxial moments .

2- It can handle any arbitrary reinforced concrete cross-section, having steel arranged in any pattern.

3- Cross-sections are defined by a polygonal system. Therefore, there is no need to open large dimensions for simulation of the arbitrary cross-sections.

4- Program is "user freindly". Data can be handled and changed interactively.

5- Program also suggests various choices for reinforcing bar dimensions upon user's request.

6- For each case (run) program can check the code specifications,(min. & max. reinforcement, min. eccentricity , etc.) and gives warning massages if necessary. ACI or TS-500 can be used without any special input.

7- Section and stress-strain relationship to be used are displayed on screen for visual inspection purposes.

Using "Area Method" with Rectangular Stress-Block :

8- Program is very fast.

9- Actual area of compression zone can be obtained for the specified polygonal system.

Using " Integration of Area Method " :

10- Program is flexible, and any concrete model (stress-strain relationship) can be employed.

11- Using strip discritization instead of the small rectangular pieces, program is fast and with less errors in simulation of compression zone.

CHAPTER 6

CASE STUDIES

6.1. Introduction.

In this chapter several case studies will be made using the computer program discussed in Chapter 5. In these case studies rectangular cross-sections having different reinforcement configurations and L-shaped cross sections will be considered. In all these examples, like in a real design problem it is assumed that the geometry and dimensions of the cross-section, configuration of reinforcement, material properties (f_{cd} and f_{yd}) and external design load effects (N_d , M_{xd} , M_{yd}) are known. The objective is to find the steel area required. To find the steel area, three different types of stress distribution have been assumed; (a) rectangular stress block, (b) CEB stress block and (c) Hognestad stress-block. Limiting concrete strain values corresponding to each block have been taken accordingly as specified for each block (Chapter-4).

In the first part of this Chapter steel areas obtained using different stress-blocks are compared and discussed. In the second part, case studies are made using authors computer program and other methods suggested by different investigators. And finaly results obtained by the computer program developed are compared with the test results.

6.2. Effect of Different Stress-Strain Relationships.

Different cross-sections and steel configurations (Fig. 6.1.) have been investigated under different levels of axial load and biaxial bending. In each case the aim was to find the required steel area. Tables 5.1-5.5 summarize the outputs of the computer program for these case studis. In these tables the type of solution and stress block used are designated as,

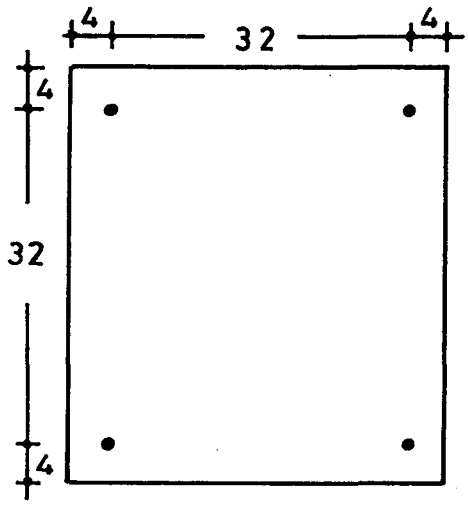
R-A = rectangular stress-block using Area Method.

R-I = rectangular stress-block using integration of area method.

CEB = CEB stress-block using integration of area method.

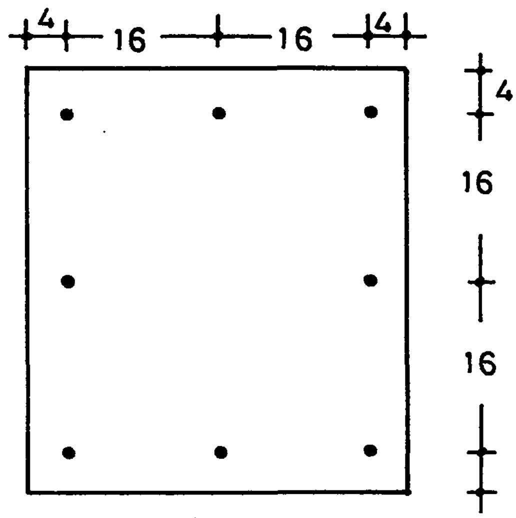
HOG = Hognestad stress-strain relationship using integration of area method.

The material strengths used are, f_{cd}=170 Kgf/cm2 and f_{yd}=3650 Kgf/cm² (representing C25 and S420). The axial load levels used (N_d / N_l) varied from 10% to 70% . Where N_l indicates the maximum axial load permited by TS500 Code (N_l= 0.6 * f_{ck} * A_c). For each level of axial load, three different ratio's of M_{yd} / M_{xd} were used, 1.0 , 1/2 and 1/4 . Dimensionless eccentricities (e_x/x ,e_y/y) varied from .075 to 2.144 .



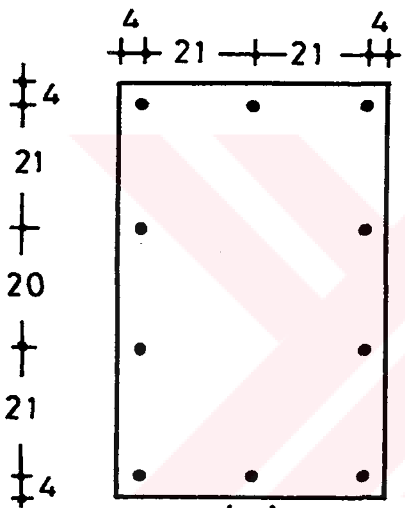
(a)

Cross-Section of Example 1.



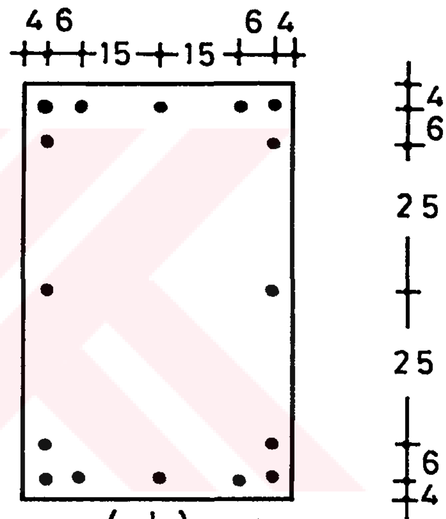
(b)

Cross-Section of Example 2.



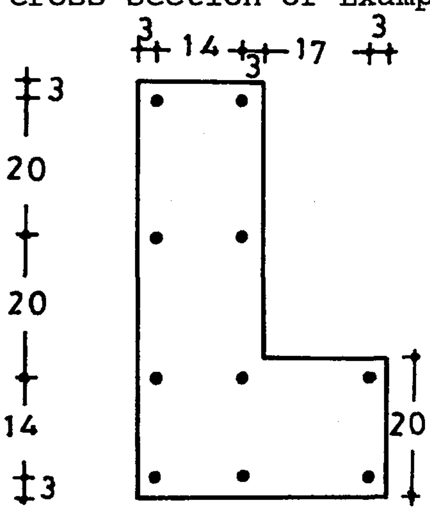
(c)

Cross-Section of Example 3.



(d)

Cross-Section of Example 4.



(e)

Cross-Section of Example 5.

U N I T S : c m

Note: It is assumed that bar diameters are identical in a given cross-section.

Fig. 6.1. Cross-Sections Used for the Case Studies.

Table 6.1. Example 1, Steel Area Using Different Stress Blocks.

Nd [kgf]	Nd / Nl	Mxd [kgf-cm]	Myd [kgf-cm]	Myd / Mxd	ey / y	ex / x	type of solv.	$\overline{Y}$ [cm]	α [deg.]	Ast [cm2]
24000	10%	1500000	1500000	1 / 1	1.563	1.563	R-A	26.543	45.000	31.928
							R-I	30.989	45.000	31.739
							CEB	32.339	45.000	32.491
							HOG	31.767	45.000	31.685
24000	10%	1897363	948681	1 / 2	1.976	0.988	R-A	20.892	31.424	29.609
							R-I	24.498	31.462	29.517
							CEB	26.080	32.769	30.136
							HOG	25.738	33.329	29.470
24000	10%	2057980	514495	1 / 4	2.144	0.536	R-A	12.548	13.648	28.935
							R-I	14.553	13.445	28.928
							CEB	15.835	14.936	29.165
							HOG	14.930	14.499	29.200
72000	30%	1500000	1500000	1 / 1	0.521	0.521	R-A	32.430	45.000	27.455
							R-I	38.108	45.000	27.429
							CEB	39.082	45.000	28.688
							HOG	37.816	45.000	28.351
72000	30%	1897363	948681	1 / 2	0.659	0.329	R-A	25.431	31.041	25.545
							R-I	30.017	30.988	25.581
							CEB	31.252	31.851	26.582
							HOG	30.819	33.075	26.252
72000	30%	2057980	514495	1 / 4	0.715	0.179	R-A	21.214	20.700	22.201
							R-I	25.236	20.503	22.344
							CEB	26.423	22.003	22.860
							HOG	26.368	23.921	22.608

Table 6.1. (Continued)

Nd [kgf]	Nd/Nl	Mxd [kgf-cm]	Myd [kgf-cm]	Myd/Mxd	ey/y	ex/x	type of solv.	$\bar{Y}$ [cm]	α [deg.]	Ast [cm2]
120000	50%	1500000	1500000	1/1	0.312	0.312	R-A	38.853	45.000	25.743
							R-I	45.850	45.000	25.858
							CEB	46.946	45.000	26.762
							HOG	44.780	45.000	26.959
120000	50%	1897363	948681	1/2	0.395	0.198	R-A	31.093	31.096	24.031
							R-I	36.801	31.102	24.247
							CEB	37.675	31.013	24.770
							HOG	36.758	32.520	25.032
120000	50%	2057980	514495	1/4	0.429	0.107	R-A	26.044	18.844	21.112
							R-I	30.403	18.806	20.896
							CEB	32.013	19.633	21.373
							HOG	31.627	21.816	21.805
168000	70%	1500000	1500000	1/1	0.223	0.223	R-A	43.884	45.000	31.148
							R-I	52.082	45.000	31.396
							CEB	53.044	45.000	32.579
							HOG	51.303	45.000	30.894
168000	70%	1897363	948681	1/2	0.282	0.141	R-A	35.651	31.038	29.743
							R-I	41.633	30.975	29.560
							CEB	42.916	31.025	30.632
							HOG	41.921	31.612	29.396
168000	70%	2057980	514495	1/4	0.306	0.077	R-A	30.704	19.301	26.936
							R-I	36.038	19.271	26.871
							CEB	37.273	19.748	27.573
							HOG	36.369	19.863	27.403

Table 6.2. Example 2, Steel Area Using Different Stress Blocks.

Nd [kgf]	Nd / Nl	Mxd [kgf-cm]	Myd [kgf-cm]	Myd / Mxd	ey / y	ex / x	type of solv.	$\bar{Y}$ [cm]	α [deg.]	Ast [cm2]
24000	10%	1500000	1500000	1/1	1.563	1.563	R-A	27.644	45.000	36.115
							R-I	32.159	45.000	35.986
							CEB	33.042	45.000	36.356
							HOG	32.073	45.000	35.722
24000	10%	1897363	948681	1/2	1.976	0.988	R-A	20.624	28.431	34.067
							R-I	24.171	28.530	34.004
							CEB	25.236	29.603	34.192
							HOG	24.226	29.603	34.097
24000	10%	2057980	514495	1/4	2.144	0.536	R-A	15.010	14.822	32.750
							R-I	17.909	14.724	32.764
							CEB	18.237	15.552	32.894
							HOG	17.968	16.620	32.850
72000	30%	1500000	1500000	1/1	0.521	0.521	R-A	32.757	45.000	32.404
							R-I	38.470	45.000	32.331
							CEB	39.276	45.000	32.108
							HOG	38.229	45.000	30.116
72000	30%	1897363	948681	1/2	0.659	0.329	R-A	25.429	30.187	29.966
							R-I	29.976	30.148	30.030
							CEB	30.619	30.285	29.965
							HOG	29.869	30.914	28.911
72000	30%	2057980	514495	1/4	0.715	0.179	R-A	20.722	18.629	26.898
							R-I	24.681	18.529	27.097
							CEB	25.264	19.105	27.025
							HOG	24.793	19.967	26.519

Table 6.2. (Continued)

Nd [kgf]	Nd / N1	Mxd [kgf-cm]	Myd [kgf-cm]	Myd / Mxd	ey / y	ex / x	type of solv.	$\overline{Y}$ [cm]	α [deg.]	Ast [cm2]
120000	50%	1500000	1500000	1/1	0.312	0.312	R-A	38.071	45.000	31.710
							R-I	44.990	45.000	31.902
							CEB	45.425	45.000	32.087
							HOG	43.830	45.000	30.631
120000	50%	1897363	948681	1/2	0.395	0.198	R-A	30.207	30.637	29.663
							R-I	35.799	30.620	29.897
							CEB	36.077	30.211	30.203
							HOG	35.423	31.283	29.220
120000	50%	2057980	514495	1/4	0.429	0.107	R-A	25.182	17.942	26.407
							R-I	29.496	17.989	26.295
							CEB	30.564	18.309	26.926
							HOG	30.232	19.644	26.545
168000	70%	1500000	1500000	1/1	0.223	0.223	R-A	42.593	45.000	36.557
							R-I	49.858	45.000	36.387
							CEB	51.151	45.000	36.491
							HOG	49.678	45.000	34.769
168000	70%	1897363	948681	1/2	0.141	0.282	R-A	34.350	30.687	34.821
							R-I	40.265	30.664	34.724
							CEB	40.732	29.779	35.484
							HOG	39.818	30.176	33.935
168000	70%	2057980	514495	1/4	0.306	0.077	R-A	29.230	18.071	32.546
							R-I	34.448	18.083	32.580
							CEB	34.972	17.646	33.034
							HOG	34.154	17.642	32.705

Table 6.3. Example 3, Steel Area Using Different Stress Blocks.

Nd [kgf]	Nd / Nl	Mxd [kgf-cm]	Myd [kgf-cm]	Myd / Mxd	ey / y	ex / x	type of solv.	$\bar{Y}$ [cm]	α [deg.]	Ast [cm2]
52500	10%	4000000	4000000	1/1	1.088	1.524	R-A	51.449	60.461	57.346
							R-I	59.912	60.449	57.196
							CEB	60.826	59.914	57.519
							HOG	58.524	59.703	56.374
52500	10%	5059640	2529820	1/2	1.377	0.964	R-A	37.028	42.859	49.043
							R-I	43.715	42.828	49.159
							CEB	44.406	42.793	49.081
							HOG	42.592	42.826	48.769
52500	10%	5487952	1371988	1/4	1.493	0.523	R-A	26.557	25.451	43.948
							R-I	31.086	25.476	43.857
							CEB	32.256	25.730	43.949
							HOG	31.501	26.607	43.225
157500	30%	4000000	4000000	1/1	0.363	0.508	R-A	63.661	60.307	44.813
							R-I	75.543	60.329	45.058
							CEB	76.187	60.013	45.693
							HOG	73.093	59.762	43.678
157500	30%	5059640	2529820	1/2	0.459	0.321	R-A	48.104	44.951	34.978
							R-I	56.584	44.949	34.959
							CEB	57.853	44.978	36.142
							HOG	55.666	45.405	35.238
157500	30%	5487952	1371988	1/4	0.498	0.174	R-A	37.814	30.022	27.316
							R-I	44.666	30.000	27.398
							CEB	45.931	29.748	28.497
							HOG	44.551	31.015	28.309

Table 6.3. (Continued)

Nd [kgf]	Nd / Nl	Mxd [kgf-cm]	Myd [kgf-cm]	Myd / Mxd	ey / y	ex / x	type of solv.	$\overline{Y}$ [cm]	α [deg.]	Ast [cm2]
262500	50%	4000000	4000000	1/1	0.218	0.305	R-A	74.855	59.427	42.925
							R-I	88.489	59.437	43.046
							CEB	91.341	59.958	45.415
							HOG	86.616	59.391	43.295
262500	50%	5059640	2529820	1/2	0.275	0.193	R-A	60.373	47.014	32.311
							R-I	70.738	47.026	32.219
							CEB	71.837	45.884	35.660
							HOG	69.531	46.619	34.396
262500	50%	5487952	1371988	1/4	0.299	0.105	R-A	49.685	31.645	23.754
							R-I	58.038	31.676	23.641
							CEB	59.644	30.693	26.297
							HOG	58.078	32.148	26.255
367500	70%	4000000	4000000	1/1	0.155	0.218	R-A	87.017	59.979	53.155
							R-I	102.731	59.949	53.334
							CEB	105.526	60.202	56.040
							HOG	100.694	59.814	53.656
367500	70%	5059640	2529820	1/2	0.197	0.138	R-A	69.056	45.950	42.626
							R-I	80.688	45.900	42.408
							CEB	82.326	45.056	47.192
							HOG	79.244	45.308	45.753
367500	70%	5487952	1371988	1/4	0.213	0.075	R-A	58.513	30.223	35.019
							R-I	68.918	30.241	35.047
							CEB	69.798	29.291	38.832
							HOG	67.226	29.570	39.829

Table 6.4. Example 4, Steel Area Using Different Stress Blocks.

Nd [kgf]	Nd / Nl	Mxd [kgf-cm]	Myd [kgf-cm]	Myd / Mxd	ey / y	ex / x	type of solv.	$\bar{Y}$ [cm]	α [deg.]	Ast [cm2]
52500	10%	4000000	4000000	1/1	1.088	1.524	R-A	55.278	63.869	56.020
							R-I	64.549	63.817	55.826
							CEB	66.018	63.294	56.625
							HOG	63.832	63.141	55.400
52500	10%	5059640	2529820	1/2	1.377	0.964	R-A	39.561	48.128	46.401
							R-I	46.900	48.146	46.548
							CEB	48.493	48.802	47.022
							HOG	46.880	48.723	45.999
52500	10%	5487952	1371988	1/4	1.493	0.523	R-A	25.557	26.860	40.806
							R-I	30.041	26.851	40.799
							CEB	31.610	28.029	41.273
							HOG	30.272	28.026	41.104
157500	30%	4000000	4000000	1/1	0.363	0.508	R-A	67.239	62.869	44.455
							R-I	78.370	62.813	44.181
							CEB	80.713	62.619	45.853
							HOG	77.034	62.264	44.551
157500	30%	5059640	2529820	1/2	0.459	0.321	R-A	50.278	47.907	33.532
							R-I	59.205	47.901	33.551
							CEB	61.401	48.533	35.130
							HOG	59.195	48.821	34.072
157500	30%	5487952	1371988	1/4	0.498	0.174	R-A	39.306	33.138	24.669
							R-I	46.405	33.112	24.711
							CEB	48.337	33.540	25.970
							HOG	47.091	34.926	25.478

Table 6.4. (Continued)

Nd [kgf]	Nd / N1	Mxd [kgf-cm]	Myd [kgf-cm]	Myd / Mxd	ey / y	ex / x	type of solv.	$\overline{Y}$ [cm]	α [deg.]	Ast [cm2]
262500	50%	4000000	4000000	1/1	0.218	0.305	R-A	78.760	61.485	41.791
							R-I	93.226	61.493	42.044
							CEB	97.113	62.279	44.125
							HOG	91.646	61.684	43.002
262500	50%	5059640	2529820	1/2	0.275	0.193	R-A	62.766	49.060	30.157
							R-I	73.545	49.070	30.026
							CEB	75.268	48.385	33.085
							HOG	72.602	49.102	32.191
262500	50%	5487952	1371988	1/4	0.299	0.105	R-A	51.413	33.887	20.918
							R-I	60.943	33.902	21.127
							CEB	62.244	33.358	23.335
							HOG	60.582	35.036	23.257
367500	70%	4000000	4000000	1/1	0.155	0.218	R-A	91.905	62.050	52.514
							R-I	108.595	62.012	52.706
							CEB	112.885	62.778	55.928
							HOG	108.029	62.519	53.796
367500	70%	5059640	2529820	1/2	0.197	0.138	R-A	71.784	48.152	40.476
							R-I	83.907	48.116	40.282
							CEB	86.766	48.042	44.642
							HOG	83.609	48.462	43.187
367500	70%	5487952	1371988	1/4	0.213	0.075	R-A	60.870	32.854	31.957
							R-I	71.584	32.843	31.941
							CEB	72.851	32.062	35.400
							HOG	70.310	32.708	35.891

Table 6.5. Example 5, Steel Area Using Different Stress Blocks.

Nd [kgf]	Nd / Nl	Mxd [Kgf-cm]	Myd [Kgf-cm]	Myd / Mxd	ey / y	ex / x	type of solv.	$\overline{Y}$ [cm]	α [deg.]	Ast [cm2]
24000	10 %	1500000	1500000	1 / 1	1.042	1.563	R-A	65.328	61.390	42.421
							R-I	69.872	61.382	42.244
							CEB	71.073	61.377	42.647
							HOG	69.806	61.232	42.030
24000	10 %	1897363	948681	1 / 2	1.318	0.988	R-A	58.214	56.505	36.959
							R-I	63.433	56.510	37.099
							CEB	63.151	56.076	36.742
							HOG	61.903	55.832	35.498
24000	10 %	2057980	514495	1 / 4	1.429	0.536	R-A	52.932	52.152	30.586
							R-I	57.777	52.144	30.606
							CEB	57.910	51.710	30.611
							HOG	57.008	51.771	29.546
72000	30 %	1500000	1500000	1 / 1	0.347	0.521	R-A	72.838	62.480	37.513
							R-I	79.118	62.467	37.636
							CEB	79.906	62.522	37.346
							HOG	77.852	62.164	36.634
72000	30 %	1897363	948681	1 / 2	0.439	0.329	R-A	63.216	56.325	30.419
							R-I	68.818	56.336	30.284
							CEB	69.724	56.179	30.641
							HOG	67.775	55.922	29.933
72000	30 %	2057980	514495	1 / 4	0.476	0.179	R-A	57.554	51.072	23.569
							R-I	63.715	51.051	23.733
							CEB	63.621	50.585	23.967
							HOG	61.882	50.566	23.368

Table 6.5. (Continued)

Nd [Kgf]	Nd / Nl	Mxd [kgf-cm]	Myd [kgf-cm]	Myd / Mxd	ey / y	ex / x	type of solv.	$\bar{Y}$ [cm]	α [deg.]	Ast [cm2]
120000	50%	1500000	1500000	1/1	0.208	0.313	R-A	80.763	63.306	35.673
							R-I	87.719	63.322	35.489
							CEB	88.494	63.094	35.810
							HOG	87.803	63.504	35.269
120000	50%	1897363	948681	1/2	0.264	0.198	R-A	67.975	55.455	28.642
							R-I	74.956	55.456	28.659
							CEB	76.232	55.603	29.249
							HOG	74.185	55.371	27.555
120000	50%	2057980	514495	1/4	0.286	0.107	R-A	61.083	48.331	22.220
							R-I	68.454	48.392	22.366
							CEB	68.765	48.243	22.848
							HOG	66.217	47.392	21.748
168000	70%	1500000	1500000	1/1	0.149	0.223	R-A	89.724	64.414	38.413
							R-I	98.715	64.397	38.533
							CEB	100.953	64.766	39.372
							HOG	101.168	65.499	38.881
168000	70%	1897363	948681	1/2	0.188	0.141	R-A	72.647	54.479	31.591
							R-I	81.024	54.536	31.805
							CEB	82.733	55.048	31.777
							HOG	79.797	54.486	30.465
168000	70%	2057980	514495	1/4	0.204	0.077	R-A	63.711	44.916	26.423
							R-I	70.900	44.806	26.254
							CEB	72.582	44.780	27.597
							HOG	68.871	42.578	26.729

Examination of results given in Tables 6.1 to 6.5 reveal that if rectangular stress block is used, almost the same steel area is obtained from area method and integration of area method. The average of the differences in steel areas obtained using these two different methods is only 0.03% , maximum difference being 0.8% .

The differences between the steel areas obtained using different stress blocks were not very significant. Under low axial loads and M_{yd}/M_{xd} = 1.0 steel areas obtained using different blocks were almost same (CEB block required 1% more steel). As M_{yd}/M_{xd} ratio decreased, results obtained using Hognestad block approached the CEB method. However in no case the difference between the rectangular block and CEB exceeded 4% .

Under high axial loads in general CEB resulted in higher steel areas than the rectangular one. In some cases this difference approached 12% . Steel areas obtained using Hognestad block under high axial loads were in general lower but closer to CEB results. However this was not consistent.

In general steel areas obtained using the rectangular block and CEB block were approximately the same, the difference in most cases did not exceed a few percentages. However in some cases this difference become as higher as 12% . Even 12% is not a significant difference, since steel areas are being compared. Such higher differences are obtained under high axial loads, it should be realized that the contribution of steel to ultimate capacity is rather small. Therefore if capacities were compared the difference would reduce to a few percent.

It can then be concluded that there is no significant difference when different stress-blocks are used. So the area method can be good enough in the design of reinforced concrete sections. One of the other adventages of R-A method is its time reducing effect. The solution of problems by Integration of area method takes quite a bit of time. This is related to the integration of the stress-strain model. As the thickness of the integration strips decreases, calculation time increases. To give an idea, time consumed to solve a problem using an IBM XT compatible 10 MHZ Computer are given below:

12 sec. By R-A method .... A_{st} = 48.13397 cm2

144 sec. By R-I method. d_{int}= 1.0 cm. A_{st} = 47.82655 cm2

208 sec. By R-I method. d_{int}= 0.5 cm. A_{st} = 48.16974 cm2

730 sec. By R-I method. d_{int}= 0.2 cm. A_{st} = 48.09923 cm2

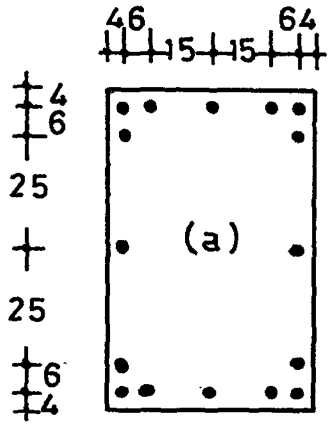
Where:

d_{int} = depth of integration strips

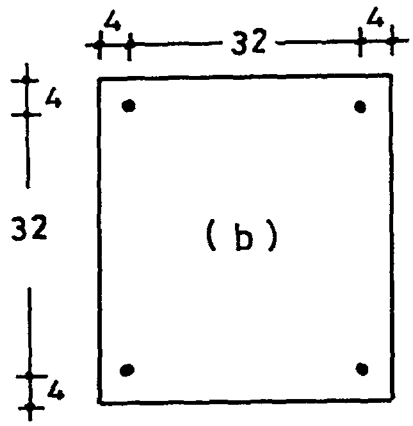
Using a 10 MHZ Math. Co-Processor, the computing time reduced to 0.7 , 16 ,26 and 85 second respectively.

6.3. Comparison with other Analytical Methods.

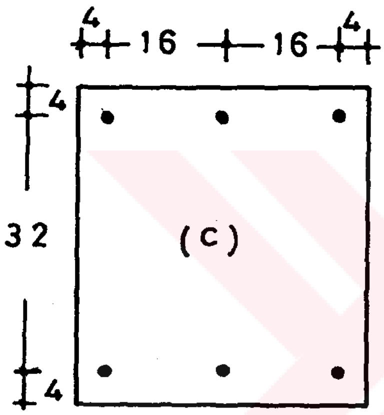
To test the program developed several examples were solved using Author's program and methods developed by others. Examples 6-13 (Fig.6.2.) are the problems that were solved by Kıral and Dündar [30] in 1987. In Table 6.6 total steel areas obtained by Kıral and Dündar (K&D) and results obtained using the computer progam developed (R-A , R-I , CEB , HOG , K&P) are summarized. The examples solved by Kıral and Dündar were taken from various sources. The results of the available references are also given in the table. In using Author's computer program an additional stress-strain model, Kent & Park (ϵ_{cu} = 0.003) was also used. It is seen that the difference between Kıral-Dündar's solutions and R-A method is about 0.4% ,maximum being 2% . It should be pointed out that while the proposed method can tackle any section with any stress-strain relationship the method of Kıral and Dündar is limited to rectangular section and rectangular stress block. The approximate method proposed by Çakıroğlu and Özer seem to give reasonable results.



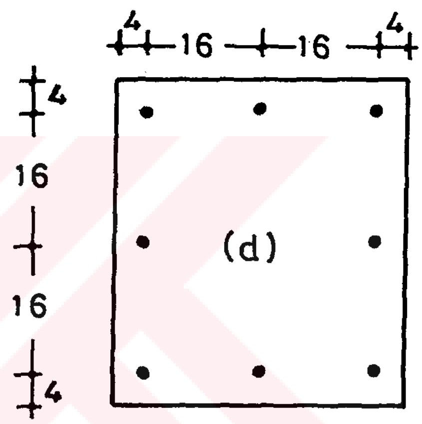
Cross-section of ex. 6.
(Çakıroğlu & Özer. [14] P.25)



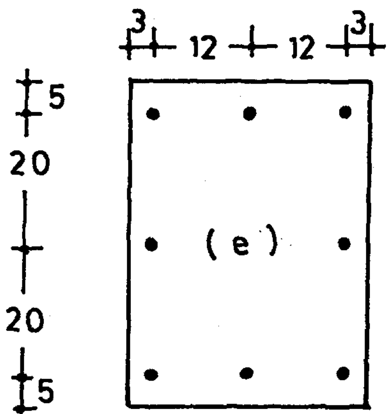
Cross-section of ex. 7 & 8.
(Çakıroğlu & Özer. [14] P.26)
(Durmuş & Eyuboğlu [17] p.17)



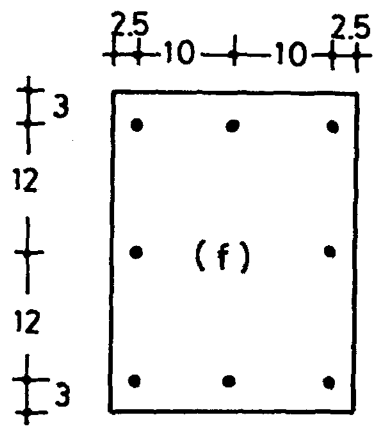
Cross section of ex. 9.
(Durmuş & Eyuboğlu.[17] P.17)



Cross section of ex.10.
(Durmuş & Eyuboğlu.[17] P.18)



Cross-Section of Ex.11 & 13.
(Durmuş & Eyuboğlu.[17] P.18)
(Bakır [5])



Cross-Section of Ex.12.
(Bakır [5])

Note: In Each Cross-section the steel bars have the same diameter.
U N I T S : c m

Fig. 6.2. Previous Analytical Examples Cross-Sections.

TABLE 6.6. Comparison of Author's Results with Others.

EX.	fcd	fyd	Nd	Mxd	Myd	Prev. analysis		Ast cm^2	Ast (program Output) cm^2			
no:	Kgf/cm^2	Kgf/cm^2	Kgf	Kgf-cm	Kgf-cm	Refer.	Ast cm^2	K & D	R-A	CEB	HOG	K&P
6	133.33	3478.26	400000	4800000	3000000	Ç & Ö	83.66	77.35	77.521	80.250	78.053	73.061
7	166.66	3478.26	80000	1200000	1200000	Ç & Ö	19.90	17.98	17.778	19.131	18.975	17.370
8	200.00	3650.00	200000	2100000	1600000	D & E	38.57	40.98	41.141	41.657	40.053	39.502
9	200.00	3650.00	200000	2100000	1600000	D & E	44.975	44.82	44.991	44.718	43.064	43.439
10	200.00	3650.00	200000	2100000	1600000	D & E	45.15	47.96	48.133	47.168	45.137	46.429
11	110.00	3650.00	19500	117000	390000	-----	-----	8.49	8.3150	8.3319	8.1720	8.0192
12	133.33	3650.00	70000	1750000	840000	Bakır	-----	29.48	29.677	30.160	29.010	29.061
13	110.00	1910.00	19500	117000	390000	Bakır	-----	14.70	14.730	15.063	15.185	14.376

K & D = Kıral and Dündar : Rectangular Stress block and $\epsilon cu=0.003$ were used.(Computer Program)

Ç & Ö = Çakıroğlu and Özer : CEB stress block and $\epsilon cu=0.003$ were used.(approx. method,Formulation)

D & E = Durmuş and Eyuboğlu: CEB stress block and $\epsilon cu=0.003$ were used.(approx. method,Formulation)

Bakır = Bakır and Bakır : Rectangular stress block and $\epsilon cu=0.003$ were used.(Tables)

6.4. Comparison with Experimental Results.

To check the reliability and accuracy of the computer program developed, some comparisions were made with experimental values.

Furlong [26] in 1979 tested nine specimens under three different levels of axial load by applying biaxial bending. In all nine specimens cross-sectional properties and reinforcement were identical, total area of steel being 7.13 cm2, Fig. 6.3. Using the ultimate values of axial loads and moments reported by Furlong and the material properties specified, steel areas required were computed for each specimen using the computer program developed. The results are summarized in Table 6.7. It can be easily seen that steel areas found using the computer program are usually higher than the existing steel areas (except specimens R4,R7,R8). If the first three specimens with high axial loads are disregarded (R1,R2 and R3) difference between the existing steel area and computed ones is not so significant.

It should be noted that the specimens where the difference is significant (R1,R2 and R3) were tested under very high axial loads. It is possible that in these specimens confinement played a very important role and maximum strains were much higher than the ones used in the computer program. Also under high axial loads the contribution of concrete become very significant and effect the steel area required.

TABLE 6.7. Evaluation of Experimental Results Obtained by Furlong. [26]
(A_{st} used = 7.13 cm2)

EX. no:	spec. No:	fck Kgf /cm²	Nu Kgf	Mux Kgf-cm	Muy Kgf-cm	Ast (program output) cm² R-A	C E B	HOG	K & P
14	R1	337	53600	69100	210000	11.686	11.731	12.056	10.503
15	R2	336	54000	141400	175600	9.870	10.276	10.363	8.885
16	R3	357	58000	261000	145200	11.396	11.802	11.895	10.576
17	R4	345	39200	45900	183300	5.893	6.133	6.524	5.272
18	R5	359	42400	146100	171500	7.222	7.505	7.626	6.623
19	R6	324	38600	48000	116300	8.128	8.593	8.350	7.646
20	R7	305	24300	59500	166300	6.897	7.121	7.163	6.621
21	R8	300	18200	98700	138300	6.287	6.570	6.625	6.200
22	R9	307	18200	217000	98500	7.365	7.498	7.342	7.174

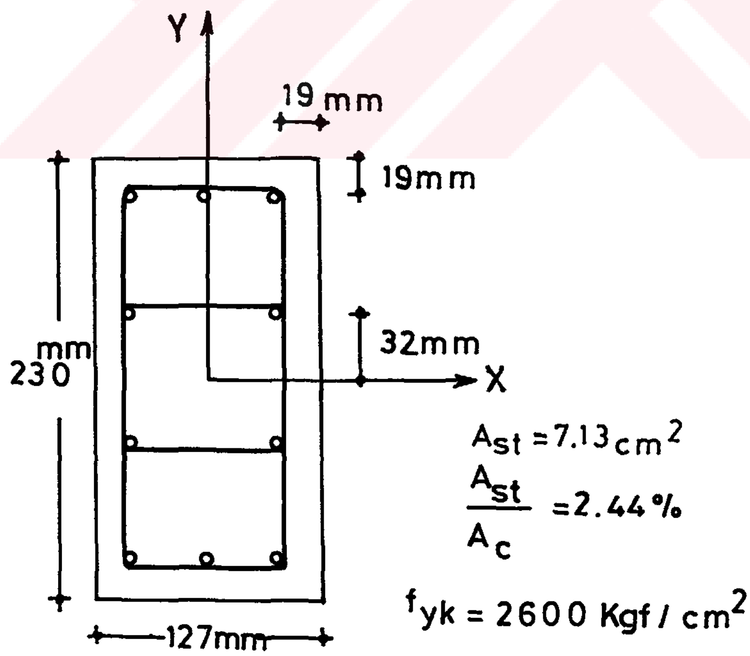


Fig. 6.3 Cross-Section of the Test Specimen Investigated by Furlong [26].

Özcebe [34] tested five specimens with square cross-sections under axial load and cyclic diagonal bending. Specimens had the same cross-section with, 39.27 cm^2 of longitudinal steel (see Fig. 6.4). The only difference between the specimens was that the confinement spacings changed. Úzcebe computed the expected ultimate diagonal moment capacity (Using ACI 318-83) under the given axial load and compared them with the observed values. The steel areas calculated using the computer program developed in this thesis was found to be very close to the area selected by Özcebe (Table 6.8). The small difference might be related to the selected K_1 value or related to the rounding in ultimate moment capacity. Using the ultimate moment capacity obtained during the tests the steel areas were recalculated by the computer program developed (Table 6.8). It was observed that steel areas found using the computer program were usually higher than the existing steel areas (except D5). Higher moments observed during the tests might be due to confinement and to the strain hardening of steel.

Hsu [21] in 1985 tests L-shaped cross sections under axial load and biaxial bending. The cross-sections of all specimens tested were identical and had the longitudinal steel area was 9.97 cm^2 (see Fig.6.5). The results obtained are presented in Table 6.9. Examination of these results reveal that (except specimen 1-a), the steel areas obtained by the computer program are very close to the

TABLE 6.8. Evaluation of Experimental Results Obtained by Özcebe. [34] (A_{st} used = 39.3 cm2)

EX. no:	spec No:	s cm	fck Kgf /cm²	fyk Kgf /cm²	fywk Kgf /cm²	Nu Kgf	Obsorved Mu Kgf-cm	Computed Mu Kgf-cm	Ast (R-A Method) cm², using the Mu Observed	Computed
23	D1	15.0	403	4530	4700	0	2850000	2310000	53.292	40.701
24	D2	15.0	302	4530	4700	60000	2700000	2470000	50.331	43.328
25	D3	7.5	348	4300	4700	60000	2750000	2420000	50.126	40.070
26	D4	5.0	436	4300	4700	60000	3000000	2720000	52.622	44.262
27	D5	15.0	493	4300	4700	50000	2540000	2740000	38.664	43.385

Note : Mx = My

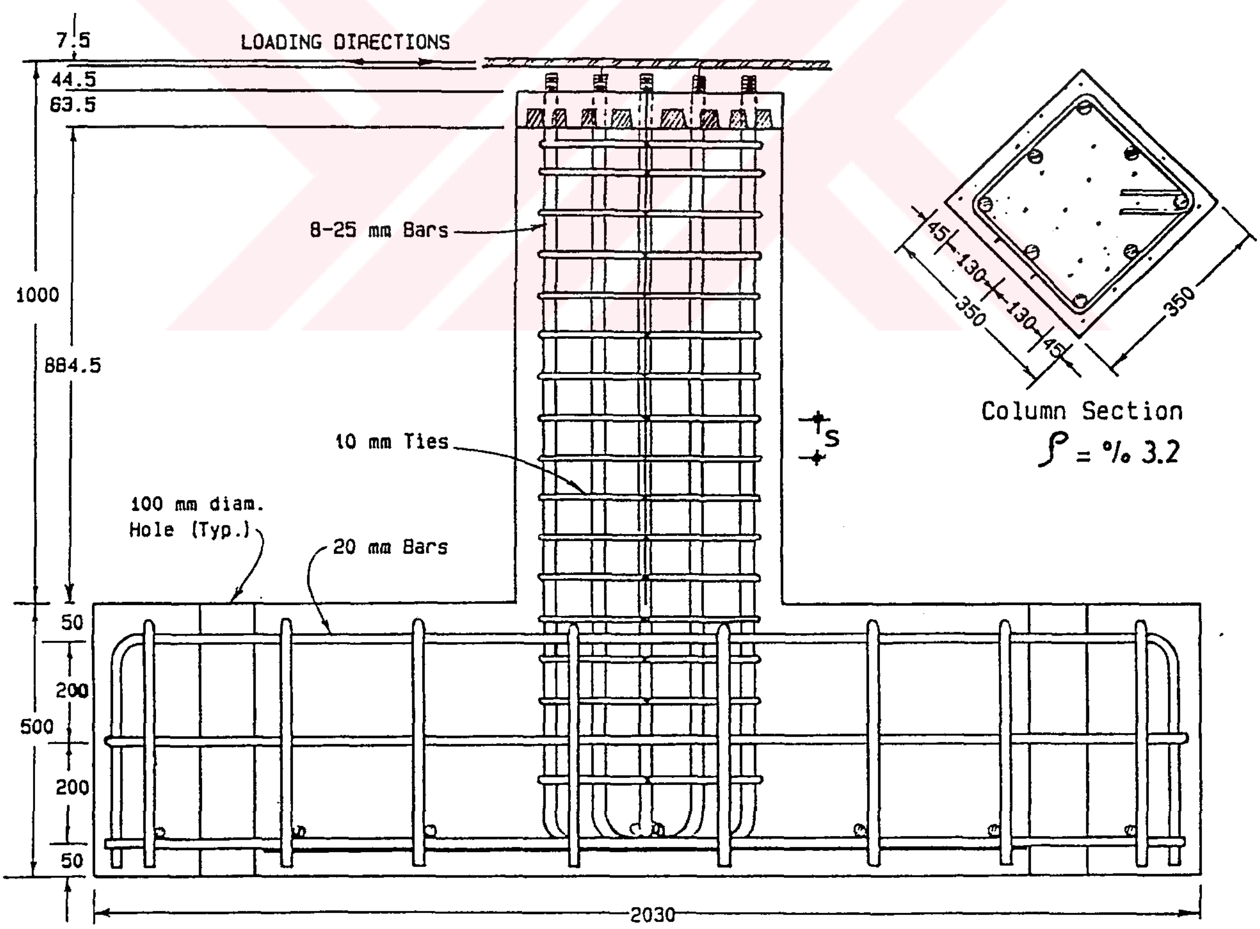


Fig. 6.4 Test specimens of Özcebe [34].

steel areas used. In the specimen 1a in which the steel area was found to be smaller, the difference might be due to the fact that the moment was applied along the weak axis (i.e. about Y-axis) and therefore the ultimate capacity could be less than the theoretical value.

Hsu [22] also tested some channel shaped columns (see Fig. 6.6). His test results and the steel areas found by the computer program developed are summuriezed in Table 6.10. Examination of these results reveal that the steel areas obtained using the computer program are somewhat higher than the areas used.

It should be noted that under high levels of axial load confinement plays an important role in ultimate capacity and also maximum strains can be much higher than the ones used in the computer program .It should be also pointed out that even 60% difference in reinforcement will results in much smaller difference in ultimate strength. Considering these the author finds the comparisons satisfactory.

Table 6.9. Evaluation of Experimental Results Obtained by Hsu [21] for L-section Column. (Ast used =9.97 cm²)

EX. no:	spec No:	section & load type	fck Kgf /cm²	fyk Kgf /cm²	Nu Kgf	Mux Kgf-cm	Muy Kgf-cm	Ast (prog.Output) cm²		
								R-A	C E B	HOG
28	1a	L	260	3572	22952	14575	102025	5.271	5.501	5.257
29	2a	L	259	3572	49553	80931	80931	10.024	10.030	10.071
30	4b	L	290	4620	16992	215800	66035	14.510	13.604	12.739
31	5b	L	290	4620	15613	218116	66625	14.753	13.792	12.908
32	6b	L	276	3999	11921	196819	50870	12.860	12.293	11.843

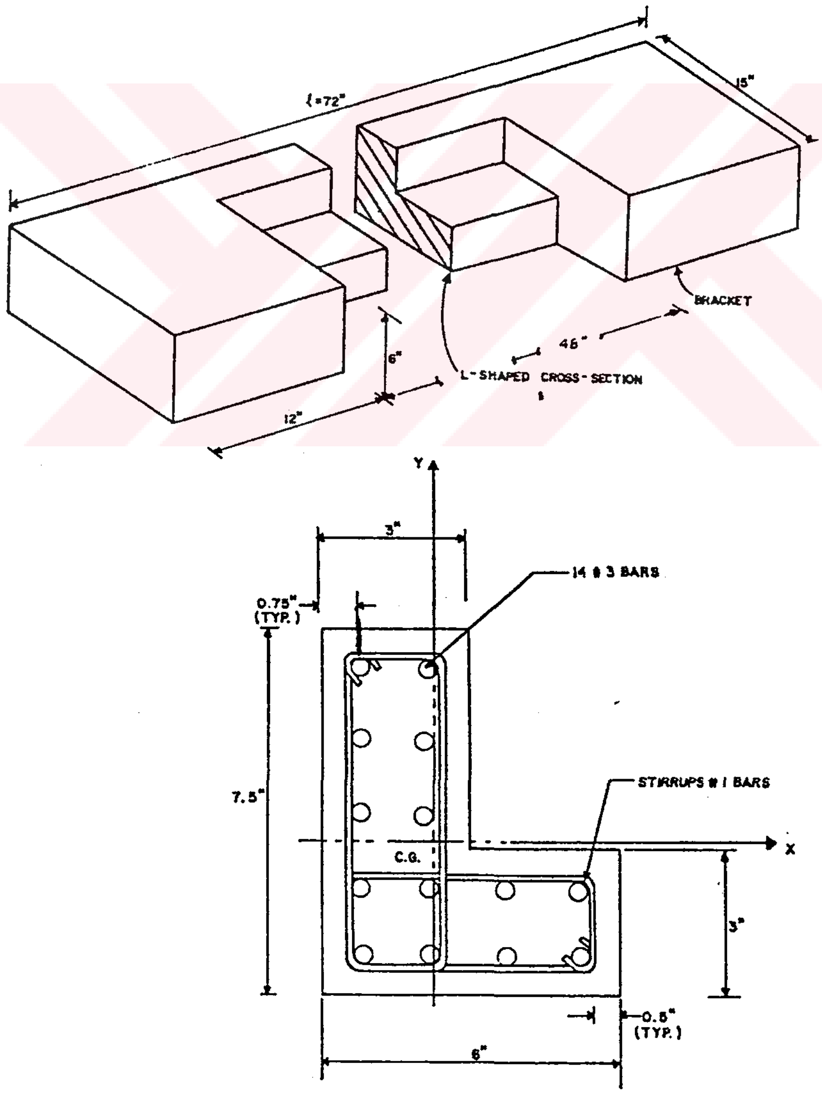


Fig. 6.5. L-Shape Specimens Tested by Hsu.

Table 6.10. Evaluation of Experimental Results Obtained by Hsu [22] on Channel Section column.(Ast used = 15.668 cm²)

EX. no:	spec No:	section & load type	fck Kgf /cm²	fyk Kgf /cm²	Nu Kgf	Mux Kgf-cm	Muy Kgf-cm	Ast (prog.Output) cm²		
								R-A	C E B	HOG
33	1c	[-A	252	3572	48178	367120	220272	22.888	22.445	21.409
34	2c	[-A	252	3572	53223	371761	243334	25.997	25.383	24.266
35	3c	[-A	252	3572	45883	349630	243334	24.085	23.400	22.306
36	4c	[-A	292	3572	47720	424233	218177	22.379	21.668	20.877
37	5c	[-A	269	3572	54143	343811	206287	20.434	20.316	19.500

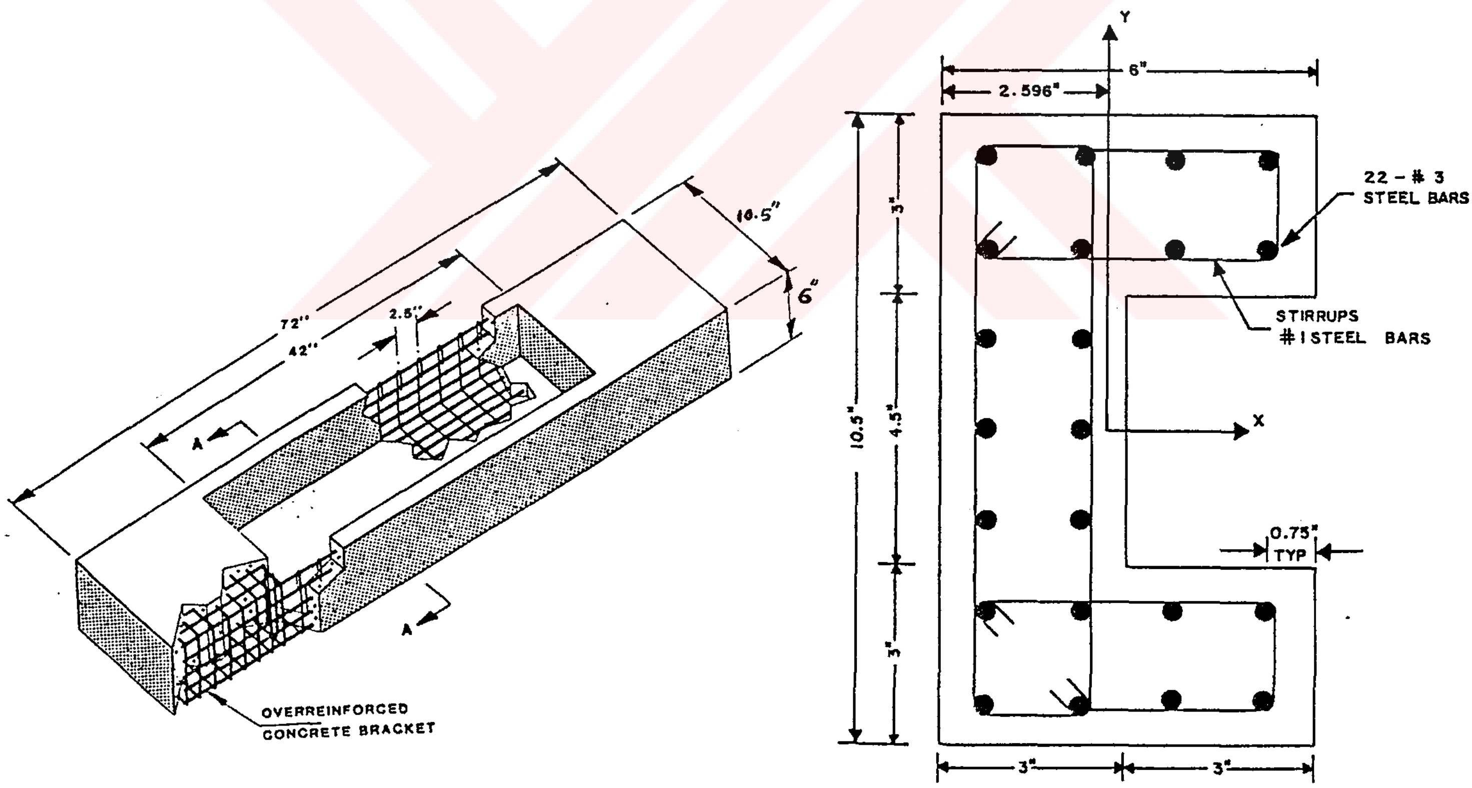


Fig. 6.6. Channel Specimens Tested by Hsu. [22]

C H A P T E R 7

CONCLUSIONS

In this study a computer program was developed for the design of reinforced concrete columns subjected to axial load and biaxial bending. The program uses a trial and adjustment procedure to find the inclination and depth of the neutral axis and area of the longitudinal steel required. The program is very efficient for the design of reinforced concrete columns subjected to axial load and biaxial bending. One of the greatest advantages of the program developed is that it can handle any arbitrary reinforced concrete cross-section having steel arranged in any pattern. Another advantage of the program is the flexibility incorporated for the concrete model. Any stress-strain relationship for concrete can be introduced to the program. In the case studies made, different stress blocks for concrete were used.

Reliability of the program was checked by comparing the results with the results obtained by existing analytical methods which are restricted to rectangular sections. Reliability was also checked by comparing the program outputs with experimental results. Test specimens considered had rectangular, L and channel shaped cross-sections.

As a result of the studies made, the following conclusions seem to be valid:

1- The computer program developed is design oriented. It is very general and can handle cross-sections having any arbitrary geometry and steel located at different locations. The program is simple to use and consumes minimum computer time.

2- Steel areas obtained using different stress blocks are not very different from each other. Therefore use of sophisticated stress blocks does not seem to be feasible. The results obtained by using the simplest block (rectangular) and area method are not very different from the ones when CEB or Hognestad blocks are used. However time consumed is about 1/30 or less.

3- Under very high axial loads the rectangular stress block leads to about 10% less steel as compared to CEB or Hognestad block. This can not be considered unsafe, since the core of columns are confined and stress-strain models used (CEB or Hognestad) are for unconfined concrete.

4 - Case studies made for comparison with other analytical methods revealed that steel areas obtained by using the program developed were almost same with the results obtained using other methods. Since the other methods are restricted to rectangular sections, the case studies included only such sections.

5-Columns tested under axial load and biaxial bending by several researchers were evaluated with the computer program developed. The test specimens considered had rectangular, L and channel shaped cross-sections. The steel areas computed for the test specimens using the computer program were not very different from the areas of reinforcement used in the test specimens. Computed areas were in general somewhat higher. However for specimens tested under very high axial loads, the steel areas predicted by the program were considerably higher. It is belived that this is due to the effect of confinement and higher crushing strains due to the confinement. In addition, under high axial loads the contribution of steel is very low, a small difference leading to high percentages.

In the future research, the effect of confinement can be studied. Also the feasibility of using a variable crushing strain instead of the constant value (ϵ_{cu} =0.003) can be investigated by using optimization techniques. However these studies recommended for future research can not be easily incorparated in design.

R E F E R E N C E S

1. Aas-Jacobsen ,A. , March 1964 ; " Biaxial Eccentricities in ultimate load design ", ACI Journal, Proceedings V.61, No.3 , PP. 293-316.

2. ACI 318-83 , 1983 ; " Building Code Requirements for Reinforced Concrete " , American Concrete Institude , Detroit , Michigan.

3. Al-Noury ,S.I. , chen ,W.F. , June 1982 ; " Behavior And Design of Reinforced And Composite Concrete Sections " , Proceedings , ASCE , Vol 108 , st6 , PP. 1266-1284.

4. Au ,T. , 1958 ; " Ultimate Strength Design of Rectangular Concrete Members Subjected to Unsymmetrical Bending " , ACI Jornal , V.29 , No. 8 , (Proceedings V.54) , PP. 657-674.

5. Bakır ,E. , Bakır ,A.R. , 1986 ; "Taşıma Gücü Yöntemi ile Kolon ve Perde Donatı Tabloları" ,I.M.O. ,Teknik Yayınevi

6. Banjar , A.M. , 1983 ; " Analysis of Reinforced Concrete Columns Subjected to Axial Load and Biaxial Bending " M.S. thesis , the University of Arizona , Civil Eng. and Eng. Mech. Dept.

7. " Betonarme Kesitlerin Hesap ve Yapım Kuralları , TS-500-84 " , 1984 , (Building Code Requirements for Reinforced Concrete) , Turkish Standarts Inst. , Ankara.

8. Bresler ,B. ,Nov.1960 ; " Design Ceriteria for Reinforced Columns Under Axial Load and Biaxial Bending " , ACI Journal , Proceedings V.57 , No. 5 , PP. 481-490.

9. Brondom-Nielsen , T. , Nov.1982 ; " Ultimate limit state of Cracked Arbitrary Concrete Sections Under Axial Load and Biaxial Bending " , Concrete International , vol 4 , no. 11 , PP. 51-55.

10. Brondom-Nielsen ,T. , Nov.-Dec. 1985 ; "Ultimate Flexural Capacity of Cracked Polygonal Concrete Sections Under Biaxial Bending " , ACI Journal , Title No.82-80 , PP. 863-869

11. Chen , W.F. , Atsuta , T. , May 1972 ; " Interaction Equations for Biaxially Loaded Sections " , Proceedings ASCE , Vol.98 , st5 , PP.1035-1052.

12. Chu ,K-H. , Pabarcius ,A. ,DEC. 1958 ; " Biaxialy Loaded Reinforced Concrete Columns " , Proceedings ASCE , V.84 , No.st 8 , Paper No. 1865.

13. Çakıroğlu ,A. , Özer ,E. , 1980 ; " Malzame ve Geometri Değişimi Bakımından Lineer Olmayan Sistemlar I. " , İ.T.Ü. University , Sayı : 1168 ,İstanbul.

14. Çakıroğlu ,A. , Özer ,E. , 1983 ; " Egik Eğilme ve Eksenel Kuvvet Etkisindeki kesitlerde Taşıma Gücü Formülleri " , Yesa Yayınları-1 , İstanbul.

15. Çokça , E. , Ersoy, U. ;" İki Doğrultuda Bileşik Eğilme Etkisindeki B/A Kesitler için Taşıma Gücü Hesap Yöntemi " Bayındırlık ve İskan Bakanlığı , Deprem Araştırma Bülteni No. 55 , Ekim 1986.

16. Dinsmore ,C.I. , Nov. 1982 ; " Column Analysis with a Programmable Calculator " ,Concrete International , vol.4 , No.11 , PP.32-36.

17. Durmuş ,A. , Eyuboğlu ,U. , 1986 , " İki Doğrultuda Bileşik Eğilme Etkisindeki Betonarme kesitlarin Taşıma Gücüne Göre Hesabı " , K.T.Ü. , Trabzon.

18. Ersoy ,U. , 1985 ; " Betonarme Temel ilkeler ve Taşıma Gücü Hesabı " , METU , Ankara.

19. Ersoy ,U. , 1988 ; " Reinforced Concrete " ,METU , Ankara

20. Farah , A. , Huggins , M.W. , July 1969 ; " Analysis of reinforced Concrete Columns Subjected to Logitudinal load Load and Biaxial Bending ", ACI Journal ,Proceedings V.66 , No.7 , PP. 569-575.

21. Hsu ,Cheng-Tzu Thomas , Dec. 1985 , " Biaxially loaded L-Shaped Reinforced Concrete Columns" , Journal of Structural Engineering ,ASCE ,V.111 ,No.12 ,PP. 2576-2595 .Also ,Errata , Journal of Structural Engineering ,ASCE ,V.114 ,No.111 ,Nov. 1988 , P. 2629.

22. Hsu ,Cheng-Tzu Thomas , May-June 1987 ; " Channel-Shaped Reinforced Concrete Compression Members under Biaxial Bending " , ACI Structural Journal , V.84 ,No.3 , PP. 201 -211 . Also , Errata , ACI Structural Journal , May-June 1988 ,p. 240.

23. Hsu , Cheng-Tzu Thomas , July-August 1989 ; " T-Shaped Reinforced Concrete members under Biaxial Bending and Axial Compression " , ACI Structural Journal , Vol.86 ,No .4 , PP. 460-468.

24. Fleming ,J.F. , Werner ,D. , March 1965 ; " Design of Columns Subjected to Biaxial Bending " , ACI Journal , Proceedings V.62 , No.3 , PP.327-342.

25. Furlong ,R.W. , March 1961 ;" Ultimate Strength of Square Columns Under Biaxially Eccentric loads " , ACI Journal , Proceedings V.57 , No.9 , PP.1129-1140.

26. Furlong ,R.W. , oct. 1979 ; " Concrete Columns Under Biaxially Eccentric Thrust " , ACI Journal , V.76 , No.10 , PP.1093-1118.

27. Gouwens ,A.J. , 1975 ; " Biaxial Bending Simplified " , Reinforced Concrete Columns , SP50-10 , ACI , Detroit , PP.233-261.

28. Heimdahl ,P.D. , Bianchini ,A.C. , 1975 ; " Ultimate Strength of Biaxially Eccentrically Loaded Concrete Columns Reinforced with High Strength Steel ", Reinforced Concrete Columns , SP 50-4 , ACI , Detroit. , pp. 93-117.

29. Kawakami ,M. , Tokuda ,H. , Kagaya ,M. , Hirata ,M. , March-Appril 1985 ; " Limit States of Cracking and Ultimate Strength of Arbitrary Concrete Sections under Biaxial Loading " , ACI Journal , Title No. 82-19 , PP.203-212.

30. Kıral ,E. , Dündar ,C. , 1987 ; " Eğik Eğilme ve Eksenel Basınç Altındaki Betonarme Kesitlerin Bilgisayar ile Hesabı " , Çukurova University , Civil Eng. Dept. Adana.

31. Lachance ,L. , March-april 1980 ; " Stress Distribution in Reinforced Concrete Sections Subjected to Biaxial Loading " , ACI Journal , V.77 , No.2 , PP.116-123.

32. Meek ,J.L. , Aug. 1963 ; " Ultimate Strength of Columns with Biaxial Eccentric Loads " , ACI Journal ,Proceedings V.60 ,No.8 , PP.1053-1064.

33. Nelson ,G. , Aktan ,A.E. ; " Response prediction of wall and T-Beam Elements in R/C Buildings and User's Guide for 'RCSA' " , Dept. of Civil Eng. , Louisiana State University , Research report 86-3.

34. Özcebe, G. , Saatçıoğlu, M. ,Jan.-Feb. 1989 ; " Response of reinforced concrete columns to simulated seismic loading " , ACI structural journal , Title no. 86-S1 , PP. 3-12.

35. Özmen ,G. , 1978 ; " Eğik Eğilme Etkisdindeki dikdörtgen Kolon Kesitlerinin Elastik Hesabı " , İ.T.Ü. Dergisi , Cilt 36 , Sayı 2.

36. Özmen ,G. , 1980 ; " Eğik Eğilme ve Normal Kuvvet Etkisindeki Betonarme Kesitlerin Taşıma Gücü Kuramına Göre Hesabı " , İ.T.Ü. Dergisi , Cilt 38 , Sayı 1-6.

37. Pannell ,F.N. , July 1960 ; " The Design of Biaxially Loaded Columns by Ultimate Load Method " , Magazine of Concrete research , V.12 , No.35.

38. Parme , A.L. , Nieves ,J.M. , Gouwens ,A. , Sept. 1966 ; " Capacity of Reinforced Rectangular Columns Subjected to Biaxial Bending " , ACI Journal , Proceedings V.63 , No.9 , PP.911-923.

39. Ramamurthy ,L.N. , 1966 ; " Investigation of the Ultimate Strength of Square and Rectangular Columns Under Biaxially Eccentric Loads " , Symposium on Reinforced Concrete Columns , Sp-13 , ACI , Detroit ,pp.263-298.

40. Shanmughasundaram ,N. , Dec. 1977 ; " Biaxial Interaction Exponent for Concrete Columns " , Proceedings , ASCE , Journal of Structural Division , V.103 , No. ST12 , PP.2295-2306.

41. Solanki ,H.T., May 1980 ; " Biaxial Bending Simplified " , Proceedings ASCE , Vol.106 , ST5 , PP.1198-1202.

42. Warner ,R.F. , May 1969; "Biaxial Moment Thrust Curvature Relations " , Proceedings ,ASCE , V.95 , ST5.

43. Weber , D.C. , Nov. 1966 ; " Ultimate Strength Design Charts for Columns with Biaxial Bending " , ACI Journal , Proceedings V.63 , PP.1205-1230.

APPENDIX: A

USER GUIDE FOR THE COMPUTER PROGRAM

APPENDIX A

USER GUIDE FOR THE COMPUTER PROGRAM

A.1. Introduction.

"Reinforced Concrete Section Design" (RCSD) is an interactive microcomputer program that operates in the "DOS" operating system. This program was developed at The Middle East Technical University as part of an M.S. thesis. The main objective of this program is to develope a practical design tool. With "RCSD" one would be able to design a reinforced concrete section under axial load and biaxial bending. This Appendix is aimed to make the engineer familiar with the programer's features.

A.2. Data Files.

Before running the program, the user should prepare two data files. One is for material properties and the other for reinforced concrete section geometry, including steel configuration.

A.2.1. Material properties Data.

The data should be written in one line with a blank between each data. The program " MATERIAL " may helps to create this data file. Note that these data files should have an extension of ".MAT"

A map of such a file is presented as the following:

File " FILENAME.MAT "

```
        COLUMN
LINE    1.........2.........3.........4.........5.........6
       ================================================================
1....  ||k1 Ec Es fcd fck fctd fyd fyk
```

Where :

$$k1 = \frac{\text{Dept of rectangular Stress-Block}}{\text{Dept of Neutral Axis}}$$

Ec = Modulus of Elasticity of Concrete.

Es = Modulus of Elasticity of Steel.

fcd = Design Strength of Concrete.

fck = Characteristic Strength of Concrete.

fyd = Design Strength of Steel.

fyk = Yield Strength of Concrete.

A.2.2. Section Geometry Data.

In this file the geometric properties of section is saved. In the first line, number of nodes that create the polygonal concrete section, and number of steel bars are given. These two are seperated by one blank between them.

In the second and following lines the coordinates of concrete nodes are presented. The coordinates of each node are in one line and each coordinate is seperated by one blank. Note that the nodes should be given in the clock-wise direction.

When the data on concrete nodes are completed, steel data (coordinates and area ratio) are given. In each line the X- and Y-coordinates and the ratio of each bars's area to the total steel of area is presented.

Note that the name of the section files can be in two forms. One by giving a file name with an extention of ".SEC" or by a file name of "SECTION" with an extention of the section number (000-999), for example "SECTION.023".

A map of such a file is presented below.

File "SECTION.#"

```
             COLUMN
  LINE     1.........2.........3.........4.........5.........6
          ||=====================================================
1.......  ||Nc Ns
2.......  ||Xc(1)  Yc(1)
3.......  ||Xc(2)  Yc(2)
4.......  ||  .      .
.......   ||  .      .
.......   ||  .      .
Nc+1...   ||Xc(Nc)  Yc(Nc)
Nc+2...   ||Xs(1)  Ys(1)  R(1)
.......   ||Xs(2)  Ys(2)  R(2)
.......   ||  .       .     .
.......   ||  .       .     .
.......   ||  .       .     .
Nc+Ns+2   ||Xs(Ns)  Ys(Ns)  R(Ns)
```

Where:

Nc = Number of Concrete Section Nodes.

Ns = Number of Steel Bars.

Xc(i) = X-Coordinate of i'th Node of Concrete Section.

Yc(i) = Y-Coordinate of i'th Node of Concrete Section.

Xs(i) = X-Coordinate of i'th Steel Bar.

Ys(i) = Y-Coordinate of i'th Steel Bar.

R (i) = Ratio of i'th longitudinal bar's area to the total steel area.

A.3. File Structure of the Computer Program Developed.

- FILE TYPE -	-FILE NAME -	- SUBROUTINES -
Include	RCSDCOM.BI	
Library	RCSDLIB.LIB	
Module	RCSD.BAS	ACI.Reduc
		Arbit.Sec.Proper
		Biaxial
		Cir.AreaMethod
		Cir.IntAreaMethod
		Circular
		Iter.Start.Value
		Max.Strain
		Pol.AreaMethod
		Pol.IntAreaMethod
		Strip.Length
		StrssStrn
		Uniaxial
	RCSDCTRL.BAS	Ast.ACI
		Ast.TS500
		Axial.Load.TS500
		Control.AlfaNE
		Control.Min.Ecc
		Control.Ybar
		RotSec.Control
	RCSDGRPH.BAS	Arbit.Sec.Window
		Graph.Draw
		GrphScr
		IntLine.Draw
		Rec.Stress.Draw
		Stress.Axis
	RCSDINP.BAS	Input.Loading
		Input.Material
		Input.Section
		Kent.Park.Pmeter
		Kind.of.Solv
		Min.Ecc.Control
		Norm.Value
		Num.Der.Val
		Peak.Stress.Inp
		Restart.Iter.Val
		Strip.Thickness
		Tensile.Strength
		Uni.Bi.Bending
	RCSDUTIL.BAS	Ast.Select
		Descriptions
		Numin
		Opening.Logo
		Output.Save
		Solver.Table
		Strip.Fname
		Table.Solv.Box
		Textin

A.4. Main Program.

Program is activated by typing "RCSD". This will Run the batch file of "RCSD.BAT" that includes the programs;

- MONITOR.EXE
- RCSD.EXE

Program "MONITOR.EXE" will check the computer graphic card and will run the essential programes related to the graphical compatibility. For example the program MSHERC.COM will be run for computers with Hercules graphic card. After this the batch file will run the main program "RCSD.EXE".

The program is quite simple in use because it has its own help commands on each page. Moreover the user can use arrow keys for selecting the choise and the user can also use the "Page Down" and "Page Up" keys fo the new pages of data input. The first page of the input pages is presented in Fig.A.1.

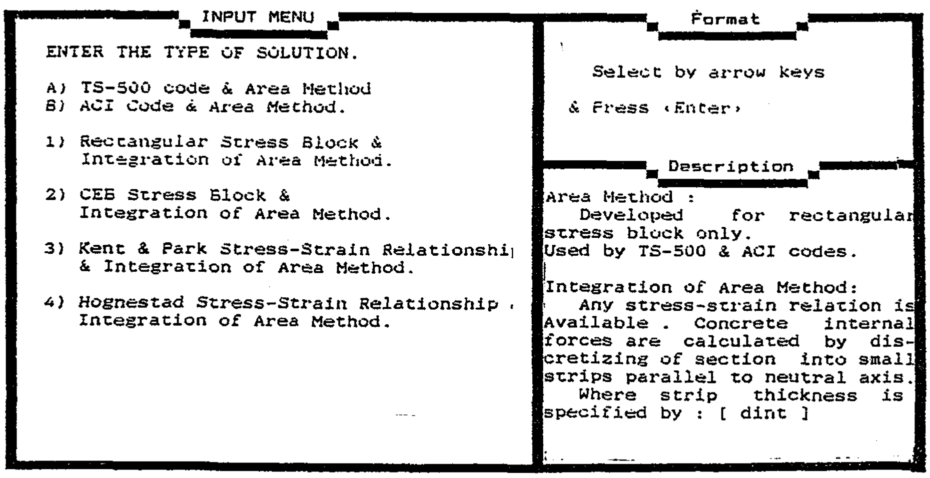


Fig. A.1. Sample Input Screen.

Details of the input form will not be presented because the sample input and output that will be presented in "Appendix C" should be good enough in explanation of these inputs.

The solving screen (Fig. A.2.) will include all of the data and the section shape. Using the Arrow keys the last modification on the data is also possible. Function keys assigned as described below.

F1 - Will select the reinforcing bars. Note that the selection of reinforcing bars is valid only in cases where the steel areas found are in cm^2 unit.

F2 - Will save the output.

F3 - Solve the problem.

F4 - Restart the program.

F5 - Terminate the program.

F6 - Returns to the solver table.

The program will come back to the solver screen to change the given starting values for iteration when any undesirable event occures while iterating. These are ; (a) The inclination of neutral axis is less than zero or greater than 90 degree. (b) Neutral axis depth is found to be less than zero. (c) The jacobian matrix is found to be zero.

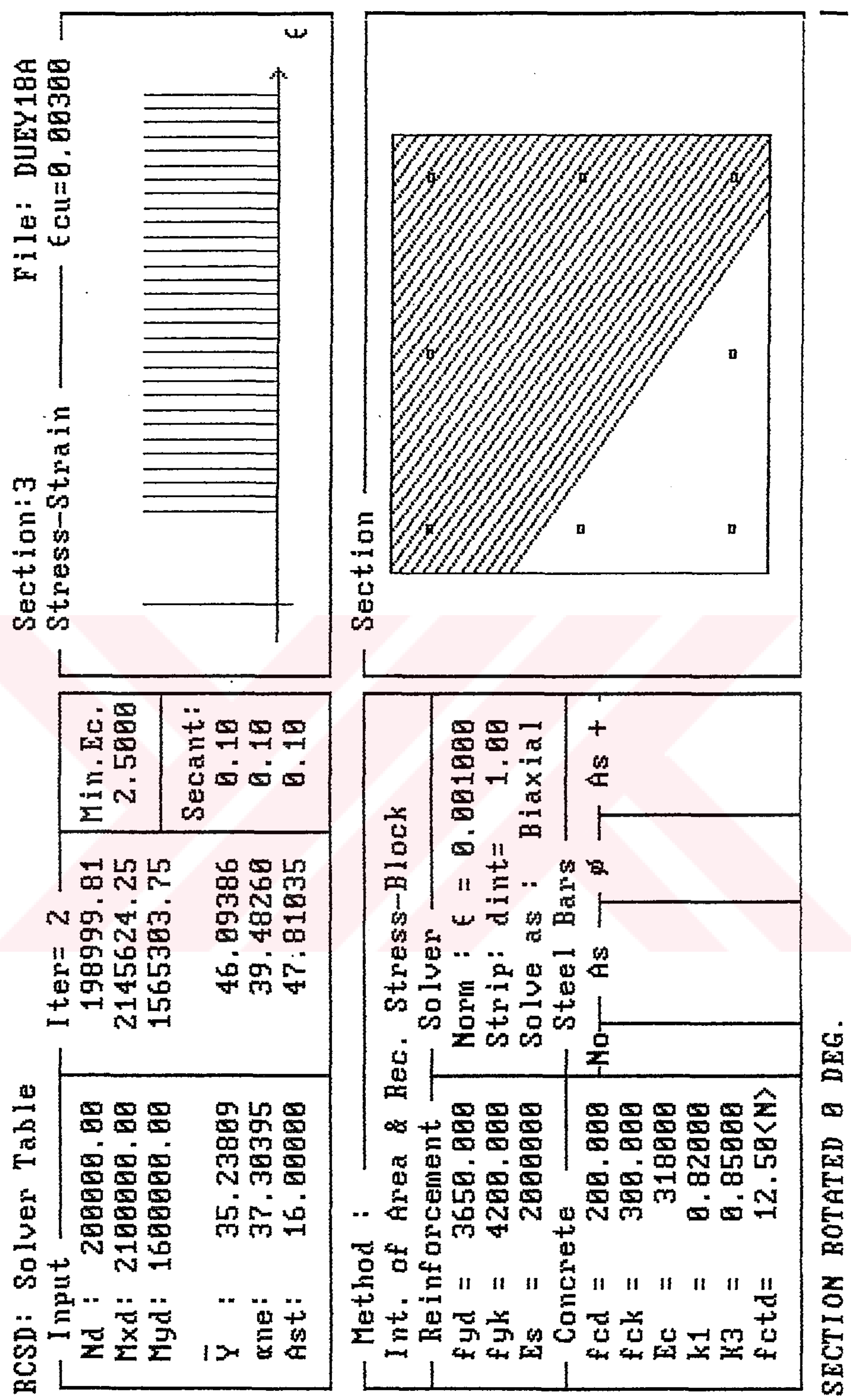


Fig. A.2. Solver Screen.

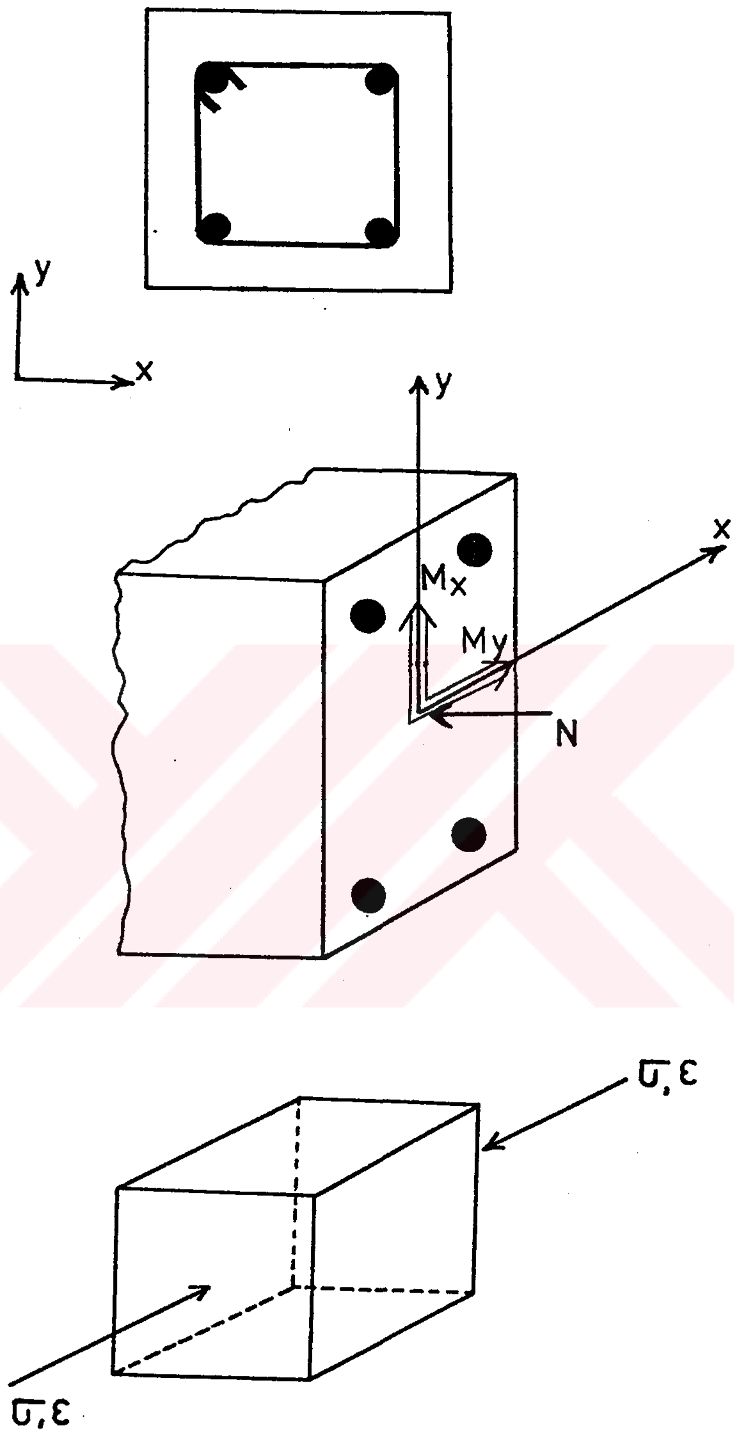


Fig. A.3. Positive sign conventions.

APPENDIX: B

LISTING OF THE PROGRAM

A P P E N D I X   B

LISTING OF THE PROGRAM

B.1. General.

As mentioned earlier main program consists of many modules. Here only the list of the essential modules are presented. The module "RCSDUTIL.BAS" which is related to the utilities of the program (descriptions and oppening titles) is not presented. The list of the programs which were developed to prepare data files are also not presented. These programs are; (a) RCSDCAD , developed to create sections using the AutoCad computer program. This program is written in Autolisp and QuickBasic programing languages. (b) RCSDSEC. This program which is written in Pascal Programing Language is also developed to help us in preparing data for the complex cross-sections shapes.

B.2. RCSD

```
DECLARE SUB Ast.Select ()
DECLARE SUB Textin (T$, Max.Length%, Caps.On%, Exit.Code%, textinbg%
            , textinfg%)
DECLARE SUB Iter.Start.Value ()
DECLARE SUB ReStart.Iter.Val ()
DECLARE SUB Numin (inn#, Max.Digits%, Max.Places%, Exit.Code%, textinbg%
            , textinfg%)
DECLARE SUB Uni.Bi.Bending ()
DECLARE SUB Ast.ACI ()
DECLARE SUB ACI.Reduc ()
DECLARE SUB Output.Save ()
DECLARE SUB Restart.Pmeters ()
DECLARE SUB Axial.Load.TS500 ()
DECLARE SUB Ast.TS500 ()
DECLARE SUB Peak.Stress.Inp ()
DECLARE SUB Descriptions (DESCRIP!)
DECLARE SUB Table.Layout ()
DECLARE SUB Pol.AreaMethod ()
DECLARE SUB Pol.IntAreaMethod ()
DECLARE SUB Cir.AreaMethod ()
DECLARE SUB Cir.IntAreaMethod ()
DECLARE SUB StrssStrn ()
DECLARE SUB Circular ()
DECLARE SUB Uniaxial ()
DECLARE SUB Biaxial ()
DECLARE SUB Rec.Stress.Draw ()
DECLARE SUB Max.Strain ()
DECLARE SUB Arbit.Sec.Proper ()
DECLARE SUB Strip.Length ()
DECLARE SUB Solver.Table ()
DECLARE SUB Opening.Logo ()
DECLARE SUB Stress.Axis ()
DECLARE SUB Arbit.Sec.Window ()
DECLARE SUB IntLine.Draw ()
DECLARE SUB Graph.Draw ()
DECLARE SUB GrphScr ()
DECLARE SUB Control.Ybar (Cont$)
DECLARE SUB Control.AlfaNE (Cont$)
DECLARE SUB Solv.Again.Cont (Answer$)
DECLARE SUB Min.Ecc.Control ()
DECLARE SUB RotSec.Control ()
DECLARE SUB Control.Min.Ecc ()
DECLARE SUB Kent.Park.Pmeter ()
DECLARE SUB Strip.Thickness ()
DECLARE SUB Tensile.Strength ()
DECLARE SUB Kind.of.Solv ()
DECLARE SUB Norm.Value ()
DECLARE SUB Num.Der.Val ()
DECLARE SUB Input.Loading ()
DECLARE SUB Input.Material ()
DECLARE SUB Input.Section ()
DECLARE FUNCTION Monitor% ()
DECLARE FUNCTION EGAMem% ()
```

```
' $INCLUDE: 'RCSDCOM.BI'

'        **********************************************************
'        *                                                        *
'        *                    *** R C S D ***                     *
'        *                                                        *
'        *             A Program For :                            *
'        *            Reinforced Concrete Section Design          *
'        *                                                        *
'        *             PROGRAMMED BY :                            *
'        *                        MARJANI F.                      *
'        *                        MAY 1989 , Ver. 5.0             *
'        *                                                        *
'        *           AS PART OF AN MS THESIS SUPERVISED BY:       *
'        *                                                        *
'        *                        U. ERSOY                        *
'        *                                                        *
'        *              PROF. DR. OF CIVIL ENGINEERING            *
'        *             MIDDLE EAST TECHNICAL UNIVERSITY           *
'        *                                                        *
'        *            ***    DISCLAIMER STATEMENT    ***          *
'        *                                                        *
'        *      ITS USE BY OTHERS IS SUBJECT TO THE FOLLOWING     *
'        *      CONDITIONS.                                       *
'        *                                                        *
'        *           (1) CONSULTING SERVICE AND ASSISTANCE  WITH  *
'        *               PROGRAM CONVERSION BE NEEDED.            *
'        *                                                        *
'        *           (2) NO WARRANTY IS EXPRESSED OR IMPLIED AS TO *
'        *               THE ACCURACY, USEFULNESS, OR COMPLETENESS *
'        *               OF THIS PROGRAM.                         *
'        *                                                        *
'        **********************************************************

   CLEAR , , 2000
   Graphcard% = Monitor%
   Memory% = EGAMem% * 64
   GrphScr
   SCREEN 0, 0, 0
   Opening.Logo

Prog.Head:
   SCREEN 0, 0, 0

   DIM x1(3), x2(3), F1(3), d(3, 3), B(3, 3), a(2, 2)
   PI# = 4 * ATN(1)

Inp.K.O.S:

   Kind.of.Solv                'Input type of solving.........
   IF Exit.Code% = -73 THEN GOTO Inp.K.O.S

                               'Input Parameters Related to
Inp.S.T:                               'Kind of Solving
```

```
   IF Kind = 0 THEN
      EpsCu = .003
   ELSE
      Strip.Thickness          'Input strip thickness for int..
      IF Exit.Code% = -73 THEN GOTO Inp.K.O.S
      Tensile.Strength         'consider concrete tensile
                               'strength or not
      IF Exit.Code% = -73 THEN GOTO Inp.S.T
   END IF
      IF Exit.Code% = -73 THEN GOTO Inp.K.O.S:

Inp.K.P.P:
   IF Kind = 3 THEN
      Kent.Park.Pmeter         'Input Ecu & E50h for K&P ......
   END IF
   IF Exit.Code% = -73 THEN GOTO Inp.S.T

Inp.P.S.I:
   Peak.Stress.Inp             'Input the peak stress in
                               'concrete stress - Block........
   IF Exit.Code% = -73 THEN GOTO Inp.K.P.P

Inp.M.E.C:
   Min.Ecc.Control             'Min Eccentricity Control.......
   IF Exit.Code% = -73 THEN GOTO Inp.P.S.I

Inp.M:
   Input.Material              'Material Properties input......
   IF Exit.Code% = -73 THEN GOTO Inp.M.E.C

Inp.S:
   Input.Section               'Section file Input.............
   IF Exit.Code% = -73 THEN GOTO Inp.M

Inp.L:
   Input.Loading               'Load Input.....................
   IF Exit.Code% = -73 THEN GOTO Inp.S

Inp.N.D.V:
   Num.Der.Val                 'Input Num. der. Values.........
  IF Exit.Code% = -73 THEN GOTO Inp.L

Inp.N.V:
   Norm.Value                  'Input Norm value (E)...........
   IF Exit.Code% = -73 THEN GOTO Inp.N.D.V

Inp.U.B.B:
   IF NOCSN <> 0 THEN
      Uni.Bi.Bending           'Input whether it is solved
                               'uniaxial or biaxial............
      IF Exit.Code% = -73 THEN GOTO Inp.N.V

      Arbit.Sec.Proper         'Compute the poligonal section
                               'properties.....................
   END IF
```

```
   IF Exit.Code% = -73 THEN GOTO Inp.N.V
   CLS                          'Main Prog. Started.............

   IF EccMin > 0 AND Nd > 0 THEN
        Control.Min.Ecc         'Minimum eccentricity control....
   END IF

   Iter.Start.Value             'starting values for iteration
                                'computed........................
Iter.Head:
   GrphScr
   Arbit.Sec.Window
   WINDOW (Winx1, Winy1)-(Winx2, Winy2)

   Nd = NdOrg
   Mxd = MxdOrg
   Myd = MydOrg
   Fcd = FcdOrg
   Fyd = FydOrg

Solv.Head:
   Solver.Table

   Graph.Draw

Restart.head:
   ReStart.Iter.Val
   IF Exit.Code% = -73 THEN SCREEN 0, 0, 0: GOTO Inp.U.B.B
   IF Exit.Code% <> -61 THEN
      IF Exit.Code% > -65 AND Exit.Code% < -59 THEN GOTO Do.Func.key
   END IF
   Iter = 0

   IF Kind = 0 THEN
      Rec.Stress.Draw
      IF ArMethCode$ = "ACI" THEN
         ACI.Reduc
         '*** Axial.Load.ACI(not Assigned)
      ELSE
         Axial.Load.TS500
      END IF
   IF NdIgnor$ = "N" THEN GOTO Restart.head
   END IF

   IF NOCSN = 0 THEN
      '   ****************************************************************
      '   *                  M A I N   P R O G R A M                     *
      '   *                FOR CIRCULAR CROSS SECTIONS.                   *
      '   ****************************************************************
      Md = SQR(Mxd * Mxd + Myd * Myd)
4060
      CIRCLE (0, 0), DE / 2, , , , AspRatio
      CIRCLE (0, 0), DI / 2, , , , AspRatio
```

```
4070
      Cont$ = ""
      Control.Ybar Cont$
      IF Cont$ = "N" THEN GOTO Restart.head

      Circular

      IF d = 0 THEN GOTO Restart.head

      IF Criter > Epsilon THEN
         IF Iter > 16 THEN
LOCATE 25, 1: PRINT "PROGRAM CAN'T OBTAIN THE RESULTS. <Strike Any Key>";
            a$ = ""
            DO
            a$ = INKEY$
            LOOP WHILE a$ = ""
            GOTO Restart.head
         END IF
         Ybar = x2(1)
         Ast = x2(2)
         GOTO 4060
      END IF
   ELSE
      ' ** i.e. Polygonal Section
      RotSec.Control
      XYC(NOCSN + 1, 1) = XYC(1, 1)
      XYC(NOCSN + 1, 2) = XYC(1, 2)

      IF UnBi$ = "U" THEN
    '      ***************************************************************
    '      *                M A I N   P R O G R A M                      *
    '      *    FOR ARBITRARY CROSS SECTION AND  UNIAXIAL BENDING        *
    '      ***************************************************************
Uni.Iter.Head:
         Cont$ = ""
         Control.Ybar Cont$
         IF Cont$ = "N" THEN GOTO Restart.head

         Uniaxial
         IF d = 0 THEN GOTO Prog.End
         IF Kind = 0 AND AreaCC <= 0 THEN GOTO Restart.head

         Criter = Criter ^ .5
         IF Criter > Epsilon THEN
            IF Iter > 16 THEN
LOCATE 25, 1: PRINT "PROGRAM CAN'T OBTAIN THE RESULTS. <Strike Any Key>";
               a$ = ""
               DO
               a$ = INKEY$
               LOOP WHILE a$ = ""
               GOTO Restart.head
            END IF
            Ybar = x2(1): Ast = x2(2)
            GOTO Uni.Iter.Head
         END IF
```

```
   ELSE
      '*** biaxial bending:
      IF Kind = 0 AND NOCSN = 0 THEN
          Graph.Draw
      END IF
  '      ************************************************************
  '      *                    M A I N   P R O G R A M               *
  '      *     FOR ARBITRARY CROSS SECTIONS UNDER BIAXIAL BENDING.   *
  '      ************************************************************
Bi.Iter.Head:
      Cont$ = ""
      Control.Ybar Cont$
      Control.AlfaNE Cont$
      IF Cont$ = "N" THEN GOTO Restart.head

      Biaxial
      IF d = 0 THEN GOTO Restart.head
      IF Kind = 0 AND AreaCC <= 0 THEN GOTO Restart.head

      IF Criter > Epsilon THEN
         IF Iter > 16 THEN
LOCATE 25, 1: PRINT "PROGRAM CAN'T OBTAIN THE RESULTS. <Strike Any Key>";
            a$ = ""
            DO
            a$ = INKEY$
            LOOP WHILE a$ = ""
            GOTO Restart.head
         END IF
         Ybar = x2(1)
         AlfaNE = x2(2)
         Ast = x2(3)
         GOTO Bi.Iter.Head
      END IF
      Ybar = x2(1)
      AlfaNE = x2(2)
      Ast = x2(3)
   END IF

END IF

IF REFL$ = "Y" THEN
   AlfaNE = 90 - AlfaNE: Ybar = TAN(AlfaNE / 180 * PI#) * Ybar
END IF

LOCATE 7, 22: PRINT USING "####.#####"; Ybar
LOCATE 8, 22: PRINT USING "####.#####"; AlfaNE
LOCATE 9, 22: PRINT USING "####.#####"; Ast

IF Kind = 0 THEN
   IF ArMethCode$ = "ACI" THEN
      Ast.ACI
   ELSE
      Ast.TS500
   END IF
END IF
```

```
   IF AstMax$ = "N" THEN GOTO Prog.End

Prog.End:
   LOCATE 25, 1: PRINT "| F1:select bar| F2:Save Output| F3:Solve|
                          F4:Restart| F5:End| F6:Solver Table |";
   LOCATE 1, 36
   Textin T$, 1, Caps.On%, Exit.Code%, -1, -1
   LOCATE 1, 36: PRINT " ";
Do.Func.key:
   IF Exit.Code% = -63 THEN END

   IF Exit.Code% = -61 THEN GOTO Iter.Head

   IF Exit.Code% = -62 THEN
      Funit% = FREEFILE
      OPEN "O", #Funit%, "GECICI.DAT"
      PRINT #Funit%, Graphcard%, AspRatio
      CLOSE #Funit%
      CLEAR
      Funit% = FREEFILE
      OPEN "I", #Funit%, "GECICI.DAT"
      INPUT #Funit%, Graphcard%, AspRatio
      CLOSE #Funit%
      KILL "GECICI.DAT"
      GOTO Prog.Head
   END IF

   IF Exit.Code% = -60 THEN Output.Save: GOTO Solv.Head:
   IF Exit.Code% = -64 THEN GOTO Restart.head
   IF Exit.Code% = -59 THEN Ast.Select: GOTO Do.Func.key
   GOTO Prog.End

SUB ACI.Reduc

   IF Nd < 0 THEN
      CapReduc = .75
   ELSE
      NdRatio = Nd / AreaGr / Fck
      IF NOCSN = 0 THEN
         IF NdRatio < .1 THEN
            CapReduc = .75 + 1.5 * (.1 - NdRatio)
         ELSE
            CapReduc = .75
         END IF
      ELSE
         IF NdRatio < .1 THEN
            CapReduc = .7 + 2 * (.1 - NdRatio)
         ELSE
            CapReduc = .7
         END IF
      END IF
   END IF
   Nd = Nd / CapReduc
   Mxd = Mxd / CapReduc
```

```
   Myd = Myd / CapReduc
   Fcd = Fck
   Fyd = Fyk
LOCATE 12, 27: PRINT ", Ò="; CapReduc

END SUB

SUB Arbit.Sec.Proper

'       ***************************************************************
'       *    THIS SUBROUTINE OBTAIN:                                  *
'       *      MAXIMUM AND MINIMUM COORDINATES OF CONCRETE SECTION.   *
'       *      AREA AND CENTER OF GRAVITY OF CONCRETE SECTION.        *
'       ***************************************************************

   CSXmax = -1E+10: CSYmax = -1 - E10
   CSXmin = 1E+10: CSYmin = 1E+10
   AreaGr = 0: SX = 0: SY = 0

   FOR I = 1 TO NOCSN
      IF CSXmax < XYC(I, 1) THEN CSXmax = XYC(I, 1)
      IF CSXmin > XYC(I, 1) THEN CSXmin = XYC(I, 1)
      IF CSYmax < XYC(I, 2) THEN CSYmax = XYC(I, 2)
      IF CSYmin > XYC(I, 2) THEN CSYmin = XYC(I, 2)
      DELTAX = XYC(I + 1, 1) - XYC(I, 1)
      DELTAY = XYC(I + 1, 2) - XYC(I, 2)
      AreaGr = AreaGr + .5 * DELTAX * (XYC(I + 1, 2) + XYC(I, 2))
SX=SX+DELTAX*(XYC(I,2)*(XYC(I,2)+XYC(I+1,2))+XYC(I+1,2)*XYC(I+1,2))/6
SY=SY-DELTAY*(XYC(I,1)*(XYC(I,1)+XYC(I+1,1))+XYC(I+1,1)*XYC(I+1,1))/6
   NEXT I

   XgGros = SY / AreaGr: YgGros = SX / AreaGr


END SUB

SUB Biaxial
   IF Kind = 0 THEN Pol.AreaMethod ELSE Pol.IntAreaMethod

   x1(1) = Ybar: x1(2) = AlfaNE: x1(3) = Ast
   F1(1) = N: F1(2) = Mx: F1(3) = My

   Ybar = Ybar + dYbar
   IF Kind = 0 THEN Pol.AreaMethod ELSE Pol.IntAreaMethod
   d(1, 1) = (N - F1(1)) / dYbar
   d(2, 1) = (Mx - F1(2)) / dYbar
   d(3, 1) = (My - F1(3)) / dYbar
   Ybar = Ybar - dYbar

   AlfaNE = AlfaNE + dAlfa
   IF Kind = 0 THEN Pol.AreaMethod ELSE Pol.IntAreaMethod
   d(1, 2) = (N - F1(1)) / dAlfa
   d(2, 2) = (Mx - F1(2)) / dAlfa
   d(3, 2) = (My - F1(3)) / dAlfa
   AlfaNE = AlfaNE - dAlfa
```

```
Ast = Ast + dAst
IF Kind = 0 THEN Pol.AreaMethod ELSE Pol.IntAreaMethod
d(1, 3) = (N - F1(1)) / dAst
d(2, 3) = (Mx - F1(2)) / dAst
d(3, 3) = (My - F1(3)) / dAst
Ast = Ast - dAst

d = d(1, 1) * d(2, 2) * d(3, 3) + d(1, 2) * d(2, 3) * d(3, 1)
d = d + d(1, 3) * d(2, 1) * d(3, 2)
d = d - d(1, 1) * d(2, 3) * d(3, 2) - d(1, 2) * d(2, 1) * d(3, 3)
d = d - d(1, 3) * d(2, 2) * d(3, 1)

IF d = 0 THEN
   LOCATE 25, 1: PRINT SPACE$(80);
   LOCATE 25, 1: PRINT "DET=0 <Strike Any Key>";
   BEEP
   a$ = ""
   DO
   a$ = INKEY$
   LOOP WHILE a$ = ""
ELSE
   B(1, 1) = (d(2, 2) * d(3, 3) - d(2, 3) * d(3, 2)) / d
   B(1, 2) = (d(3, 2) * d(1, 3) - d(1, 2) * d(3, 3)) / d
   B(1, 3) = (d(1, 2) * d(2, 3) - d(2, 2) * d(1, 3)) / d
   B(2, 1) = (d(2, 3) * d(3, 1) - d(2, 1) * d(3, 3)) / d
   B(2, 2) = (d(1, 1) * d(3, 3) - d(3, 1) * d(1, 3)) / d
   B(2, 3) = (d(2, 1) * d(1, 3) - d(1, 1) * d(2, 3)) / d
   B(3, 1) = (d(2, 1) * d(3, 2) - d(2, 2) * d(3, 1)) / d
   B(3, 2) = (d(3, 1) * d(1, 2) - d(1, 1) * d(3, 2)) / d
   B(3, 3) = (d(1, 1) * d(2, 2) - d(2, 1) * d(1, 2)) / d

   x2(1) = 0
   x2(2) = 0
   x2(3) = 0

   Criter = 0

   F1(1) = F1(1) - Nd
   F1(2) = F1(2) - Mxd
   F1(3) = F1(3) - Myd

   FOR I = 1 TO 3
      FOR J = 1 TO 3
         d(I, J) = -B(I, J)
      NEXT J
   NEXT I

   FOR I = 1 TO 3
      FOR J = 1 TO 3
         x2(I) = x2(I) + d(I, J) * F1(J)
      NEXT J
      Criter = Criter + x2(I) ^ 2
      x2(I) = x1(I) + x2(I)
   NEXT I
```

```
      LOCATE 2, 27: PRINT Iter
      Iter = Iter + 1

      Criter = Criter ^ .5

      IF REFL$ = "Y" THEN
         unkn1 = 90 - AlfaNE: unkn2 = TAN(AlfaNE / 180 * PI#) * Ybar
      ELSE
         unkn1 = x2(1): unkn2 = x2(2)
      END IF

      LOCATE 7, 22: PRINT USING "####.#####"; unkn1
      LOCATE 8, 22: PRINT USING "####.#####"; unkn2
      LOCATE 9, 22: PRINT USING "####.#####"; x2(3)

      LOCATE 3, 20: PRINT USING "#########.##"; F1(1) + Nd
      LOCATE 4, 20: PRINT USING "#########.##"; F1(2) + Mxd
      LOCATE 5, 20: PRINT USING "#########.##"; F1(3) + Myd
   END IF
END SUB

SUB Cir.AreaMethod

' ***************  Subroutine for Area Method ***************************

   IF Ybar >= DE THEN
      FI = 2 * PI#
   ELSE
      IF Ybar = DE / 2 THEN
         FI = PI#
      ELSE
         Y = DE / 2 - Ybar
         x = (DE * DE / 4 - Y * Y) ^ .5
         FI = 2 * ATN(x / Y)
         IF Y < 0 THEN
            FI = PI# + 2 * ATN(ABS(Y) / x)
         END IF
      END IF
   END IF

   FEXT = DE * DE / 8 * (FI - SIN(FI))
   EMEXT = 2 / 3 * (DE ^ 3 / 8 * (SIN(FI / 2)) ^ 3) / FEXT
   DD = DE - DI

   IF (DI <= 0) OR (Ybar <= DD / 2) THEN
      FINT = 0
      EMINT = 0
   ELSE
      IF Ybar >= (DD / 2 + DI) THEN
         FI = 2 * PI#
      ELSE
         IF Ybar = DE / 2 THEN
            FI = PI#
         ELSE
```

```
            Y = DI / 2 - Ybar + DD / 2
            x = (DI * DI / 4 - Y * Y) ^ .5
            FI = 2 * ATN(x / Y)
            IF Y < 0 THEN FI = PI# + 2 * ATN(ABS(Y) / x)
         END IF
      END IF
      FINT = DI * DI / 8 * (FI - SIN(FI))
      EMINT = 2 / 3 * (DI ^ 3 / 8 * (SIN(FI / 2)) ^ 3) / FINT
   END IF

   ANET = FEXT - FINT
   EM = (EMEXT * FEXT - EMINT * FINT) / ANET
   N = .85 * Fcd * ANET
   M = .85 * Fcd * ANET * EM

   FOR I = 1 TO NOSBAR
      YI = Rbar * COS((I - 1) * 2 * PI# / NOSBAR)
      Xi = Rbar * SIN((I - 1) * 2 * PI# / NOSBAR)
      CIRCLE (Xi, YI), DE / 100
      HI = Ybar / K1 - DE / 2 + YI
      SIGSI = Es * EpsCu * HI * K1 / Ybar
      IF SIGSI < -Fyd THEN SIGSI = -Fyd
      IF SIGSI > Fyd THEN SIGSI = Fyd
      FORCEI = Ast * SIGSI * XYS(I, 3)
      N = N + FORCEI
      M = M + FORCEI * HI
   NEXT I
END SUB

SUB Cir.IntAreaMethod

' *********  Subroutine for Integration of area Method ******************

   NC = 0: MC = 0
   Max.Strain
   HNEMAX = Ybar

   col% = 43

   FOR ROW% = 3 TO 9
      LOCATE ROW%, col%: PRINT SPACE$(37);
   NEXT ROW%

   FOR ROW% = 12 TO 23
      LOCATE ROW%, col%: PRINT SPACE$(37);
   NEXT ROW%

   Stress.Axis

   CIRCLE (0, 0), DE / 2, , , , AspRatio
   CIRCLE (0, 0), DI / 2, , , , AspRatio

   IF Ybar >= DE THEN
      YSTART = DE - dint / 2
   ELSE
```

```
   YSTART = Ybar - dint / 2
END IF

FOR YINT = YSTART TO 0 STEP -dint

   IF YINT > (DE - DI) / 2 AND YINT < (DE + DI) / 2 THEN
      HI = YINT - (DE - DI) / 2
      LI = 2 * (HI * (DI - HI)) ^ .5
   ELSE
      LI = 0
   END IF

   LE = 2 * (YINT * (DE - YINT)) ^ .5
   LNET = LE - LI
   LINE (-LE / 2, DE / 2 - YINT)-(-LI / 2, DE / 2 - YINT)
   LINE (LI / 2, DE / 2 - YINT)-(LE / 2, DE / 2 - YINT)
   HDINT = Ybar - YINT
   StrssStrn
   FDINT = LNET * dint * Stress
   NC = NC + FDINT
   MC = MC + FDINT * (DE / 2 - YINT)
NEXT YINT

YINTEND = DE
IF KindT$ = "Y" THEN

   IF Ybar + dint / 2 < DE THEN
      YSTART = Ybar + dint / 2
      YINTEND = DE
      FOR YPint = YSTART TO YINTEND STEP dint

         IF YPint > (DE - DI) / 2 AND YPint < (DE + DI) / 2 THEN
            HI = YINT - (DE - DI) / 2
            LI = 2 * (HI * (DI - HI)) ^ .5
         ELSE
            LI = 0
         END IF
         LE = 2 * (YPint * (DE - YPint)) ^ .5
         LNET = LE - LI

         LINE (-LE / 2, DE / 2 - YPint)-(-LI / 2, DE / 2 - YPint)
         LINE (LI / 2, DE / 2 - YPint)-(LE / 2, DE / 2 - YPint)

         HDINT = Ybar - YPint
         StrssStrn
         FDINT = LNET * dint * Stress
         NC = NC + FDINT
         MC = MC + FDINT * (DE / 2 - YINT)
      NEXT YPint
   END IF
END IF

N = NC
M = MC
```

```
   FOR I = 1 TO NOSBAR
      YI = Rbar * COS((I - 1) * 2 * PI# / NOSBAR)
      Xi = Rbar * SIN((I - 1) * 2 * PI# / NOSBAR)
      CIRCLE (Xi, YI), DE / 100
      HI = (YI + Ybar - DE / 2)
      SIGSI = EpsCu * HI / Ybar * Es
      IF SIGSI < -Fyd THEN SIGSI = -Fyd
      IF SIGSI > Fyd THEN SIGSI = Fyd
      FORCEI = Ast * SIGSI * XYS(I, 3)
      N = N + FORCEI
      M = M + FORCEI * YI
   NEXT I
END SUB

SUB Circular

   IF Kind = 0 THEN
      Cir.AreaMethod
   ELSE
      Cir.IntAreaMethod
   END IF

   x1(1) = Ybar: x1(2) = Ast
   F1(1) = N: F1(2) = M

   Ybar = Ybar + dYbar
   IF Kind = 0 THEN
      Cir.AreaMethod
   ELSE
      Cir.IntAreaMethod
   END IF

   d(1, 1) = (N - F1(1)) / dYbar
   d(2, 1) = (M - F1(2)) / dYbar
   Ybar = Ybar - dYbar

   Ast = Ast + dAst
   IF Kind = 0 THEN
      Cir.AreaMethod
   ELSE
      Cir.IntAreaMethod
   END IF

   d(1, 2) = (N - F1(1)) / dAst
   d(2, 2) = (M - F1(2)) / dAst
   Ast = Ast - dAst

   d = d(1, 1) * d(2, 2) - d(2, 1) * d(1, 2)

   IF d = 0 THEN
      LOCATE 25, 1: PRINT SPACE$(80);
      LOCATE 25, 1: PRINT "DET=0 <Strike Any Key>";
      BEEP
      a$ = ""
```

```
      DO
      a$ = INKEY$
      LOOP WHILE a$ = ""
      ELSE
      B(1, 1) = d(2, 2) / d
      B(1, 2) = -d(1, 2) / d
      B(2, 1) = -d(2, 1) / d
      B(2, 2) = d(1, 1) / d

      x2(1) = 0: x2(2) = 0: Criter = 0

      F1(1) = F1(1) - Nd: F1(2) = F1(2) - Md

      FOR I = 1 TO 2
         FOR J = 1 TO 2
            d(I, J) = -B(I, J)
         NEXT J
      NEXT I

      FOR I = 1 TO 2
         FOR J = 1 TO 2
            x2(I) = x2(I) + d(I, J) * F1(J)
         NEXT J
         Criter = Criter + x2(I) ^ 2
         x2(I) = x1(I) + x2(I)
      NEXT I

      LOCATE 2, 27: PRINT Iter
      Iter = Iter + 1
      Criter = Criter ^ .5

      LOCATE 7, 22: PRINT USING "####.#####"; x2(1)
      LOCATE 9, 22: PRINT USING "####.#####"; x2(2)

      LOCATE 3, 20: PRINT USING "#########.##"; F1(1) + Nd
      LOCATE 6, 20: PRINT USING "#########.##"; F1(2) + Md
   END IF
END SUB

SUB Iter.Start.Value

   IF NOCSN = 0 THEN
      Ybar = DE / 2
      AreaGr = PI# * (DE * DE - DI * DI) / 4
      Ast = .01 * AreaGr
   ELSE
      IF UnBi$ = "U" THEN
         Ybar = (CSYmax - CSYmin) / 2
         AlfaNE = 0
         Ast = .01 * AreaGr
      ELSE
        IF MxdOrg = 0 THEN AlfaNE = 0 ELSE AlfaNE = ATN(MydOrg / MxdOrg)
         AlfaNE = AlfaNE / PI# * 180
         Ybar = CSYmax - CSYmin - YgGros
         Ybar = Ybar + TAN(AlfaNE * PI# / 180) * (CSXmax - XgGros)
```

```
        Ast = .01 * AreaGr
      END IF
   END IF

END SUB

SUB Max.Strain

'
'      ****************************************************************
'      *            THIS SUBROUTINE OBTAIN MAXIMUM STRAIN.             *
'      ****************************************************************
'

   SELECT CASE Kind

      CASE 1
         EpsCu = .003

      CASE 2
         IF NOCSN = 0 THEN
            YIMIN = 1E+10
            FOR I = 1 TO NOSBAR
               YI = Rbar * COS((I - 1) * 2 * PI# / NOSBAR)
               IF YI < YIMIN THEN YIMIN = YI
            NEXT I
            HYIMIN = (.0035 / .0135) * (DE / 2 - YIMIN)
            IF Ybar <= HYIMIN THEN
               EpsCu = .01 * Ybar / (DE / 2 - YIMIN - Ybar)
            ELSE
               IF Ybar <= DE THEN
                  EpsCu = .0035
               ELSE
                  EpsCu = Ybar / (Ybar - 3 * DE / 7) * .002
               END IF
            END IF
         ELSE
            HNESM = 1E+10
            FOR I = 1 TO NOSBAR
               HNES = (XYS(I, 1) - Xp) * SIN(AlfaNE * PI# / 180)
               HNES = HNES + (XYS(I, 2) - Yp) * COS(AlfaNE * PI# / 180)
               IF HNES < HNESM THEN HNESM = HNES
            NEXT I
            IF HNEMIN > 0 THEN
               EpsCu = .014 * HNEMAX / (4 * HNEMAX + 3 * HNEMIN)
            ELSE
               IF HNEMAX < -.35 * HNESM THEN
                  EpsCu = -.01 * HNEMAX / HNESM
               ELSE
                  EpsCu = .0035
               END IF
            END IF
         END IF

      CASE 3
```

```
      EPS50U = (3 + .0285 * Fck) / (14.2 * Fck - 1000)
      Z = .5 / (EPS50U + Eps50h - .002)
      IF EpsCu = 0 THEN EpsCu = EPS50U

    CASE 4
      EpsCu = .0038
      eps0 = 2 * .85 * Fck / Ec

    CASE ELSE
      'no operation...

  END SELECT

  LOCATE 2, 71: PRINT USING "#.#####"; EpsCu

END SUB

SUB Pol.AreaMethod

' ***************  Subroutine for Area Method ***************************

  MNE = TAN(PI# * (1 - AlfaNE / 180))
  Xp = CSXmax
  Yp = CSYmax - Ybar

  FOR I = 1 TO NOCSN
    IF XYC(I, 1) = Xp THEN
      SELECT CASE XYC(I, 2)
        CASE IS > Yp
          XYC(I, 3) = 1
        CASE IS = Yp
          XYC(I, 3) = 2
        CASE ELSE
          XYC(I, 3) = 0
      END SELECT
    ELSE
      Mi = (XYC(I, 2) - Yp) / (XYC(I, 1) - Xp)
      SELECT CASE Mi
        CASE IS < MNE
          XYC(I, 3) = 1
        CASE IS = MNE
          XYC(I, 3) = 2
        CASE ELSE
          XYC(I, 3) = 0
      END SELECT
    END IF
  NEXT I
  XYC(NOCSN + 1, 3) = XYC(1, 3)
  NodNew = 0

  FOR I = 1 TO NOCSN
    IF XYC(I, 3) > 0 THEN 6180
    IF XYC(I + 1, 3) = 1 THEN 6230 ELSE 6330
6180
    NodNew = NodNew + 1
```

```
      XYCN(NodNew, 1) = XYC(I, 1)
      XYCN(NodNew, 2) = XYC(I, 2)
      IF XYC(I + 1, 3) = 2 THEN 6330
      IF XYC(I + 1, 3) <> 0 THEN 6330
6230
      IF XYC(I + 1, 1) = XYC(I, 1) THEN
      XCUT = XYC(I, 1)
      YCUT = MNE * (XCUT - Xp) + Yp
      GOTO 6300
      END IF
      MIJ = (XYC(I + 1, 2) - XYC(I, 2)) / (XYC(I + 1, 1) - XYC(I, 1))
      IF (MNE = 0) AND (MIJ = 0) THEN
         GOTO 6330
      ELSE
         IF MNE = 0 THEN
            YCUT = Yp
            XCUT = (Yp - XYC(I, 2) + MIJ * XYC(I, 1)) / MIJ
         ELSE
            IF MIJ = 0 THEN
               YCUT = XYC(I, 2)
               XCUT = (YCUT - Yp + MNE * Xp) / MNE
            ELSE
               XCUT=(-XYC(I,2)+MIJ*XYC(I,1)+Yp-MNE*Xp)/(-MNE+MIJ)
               YCUT = XYC(I, 2) + MIJ * (XCUT - XYC(I, 1))
            END IF
         END IF
      END IF
6300
      NodNew = NodNew + 1
      XYCN(NodNew, 1) = XCUT
      XYCN(NodNew, 2) = YCUT
6330
   NEXT I

   XYCN(NodNew + 1, 1) = XYCN(1, 1)
   XYCN(NodNew + 1, 2) = XYCN(1, 2)
   AreaCC = 0: SX = 0: SY = 0

   FOR I = 1 TO NodNew
      DELTAX = XYCN(I + 1, 1) - XYCN(I, 1)
      DELTAY = XYCN(I + 1, 2) - XYCN(I, 2)
      AreaCC = AreaCC + .5 * DELTAX * (XYCN(I + 1, 2) + XYCN(I, 2))
SX=SX+DELTAX*(XYCN(I,2)*(XYCN(I,2)+XYCN(I+1,2))+XYCN(I+1,2)*XYCN(I+1,2))/6
SY=SY-DELTAY*(XYCN(I,1)*(XYCN(I,1)+XYCN(I+1,1))+XYCN(I+1,1)*XYCN(I+1,1))/6
   NEXT I

   IF AreaCC = 0 THEN
      LOCATE 25, 1: PRINT SPACE$(80);
      LOCATE 25, 1: PRINT "AREACC=0";
      BEEP
   ELSE
      XGACC = SY / AreaCC
      YGACC = SX / AreaCC
      HAEMAX = -1E+10
```

```
      FOR I = 1 TO NOCSN
         HAE = (XYC(I, 1) - Xp) * SIN(AlfaNE * PI# / 180)
         HAE = HAE + (XYC(I, 2) - Yp) * COS(AlfaNE * PI# / 180)
         IF HAEMAX < HAE THEN
            HAEMAX = HAE
         END IF
      NEXT I

      NSIGS = 0
      MXSIGS = 0
      MYSIGS = 0

      FOR I = 1 TO NOSBAR
         HAE = (XYS(I, 1) - Xp) * SIN(AlfaNE * PI# / 180)
         HAE = HAE + (XYS(I, 2) - Yp) * COS(AlfaNE * PI# / 180)
         SIGS = EpsCu * Es * (1 - K1 * (1 - HAE / HAEMAX))
         IF SIGS > Fyd THEN SIGS = Fyd
         IF SIGS < -Fyd THEN SIGS = -Fyd
         NSIGS = NSIGS + XYS(I, 3) * SIGS
         MXSIGS = MXSIGS + XYS(I, 3) * SIGS * (XYS(I, 2) - YgGros)
         MYSIGS = MYSIGS + XYS(I, 3) * SIGS * (XYS(I, 1) - XgGros)
      NEXT I

      N = .85 * Fcd * AreaCC + Ast * NSIGS
      Mx = .85 * Fcd * AreaCC * (YGACC - YgGros) + Ast * MXSIGS
      My = .85 * Fcd * AreaCC * (XGACC - XgGros) + Ast * MYSIGS
   END IF
END SUB

SUB Pol.IntAreaMethod

' *********  Subroutine for Integration of area Method *****************

   MNE = TAN(PI# * (1 - AlfaNE / 180))
   Xp = CSXmax
   Yp = CSYmax - Ybar
   HNEMAX = -1E+10: HNEMIN = 1E+10

   FOR I = 1 TO NOCSN
      HNE = (XYC(I, 1) - Xp) * SIN(AlfaNE * PI# / 180)
      HNE = HNE + (XYC(I, 2) - Yp) * COS(AlfaNE * PI# / 180)
      IF HNEMAX < HNE THEN
         HNEMAX = HNE
      END IF
      IF HNEMIN > HNE THEN
         HNEMIN = HNE
      END IF
   NEXT I
Max.Strain
   Adim = dint / COS(AlfaNE / 180 * PI#)
   HDINT = dint / 2
   NC = 0: MXC = 0: MYC = 0
   Graph.Draw

   FOR YPint = (Yp + Adim / 2) TO CSYmax STEP Adim
```

```
      Strip.Length
      StrssStrn
      IntLine.Draw
      FDINT = dint * SumL * Stress
      NC = NC + FDINT
      MXC = MXC + (YL - YgGros) * FDINT
      MYC = MYC + (XL - XgGros) * FDINT
      HDINT = HDINT + dint
      IF HDINT > HNEMAX THEN
         YPint = CSYmax
      END IF
   NEXT YPint

   IF KindT$ = "Y" THEN
      HDINT = -dint / 2
      YINTEND = CSYmin - TAN(AlfaNE * PI# / 180) * (CSXmax - CSXmin)
      FOR YPint = (Yp - Adim / 2) TO YINTEND STEP -Adim
         Strip.Length
         StrssStrn
         IntLine.Draw
         FDINT = dint * SumL * Stress
         NC = NC + FDINT
         MXC = MXC + (YL - YgGros) * FDINT
         MYC = MYC + (XL - XgGros) * FDINT
         HDINT = HDINT - dint
      NEXT YPint
   END IF

   NSIGS = 0: MXSIGS = 0: MYSIGS = 0

   FOR I = 1 TO NOSBAR
      HNE = (XYS(I, 1) - Xp) * SIN(AlfaNE * PI# / 180)
      HNE = HNE + (XYS(I, 2) - Yp) * COS(AlfaNE * PI# / 180)
      SIGS = EpsCu * Es * HNE / HNEMAX
      IF SIGS > Fyd THEN SIGS = Fyd
      IF SIGS < -Fyd THEN SIGS = -Fyd
      NSIGS = NSIGS + XYS(I, 3) * SIGS
      MXSIGS = MXSIGS + XYS(I, 3) * SIGS * (XYS(I, 2) - YgGros)
      MYSIGS = MYSIGS + XYS(I, 3) * SIGS * (XYS(I, 1) - XgGros)
   NEXT I

   N = NC + Ast * NSIGS
   Mx = MXC + Ast * MXSIGS
   My = MYC + Ast * MYSIGS
END SUB

SUB Strip.Length
'
'       *****************************************************************
'       *  THIS SUBROUTINE CALCULATES THE LENGTH OF THE LINE THAT      *
'       *             CROSS ANY ARBITRARY CROSS SECTION.               *
'       *****************************************************************
'

   FOR I = 1 TO NOCSN
```

```
      IF XYC(I, 1) = Xp THEN

         SELECT CASE XYC(I, 2)
            CASE IS > YPint
               XYC(I, 3) = 1
            CASE IS = YPint
               XYC(I, 3) = 0
            CASE ELSE
               XYC(I, 3) = -1
         END SELECT
      ELSE
         Mi = (XYC(I, 2) - YPint) / (XYC(I, 1) - Xp)
         SELECT CASE Mi
            CASE IS < MNE
               XYC(I, 3) = 1
            CASE IS = MNE
               XYC(I, 3) = 0
            CASE ELSE
               XYC(I, 3) = -1
         END SELECT

      END IF
   NEXT I

   XYC(NOCSN + 1, 3) = XYC(1, 3)
   NodNew = 0
   XYC(0, 1) = XYC(NOCSN, 1)
   XYC(0, 2) = XYC(NOCSN, 2)
   XYC(0, 3) = XYC(NOCSN, 3)

   FOR I = 1 TO NOCSN
      IF (XYC(I, 3) = -1) AND (XYC(I + 1, 3) = 1) THEN GOTO Skip1
      IF (XYC(I, 3) = 1) AND (XYC(I + 1, 3) = -1) THEN GOTO Skip1
      IF XYC(I, 3) <> 0 THEN GOTO Skip2
      IF XYC(I - 1, 3) = XYC(I + 1, 3) THEN GOTO Skip2
      NodNew = NodNew + 1
      XYCN(NodNew, 1) = XYC(I, 1)
      XYCN(NodNew, 2) = XYC(I, 2)
      GOTO Skip2

Skip1:
      IF XYC(I + 1, 1) = XYC(I, 1) THEN
         XCUT = XYC(I, 1)
         YCUT = MNE * (XCUT - Xp) + YPint
      ELSE
         MIJ = (XYC(I + 1, 2) - XYC(I, 2)) / (XYC(I + 1, 1) - XYC(I, 1))
         IF AlfaNE = 0 THEN
            YCUT = YPint
            XCUT = (YPint - XYC(I, 2) + MIJ * XYC(I, 1)) / MIJ
         ELSE
            IF MIJ = 0 THEN
               YCUT = XYC(I, 2)
               XCUT = (YCUT - YPint + MNE * Xp) / MNE
            ELSE
XCUT = (-XYC(I, 2) + MIJ * XYC(I, 1) + YPint - MNE * Xp) / (-MNE + MIJ)
```

```
                YCUT = XYC(I, 2) + MIJ * (XCUT - XYC(I, 1))
            END IF
         END IF
      END IF

      NodNew = NodNew + 1
      XYCN(NodNew, 1) = XCUT
      XYCN(NodNew, 2) = YCUT

Skip2:
   NEXT I

   IF NodNew < 2 THEN EXIT SUB

   IF NodNew <= 2 THEN
      IF XYCN(2, 1) < XYCN(1, 1) THEN

         CHANGE = XYCN(2, 1)
         XYCN(2, 1) = XYCN(1, 1)
         XYCN(1, 1) = CHANGE

         CHANGE = XYCN(2, 2)
         XYCN(2, 2) = XYCN(1, 2)
         XYCN(1, 2) = CHANGE

      END IF
   ELSE
      FOR I = 1 TO NodNew - 1
         FOR J = I + 1 TO NodNew
            IF XYCN(J, 1) <= XYCN(I, 1) THEN

               CHANGE = XYCN(J, 1)
               XYCN(J, 1) = XYCN(I, 1)
               XYCN(I, 1) = CHANGE

               CHANGE = XYCN(J, 2)
               XYCN(J, 2) = XYCN(I, 2)
               XYCN(I, 2) = CHANGE

            END IF
         NEXT J
      NEXT I
   END IF

   SUMDX = 0: SUMDY = 0: SumL = 0: XSUM = 0: YSUM = 0

   FOR I = 2 TO NodNew + 1 STEP 2

      dx = XYCN(I, 1) - XYCN(I - 1, 1)
      dy = XYCN(I - 1, 2) - XYCN(I, 2)

      SumL = SumL + SQR(dx * dx + dy * dy)
      SUMDX = SUMDX + dx
      XSUM = dx * (XYCN(I, 1) - dx / 2) + XSUM
      SUMDY = SUMDY + dy
```

```
      YSUM = dy * (XYCN(I - 1, 2) - dy / 2) + YSUM

   NEXT I

   IF SUMDX <> 0 THEN
      XL = XSUM / SUMDX
   ELSE
      XL = 0
   END IF

   IF SUMDY <> 0 THEN
      YL = YSUM / SUMDY
   ELSE
      YL = YPint
   END IF

END SUB

SUB StrssStrn

' ***********************************************************************
' * Subroutines of Stress-Strain Relation in integration of area method *
' ***********************************************************************

Stress.Strain:

   Strain = EpsCu / HNEMAX * HDINT

   IF Strain <= 0 THEN

      IF Strain < -.0002 THEN
         YPint = YINTEND
         Stress = 0
      ELSE
         IF Strain > -.0001 THEN
            Stress = Strain / .0001 * Fctd
         ELSE
            Stress = -Fctd * (1.5 + Strain / .0002)
         END IF
      END IF

   ELSE

      SELECT CASE Kind
         CASE 1
            IF Strain < (1 - K1) * EpsCu THEN
               Stress = 0
            ELSE
               Stress = Peak * Fcd
            END IF

         CASE 2
            IF Strain < .002 THEN
               Stress = Peak * Fcd * (2 * Strain / .002 - (Strain / .002)
            ELSE
```

```
               Stress = Peak * Fcd
            END IF

         CASE 3
            IF Strain < .002 THEN
               Stress = Peak * Fcd * (2 * Strain / .002 - (Strain / .002)
            ELSE
               Stress = Peak * Fcd * (1 - Z * (Strain - .002))
            END IF

         CASE 4
            IF Strain < eps0 THEN
               Stress = Peak * Fcd * (2 * Strain / eps0 - (Strain / eps0)
            ELSE
               Stress = Peak * Fcd
 Stress = Stress - .15 * Peak * Fcd * (Strain - eps0) / (.0038 - eps0)
            END IF

         CASE ELSE
            'no operation...

      END SELECT

   END IF

   XStrain = Referx1 + (ReferX2 - Referx1) * Strain / EpsCu
   YStrain = ReferY1 + (ReferY2 - ReferY1) * Stress / Fcd
   LINE (XStrain, ReferY1)-(XStrain, YStrain)
END SUB

SUB Uniaxial

   IF Kind = 0 THEN
      Pol.AreaMethod
   ELSE
      Pol.IntAreaMethod
   END IF

   x1(1) = Ybar: x1(2) = Ast
   F1(1) = N: F1(2) = Mx
   Ybar = Ybar + dYbar

   IF Kind = 0 THEN
      Pol.AreaMethod
   ELSE
      Pol.IntAreaMethod
   END IF

   d(1, 1) = (N - F1(1)) / dYbar
   d(2, 1) = (Mx - F1(2)) / dYbar
   Ybar = Ybar - dYbar

   Ast = Ast + dAst
   IF Kind = 0 THEN
      Pol.AreaMethod
```

```
   ELSE
      Pol.IntAreaMethod
   END IF

   d(1, 2) = (N - F1(1)) / dAst
   d(2, 2) = (Mx - F1(2)) / dAst
   Ast = Ast - dAst

   d = d(1, 1) * d(2, 2) - d(2, 1) * d(1, 2)

   IF d = 0 THEN
      LOCATE 25, 1: PRINT SPACE$(80);
      LOCATE 25, 1: PRINT "DET=0 <Strike Any Key>";
      BEEP
      a$ = ""
      DO
      a$ = INKEY$
      LOOP WHILE a$ = ""
   ELSE
      B(1, 1) = d(2, 2) / d
      B(1, 2) = -d(1, 2) / d
      B(2, 1) = -d(2, 1) / d
      B(2, 2) = d(1, 1) / d
      x2(1) = 0
      x2(2) = 0
      Criter = 0
      F1(1) = F1(1) - Nd
      F1(2) = F1(2) - Mxd

      FOR I = 1 TO 2
         FOR J = 1 TO 2
            d(I, J) = -B(I, J)
         NEXT J
      NEXT I

      FOR I = 1 TO 2
         FOR J = 1 TO 2
             x2(I) = x2(I) + d(I, J) * F1(J)
         NEXT J
         Criter = Criter + x2(I) ^ 2
         x2(I) = x1(I) + x2(I)
      NEXT I

      LOCATE 2, 27: PRINT Iter
      Iter = Iter + 1

      LOCATE 7, 22: PRINT USING "####.#####"; x2(1)
      LOCATE 9, 22: PRINT USING "####.#####"; x2(2)

      LOCATE 3, 20: PRINT USING "#########.##"; F1(1) + Nd
      LOCATE 4, 20: PRINT USING "#########.##"; F1(2) + Mxd
   END IF
END SUB
```

B.3. RCSDCTRL

```
DECLARE SUB Arbit.Sec.Proper ()
DECLARE SUB Arbit.Sec.Window ()
DECLARE SUB Graph.Draw ()
DECLARE SUB Descriptions (DESCRIP!)
DECLARE SUB Table.Layout ()
' $INCLUDE: 'RCSDCOM.BI'

SUB Ast.ACI

'**** control Ast before saving according to ACI restrictions ***

   IF Ast < .01 * AreaGr THEN
      AstOK = .01 * AreaGr
      AST1MSG$ = "Ast FOUND TO BE LESS THAN MINIMUM VALUE."
      BEEP
      LOCATE 25, 1: PRINT " Ast < .01 Ac ,Ast SHOULD be at least Ast=";
      PRINT AstOK; " (ACI)";
      a$ = ""
      DO
      a$ = INKEY$
      LOOP WHILE a$ = ""
      LOCATE 25, 1: PRINT SPACE$(80);
   END IF

   IF Ast > .08 * AreaGr THEN
      AST1MSG$ = "Ast FOUND TO BE GREATHER THAN MAXIMUM VALUE."
      BEEP
      LOCATE 25, 1: PRINT "Ast > .08 Ac (ACI)."
      PRINT "IGNOR THIS CONDITION AND CONTINUE ? (Y/N)"
      DO
         AstMax$ = UCASE$(INKEY$)
      LOOP UNTIL AstMax$ = "Y" OR AstMax$ = "N"
   END IF

END SUB

SUB Ast.TS500

'**** control Ast before saving according to TS500 restrictions ***

   IF Ast < .01 * AreaGr THEN
      AstOK = .01 * AreaGr
      AST1MSG$ = "Ast FOUND TO BE LESS THAN MINIMUM VALUE."
      BEEP
      LOCATE 25, 1: PRINT " Ast < .01 Ac ,Ast SHOULD be at least Ast=";
      PRINT AstOK; " (TS-500)";
      a$ = ""
      DO
      a$ = INKEY$
      LOOP WHILE a$ = ""
      LOCATE 25, 1: PRINT SPACE$(80);
   END IF
```

```
   IF Ast > .04 * AreaGr THEN
      AST1MSG$ = "Ast FOUND TO BE GREATHER THAN MAXIMUM VALUE."
      BEEP
      LOCATE 25, 1: PRINT "Ast > .04 Ac (TS500).";
      PRINT "IGNOR THIS CONDITION AND CONTINUE ? (Y/N)";
      DO
      AstMax$ = UCASE$(INKEY$)
      LOOP UNTIL AstMax$ = "Y" OR AstMax$ = "N"
      LOCATE 25, 1: PRINT SPACE$(80);
   END IF

END SUB

SUB Axial.Load.TS500

   N0max = .6 * Fck * AreaGr

   IF Nd > N0max THEN
      LOCATE 25, 1: PRINT "Nd > .6 Fck Ac :";
      PRINT " SECTION IS UNSAFE.(TS500)";
      BEEP
      PRINT " IGNOR THIS CONDITION ? (Y/N)";
      DO
         NdIgnor$ = UCASE$(INKEY$)
      LOOP UNTIL NdIgnor$ = "Y" OR NdIgnor$ = "N"
      LOCATE 25, 1: PRINT SPACE$(80);
   END IF

END SUB

SUB Control.AlfaNE (Cont$)

   IF AlfaNE >= 90 THEN
      LOCATE 25, 1: PRINT "ALFANE CAN'T BE >= 90 DEG.";
      Cont$ = "N"
   END IF
   IF AlfaNE < 0 THEN AlfaNE = 0

END SUB

SUB Control.Min.Ecc

'      ***************************************************************
'      *         THIS SUBROUTINE CONTROL MINIMUM ECCENTRICITY.       *
'      ***************************************************************

   IF NOCSN > 0 THEN GOTO Arbitrary.Section
   EMIN = .1 * DE
   IF EMIN < EccMin THEN EMIN = EccMin

   IF MxdOrg >= EMIN * Nd THEN GOTO My.Control
   LOCATE 25, 1: PRINT "MIN. Ecc. IN Y-DIRECTION NOT SATISFIED.";
   BEEP
   MxdOrg = EMIN * Nd
```

```
   PRINT " Mxd CHANGED TO "; MxdOrg;
   LOCATE 4, 7: PRINT USING "#########.##"; MxdOrg
My.Control:
   IF MydOrg >= EMIN * Nd THEN GOTO Line.End
   LOCATE 25, 1: PRINT "MIN. Ecc. IN X-DIRECTION NOT SATISFIED.";
   BEEP
   MydOrg = EMIN * Nd
   PRINT " Myd CHANGED TO "; MydOrg;
   LOCATE 5, 7: PRINT USING "#########.##"; MydOrg
   GOTO Line.End

Arbitrary.Section:

   EXMIN = .1 * (CSXmax - CSXmin)
   EYMIN = .1 * (CSYmax - CSYmin)
   IF EXMIN < EccMin THEN EXMIN = EccMin
   IF EYMIN < EccMin THEN EYMIN = EccMin

   IF MxdOrg > EYMIN * Nd THEN
      GOTO Other.Controls
   ELSE
      MxdOrg = EYMIN * Nd
   END IF

   LOCATE 25, 1: PRINT "MIN. Ecc. IN Y-DIRECTION NOT SATISFIED.";
   BEEP
   PRINT " Mxd CHANGED TO "; MxdOrg;
   LOCATE 4, 7: PRINT USING "#########.##"; MxdOrg

Other.Controls:

   IF MydOrg > EXMIN * Nd THEN
      GOTO Line.End
   ELSE
      MydOrg = EXMIN * Nd
   END IF
   LOCATE 25, 1: PRINT "MIN. Ecc. IN X-DIRECTION NOT SATISFIED.";
   BEEP
   PRINT " Myd CHANGED TO "; MydOrg;
   LOCATE 5, 7: PRINT USING "#########.##"; MydOrg

Line.End:
   Mxd = MxdOrg
   Myd = MydOrg

END SUB

SUB Control.Ybar (Cont$)

   IF Ybar <= 0 THEN
      LOCATE 25, 1: PRINT "YBAR CAN'T BE LESS OR EQUAL TO ZERO.";
      Cont$ = "N"
   END IF

END SUB
```

```
SUB RotSec.Control

'     *************************************************************
'     *   THIS PART OF PROGRAM ROTATES SECTION RELATED TO THE SIGN  *
'     *       OF MOMENTS,TO HAVE A MOMENT WITH A POSITIVE SIGN.     *
'     *************************************************************

   DIM Rot(2, 2)

   IF Mxd < 0 THEN GOTO 1120

   IF Mxd = 0 THEN GOTO 1060

   IF Myd >= 0 THEN
      Rot$ = "0 DEG."
      GOTO PrntRot
   END IF

1030
   Rot(1, 1) = 0: Rot(1, 2) = -1
   Rot(2, 1) = 1: Rot(2, 2) = 0
   Rot$ = "-90 DEG."
   CHANGE = Mxd: Mxd = -Myd: Myd = CHANGE
   GOTO 1160

1060
   IF Myd > 0 THEN
      GOTO 1090
   END IF
   IF Myd < 0 THEN
      GOTO 1030
   END IF
   GOTO PrntRot

1090
   Rot(1, 1) = 0: Rot(1, 2) = 1
   Rot(2, 1) = -1: Rot(2, 2) = 0
   Rot$ = "+90 DEG."
   CHANGE = Mxd: Mxd = Myd: Myd = -CHANGE
   GOTO 1160

1120
   IF Myd <= 0 THEN
      GOTO 1140
   END IF

   GOTO 1090


1140 Rot(1, 1) = -1: Rot(1, 2) = 0
     Rot(2, 1) = 0: Rot(2, 2) = -1
     Rot$ = "180 DEG."
     Myd = -Myd: Mxd = -Mxd
```

```
1160
   FOR i = 1 TO NOCSN
      old = XYC(i, 1)
      XYC(i, 1) = old * Rot(1, 1) + XYC(i, 2) * Rot(2, 1)
      XYC(i, 2) = old * Rot(1, 2) + XYC(i, 2) * Rot(2, 2)
   NEXT i

   FOR i = 1 TO NOSBAR
      old = XYS(i, 1)
      XYS(i, 1) = old * Rot(1, 1) + XYS(i, 2) * Rot(2, 1)
      XYS(i, 2) = old * Rot(1, 2) + XYS(i, 2) * Rot(2, 2)
   NEXT i

   XYC(NOCSN + 1, 1) = XYC(1, 1): XYC(NOCSN + 1, 2) = XYC(1, 2)

PrntRot:

   LOCATE 25, 1: PRINT "SECTION ROTATED "; Rot$;

   IF Myd <= Mxd THEN
      GOTO End.of.Sub
   END IF
   a$ = ""
   LOCATE 25, 1: PRINT "SECTION ROTATED "; Rot$; "<Strike Any Key>";
   DO
   a$ = INKEY$
   LOOP WHILE a$ = ""
LOCATE 25, 1: PRINT "It is suggested to reflect the section (Myd>Mxd).";
   PRINT "Do you want to reflect <Y/N>?";

   DO
   REFL$ = UCASE$(INKEY$)
   LOOP UNTIL REFL$ = "Y" OR REFL$ = "N"
   LOCATE 25, 1: PRINT SPACE$(80);
   IF REFL$ = "N" THEN GOTO End.of.Sub

   Rot(1, 1) = 0: Rot(1, 2) = 1: Rot(2, 1) = 1: Rot(2, 2) = 0

   CHANGE = Myd: Myd = Mxd: Mxd = CHANGE

   FOR i = 1 TO NOCSN
      old = XYC(i, 1)
      XYC(i, 1) = old * Rot(1, 1) + XYC(i, 2) * Rot(2, 1)
      XYC(i, 2) = old * Rot(1, 2) + XYC(i, 2) * Rot(2, 2)
   NEXT i

   FOR i = 1 TO NOSBAR
     old = XYS(i, 1)
     XYS(i, 1) = old * Rot(1, 1) + XYS(i, 2) * Rot(2, 1)
     XYS(i, 2) = old * Rot(1, 2) + XYS(i, 2) * Rot(2, 2)
   NEXT i

   LOCATE 25, 1: PRINT "Section reflected. Mxd and Myd swaped!";

   DIM CONCHANG(NOCSN + 1, 2), STCHANG(NOSBAR + 1, 3)
```

```
    FOR i = 1 TO NOCSN
       CONCHANG(i, 1) = XYC(i, 1)
       CONCHANG(i, 2) = XYC(i, 2)
    NEXT i

    FOR i = 1 TO NOCSN
       XYC(i, 1) = CONCHANG(NOCSN + 1 - i, 1)
       XYC(i, 2) = CONCHANG(NOCSN + 1 - i, 2)
    NEXT i

    FOR i = 1 TO NOSBAR
       STCHANG(i, 1) = XYS(i, 1)
       STCHANG(i, 2) = XYS(i, 2)
       STCHANG(i, 3) = XYS(i, 3)
    NEXT i

    FOR i = 1 TO NOSBAR
       XYS(i, 1) = STCHANG(NOSBAR + 1 - i, 1)
       XYS(i, 2) = STCHANG(NOSBAR + 1 - i, 2)
       XYS(i, 3) = STCHANG(NOSBAR + 1 - i, 3)
    NEXT i
End.of.Sub:
    IF Rot$ <> "0 DEG." OR REFL$ = "Y" THEN
       Arbit.Sec.Proper
       Arbit.Sec.Window
       WINDOW (Winx1, Winy1)-(Winx2, Winy2)
       Graph.Draw
    END IF

END SUB

????
```

B.4. RCSDGRPH

```
DECLARE SUB StrssStrn ()
DECLARE SUB Stress.Axis ()
' $INCLUDE: 'RCSDCOM.BI'

SUB Arbit.Sec.Window
   IF NOCSN <> 0 THEN
      IF (CSXmax - CSXmin) < (CSYmax - CSYmin) THEN
         CSMAX = CSYmax - CSYmin
      ELSE
         CSMAX = CSXmax - CSXmin
      END IF
      MIDXC = (CSXmax + CSXmin) / 2
      MIDYC = (CSYmax + CSYmin) / 2
      Winx1 = MIDXC - CSMAX / 2 / AspRatio
      Winx2 = MIDXC + CSMAX / 2 / AspRatio
      Winy1 = MIDYC - CSMAX / 2
      Winy2 = MIDYC + CSMAX / 2
   ELSE
      Winx1 = -DE / 1.4
      Winy1 = -DE / 2
      Winx2 = DE / 1.4
      Winy2 = DE / 2
   END IF
   ' ** Review for corner (using in Solver.table)
   Winx1 = Winx1 - (Winx2 - Winx1)
   Winy2 = Winy2 + (Winy2 - Winy1)
   Winy1 = Winy1 - (Winy2 - Winy1) * .1
   Referx1 = (Winx2 + Winx1) / 2 + (Winx2 - Winx1) * .07
   ReferY1 = Winy1 + (Winy2 - Winy1) * 17 / 25
   ReferX2 = Winx2 - (Winx2 - Winx1) * 5 / 80
   ReferY2 = Winy2 - (Winy2 - Winy1) * 3 / 25
END SUB
SUB Graph.Draw
'       **************************************************************
'       *            THIS SUBROUTINE DRAW THE GRAPHICS.              *
'       **************************************************************
   col% = 43
   FOR ROW% = 3 TO 9
      LOCATE ROW%, col%: PRINT SPACE$(37);
   NEXT ROW%
   FOR ROW% = 12 TO 23
      LOCATE ROW%, col%: PRINT SPACE$(37);
   NEXT ROW%
   Stress.Axis
   IF NOCSN = 0 THEN
      CIRCLE (0, 0), DE / 2, , , , AspRatio
      CIRCLE (0, 0), DI / 2, , , , AspRatio
      FOR i = 1 TO NOSBAR
         YI = Rbar * COS((i - 1) * 2 * PI# / NOSBAR)
         Xi = Rbar * SIN((i - 1) * 2 * PI# / NOSBAR)
         CIRCLE (Xi, YI), DE / 100
      NEXT i
   ELSE
```

```
   FOR i = 1 TO NOCSN
      LINE (XYC(i, 1), XYC(i, 2))-(XYC(i + 1, 1), XYC(i + 1, 2))
   NEXT i
   FOR i = 1 TO NOSBAR
      CIRCLE (XYS(i, 1), XYS(i, 2)), (CSXmax - CSXmin) / 150
   NEXT i
   END IF
END SUB
SUB GrphScr
        SELECT CASE Graphcard%
          CASE 1
               'PRINT "It is a plain old monochrome adapter"
               PRINT "Sorry! This graphic card did not assign."
          CASE 2
               'PRINT " It is a Hercules card."
               AspRatio = .6296296
               SCREEN 3
          CASE 3
       'PRINT " It is a plain old CGA adapter or an EGA emulating a CGA"
               AspRatio = .4166667
               SCREEN 2
          CASE 4
               'PRINT "It is an EGA card with a monochrome monitor,"
               'PRINT "And it has"; Memory; "K of memory on board"
               AspRatio = .7291667
              SCREEN 9
          CASE 5
               'PRINT "It is an EGA card with a color monitor,"
               'PRINT "And it has"; Memory; "K of memory on board"
               AspRatio = .7291667
               SCREEN 9
          CASE 6
               'PRINT "It is a VGA adapter with a monochrome monitor"
               AspRatio = .7291667
               SCREEN 9
          CASE 7
               'PRINT "It is a VGA adapter with a color monitor"
               AspRatio = .7291667
               SCREEN 9
        END SELECT
END SUB
SUB IntLine.Draw
'                    ********************************
'                    * LINE DRAWING FOR INTEGRATION *
'                    ********************************

   FOR i = 2 TO NodNew + 1 STEP 2
      LINE (XYCN(i - 1, 1), XYCN(i - 1, 2))-(XYCN(i, 1), XYCN(i, 2))
   NEXT i
END SUB
SUB Rec.Stress.Draw
    col% = 43
   FOR ROW% = 3 TO 9
      LOCATE ROW%, col%: PRINT SPACE$(37);
   NEXT ROW%
```

```
   Stress.Axis
   dxref = ReferX2 - Referx1
   dyref = ReferY2 - ReferY1
   LINE (Referx1 + (1 - K1) * dxref, ReferY2)-(ReferX2, ReferY2)
   FOR i = 0 TO 20
   x = Referx1 + (1 - K1) * dxref + K1 * dxref * i / 20
   LINE (x, ReferY1)-(x, ReferY2)
   NEXT i
   LOCATE 2, 71: PRINT USING "#.#####"; EpsCu
END SUB
SUB Stress.Axis
   dxref = ReferX2 - Referx1
   dyref = ReferY2 - referty1
   LINE (Referx1 - .07 * dxref, ReferY1)-(ReferX2 + .05 * dxref, ReferY1)
   LINE (ReferX2+.05*dxref,ReferY1)-(ReferX2+.03*dxref,ReferY1+.01*dyref)
   LINE (ReferX2+.05*dxref,ReferY1)-(ReferX2+.03*dxref,ReferY1-.01*dyref)
   LINE (Referx1,ReferY1-.02*dyref)-(Referx1,ReferY2+.02*dyref)
   LINE (Referx1,ReferY2+.02*dyref)-(Referx1-.01*dxref,ReferY2-.01*dyref)
   LINE (Referx1,ReferY2+.02*dyref)-(Referx1+.01*dxref,ReferY2-.01*dyref)
   LOCATE 3, 43: PRINT "r"
   LOCATE 9, 79: PRINT "E"
END SUB
SUB Stress.Draw
   col% = 43
   FOR ROW% = 3 TO 9
      LOCATE ROW%, col%: PRINT SPACE$(37);
   NEXT ROW%
   Stress.Axis
   dxref = ReferX2 - Referx1
   dyref = ReferY2 - ReferY1
   HNEMAX = 40
   FOR HDINT = 1 TO 40
      StrssStrn
   NEXT HDINT
   LOCATE 2, 71: PRINT USING "#.#####"; EpsCu
END SUB
```

RCSDINPT

```
DECLARE SUB Rec.Stress.Draw ()
DECLARE SUB Stress.Draw ()
DECLARE SUB Arbit.Sec.Proper ()
DECLARE SUB Control.Min.Ecc ()
DECLARE SUB Arbit.Sec.Window ()
DECLARE SUB Graph.Draw ()
DECLARE SUB Numin (inn#, Max.Digits%, Max.Places%, Exit.Code%
                  , textinbg%, textinfg%)
DECLARE SUB Textin (T$, Max.Length%, Caps.On%, Exit.Code%
                  , textinbg%, textinfg%)
DECLARE SUB Strip.FName (FileName$, Extension$)
DECLARE SUB Descriptions (DESCRIP!)
DECLARE SUB Table.Layout ()
DECLARE SUB Yes.No (Message$, Answer$)
' $INCLUDE: 'RCSDCOM.BI'

SUB Input.Loading
   Table.Layout
   Descriptions 10
   LOCATE 5, 5: PRINT "Input the design loads :"
   tbg% = 0
   tfg% = 7
   InpNdOrg# = NdOrg
   InpMxdOrg# = MxdOrg
   InpMydOrg# = MydOrg
   LOCATE 8, 5: PRINT "Nd =";
   PRINT USING "#######.##"; NdOrg
   LOCATE 10, 5: PRINT "Mxd=";
   PRINT USING "#######.##"; MxdOrg
   LOCATE 12, 5: PRINT "Myd=";
   PRINT USING "#######.##"; MydOrg

Get.Nd:
   LOCATE 8, 9
   Numin InpNdOrg#, 8, 2, Exit.Code%, tbg%, tfg%
   NdOrg = InpNdOrg#
   IF Exit.Code% = -72 THEN GOTO Get.Myd
   IF Exit.Code% = -80 OR Exit.Code% = 0 THEN GOTO Get.Mxd
   GOTO End.Get.Load

Get.Mxd:
   LOCATE 10, 9
   Numin InpMxdOrg#, 8, 2, Exit.Code%, tbg%, tfg%
   MxdOrg = InpMxdOrg#
   IF Exit.Code% = -72 THEN GOTO Get.Nd
   IF Exit.Code% = -80 OR Exit.Code% = 0 THEN GOTO Get.Myd
   GOTO End.Get.Load

Get.Myd:
   LOCATE 12, 9
   Numin InpMydOrg#, 8, 2, Exit.Code%, tbg%, tfg%
   MydOrg = InpMydOrg#
   IF Exit.Code% = -72 THEN GOTO Get.Mxd
```

```
   IF Exit.Code% = -80 OR Exit.Code% = 0 THEN GOTO Get.Nd

End.Get.Load:
   Nd = NdOrg
   Mxd = MxdOrg
   Myd = MydOrg
END SUB

SUB Input.Material
      Table.Layout
      Descriptions 8
      LOCATE 5, 5: PRINT "Enter Material file name :"
      LOCATE 8, 5: PRINT "Material data file name =";
      tbg% = 0
      tfg% = 7

text.inp:
      LOCATE 8, 30
      Textin DataFile$, 8, 1, Exit.Code%, tbg%, tfg%
      Strip.FName DataFile$, Extension$
      IF DataFile$ = "" THEN BEEP: GOTO text.inp
      MatFile$ = DataFile$ + ".MAT"
      Funit% = FREEFILE
      OPEN "I", #Funit%, MatFile$
      INPUT #Funit%, K1, Ec, Es, FcdOrg, Fck, Fctd, FydOrg, Fyk
      CLOSE #Funit%
      LOCATE 10, 20: PRINT "K1="; K1
      LOCATE 11, 20: PRINT "Ec="; Ec
      LOCATE 12, 20: PRINT "Es="; Es
      LOCATE 13, 20: PRINT "Fcd="; FcdOrg
      LOCATE 14, 20: PRINT "Fck="; Fck
      LOCATE 15, 20: PRINT "Fctd="; Fctd
      LOCATE 16, 20: PRINT "Fyd="; FydOrg
      LOCATE 17, 20: PRINT "Fyk="; Fyk
      IF Exit.Code% = -73 OR Exit.Code% = -81 OR Exit.Code% = 2 THEN
         'Nothing
      ELSE
         GOTO text.inp
      END IF
END SUB

SUB Input.Section

start.input:
      Table.Layout
      Descriptions 9
      LOCATE 8, 27: PRINT "                    "
      LOCATE 5, 6: PRINT "Input section file :"
      LOCATE 8, 6: PRINT "Section: "
      LOCATE 8, 15
      tbg% = 0: tfg% = 7
      Textin Section.Fname$, 8, 1, Exit.Code%, tbg%, tfg%
      IF Section.Fname$ = "" THEN BEEP: GOTO start.input
      ltext = LEN(Section.Fname$)
      textkind$ = "Text"
```

```
      IF ASC(MID$(Section.Fname$, 1, 1)) < 65 THEN
         FOR i = 1 TO ltext
     IF ASC(MID$(Section.Fname$, i, 1)) < 65 THEN textkind$ = "Number"
         NEXT i
         IF textkind$ = "Number" THEN
            secnum = VAL(Section.Fname$)
            IF secnum > -1 AND secnum < 1000 THEN
               SELECT CASE ltext
                  CASE IS = 1
                     Sec$ = "SECTION.00" + Section.Fname$
                  CASE IS = 2
                     Sec$ = "SECTION.0" + Section.Fname$
                  CASE IS = 3
                     Sec$ = "SECTION." + Section.Fname$
                  CASE ELSE
                     GOTO start.input
                  END SELECT
               GOTO start.Read
            ELSE
               GOTO start.input
            END IF
         END IF
      ELSE
         Sec$ = Section.Fname$ + ".SEC"
      END IF
start.Read:
   Funit% = FREEFILE
   OPEN "I", #Funit%, Sec$
   INPUT #Funit%, NOCSN, NOSBAR
   ERASE XYC, XYCN, XYS
   DIM XYC(NOCSN + 1, 3), XYCN(NOCSN * 2, 2), XYS(NOSBAR, 3)
   IF NOCSN = 0 THEN
      INPUT #Funit%, DE, DI, Rbar
      FOR i = 1 TO NOSBAR
         INPUT #Funit%, XYS(i, 3)
      NEXT i
   ELSE
      FOR i = 1 TO NOCSN
         INPUT #Funit%, XYC(i, 1), XYC(i, 2)
      NEXT i
      XYC(NOCSN + 1, 1) = XYC(1, 1)
      XYC(NOCSN + 1, 2) = XYC(1, 2)
   FOR i = 1 TO NOSBAR
         INPUT #Funit%, XYS(i, 1), XYS(i, 2), XYS(i, 3)
      NEXT i
   END IF
   CLOSE #Funit%
   SUMRAT = 0
   FOR i = 1 TO NOSBAR
      SUMRAT = SUMRAT + XYS(i, 3)
   NEXT i
   FOR i = 1 TO NOSBAR
      XYS(i, 3) = XYS(i, 3) / SUMRAT
   NEXT i
END SUB
```

```
SUB Kent.Park.Pmeter
   ' *****  For Kent & park Input the Strain paremeters *****
   tbg% = 0
   tfg% = 7
   Table.Layout
   Descriptions 7
   LOCATE 5, 3: PRINT "Enter the strain parameters :"
   LOCATE 8, 3: PRINT "Ecu  = <Print zero for  E50U>";
   LOCATE 10, 3: PRINT "E50h = <Press Enter for zero> ";
   PRINT USING "#.######"; Eps50h

Get.Epscu:
   LOCATE 8, 33
   InpEpsCu# = EpsCu
   Numin InpEpsCu#, 1, 6, Exit.Code%, tbg%, tfg%
   EpsCu = InpEpsCu#
IF Exit.Code%=-72 OR Exit.Code%=-80 OR Exit.Code%=0 THEN GOTO Get.Eps50h
   GOTO End.Get.Kent.Park:

Get.Eps50h:
   LOCATE 10, 33
   InpEps50h# = Eps50h
   Numin InpEps50h#, 1, 6, Exit.Code%, tbg%, tfg%
   Eps50h = InpEps50h#
IF Exit.Code%=-72 OR Exit.Code%=-80 OR Exit.Code%=0 THEN GOTO Get.Epscu

End.Get.Kent.Park:

END SUB

SUB Kind.of.Solv
   Table.Layout
   Descriptions 3
   LOCATE 3, 5: PRINT "ENTER THE TYPE OF SOLUTION."
   GOSUB sub1
   GOSUB sub2
   GOSUB sub3
   GOSUB sub4
   GOSUB sub5
   GOSUB sub6
   IF ArMethCode$ = "TS-500" AND Kind = 0 THEN line.no% = 1
   IF ArMethCode$ = "ACI" AND Kind = 0 THEN line.no% = 2
   IF Kind = 1 THEN line.no% = 3
   IF Kind = 2 THEN line.no% = 4
   IF Kind = 3 THEN line.no% = 5
   IF Kind = 4 THEN line.no% = 6
   IF line.no% = 0 THEN line.no% = 1
   Exit.Code% = 1
   COLOR 0, 7
   GOSUB subselect
   COLOR 7, 0
   DO
      LOCATE 3, 3
      Textin T$, 1, Caps.On%, Exit.Code%, -1, -1
      IF Exit.Code% = -72 THEN
```

```
         COLOR 7, 0
         GOSUB subselect
         COLOR 0, 7
         line.no% = line.no% - 1
         IF line.no% < 1 THEN line.no% = 6
         COLOR 0, 7
         GOSUB subselect
         COLOR 7, 0
      ELSE
         IF Exit.Code% = -80 THEN
            COLOR 7, 0
            GOSUB subselect
            COLOR 0, 7
            line.no% = line.no% + 1
            IF line.no% > 6 THEN line.no% = 1
            COLOR 0, 7
            GOSUB subselect
            COLOR 7, 0
         END IF
      END IF
      first.time$ = "N"
LOOP UNTIL Exit.Code%=0 OR Exit.Code%=-73 OR Exit.Code%=2
                                        OR Exit.Code%=-81
     SELECT CASE line.no%
      CASE IS = 1
         ArMethCode$ = "TS-500": Kind = 0: KindT$ = "N"
      CASE IS = 2
         ArMethCode$ = "ACI": Kind = 0: KindT$ = "N"
      CASE IS = 3
         Kind = 1
      CASE IS = 4
         Kind = 2
      CASE IS = 5
         Kind = 3
      CASE IS = 6
         Kind = 4
      CASE ELSE
      ' not assigned
   END SELECT
EXIT SUB

sub1:
   LOCATE 5, 5: PRINT "A) TS-500 code & Area Method                 "
RETURN

sub2:
   LOCATE 6, 5: PRINT "B) ACI Code & Area Method.                   "
RETURN

sub3:
   LOCATE 8, 5: PRINT "1) Rectangular Stress Block &                "
   LOCATE 9, 5: PRINT "   Integration of Area Method.               "
RETURN
```

```
sub4:
   LOCATE 11, 5: PRINT "2) CEB Stress Block &                              "
   LOCATE 12, 5: PRINT "   Integration of Area Method.                     "
RETURN

sub5:
   LOCATE 14, 5: PRINT "3) Kent & Park Stress-Strain Relationship"
   LOCATE 15, 5: PRINT "   & Integration of Area Method.         "
RETURN

sub6:
   LOCATE 17, 5: PRINT "4) Hognestad Stress-Strain Relationship &"
   LOCATE 18, 5: PRINT "   Integration of Area Method.           "
RETURN

subselect:
   SELECT CASE line.no%
      CASE IS = 1
         GOSUB sub1
         Kind$ = "A"
      CASE IS = 2
         GOSUB sub2
         Kind$ = "B"
      CASE IS = 3
         GOSUB sub3
         Kind$ = "1"
      CASE IS = 4
         GOSUB sub4
         Kind$ = "2"
      CASE IS = 5
         GOSUB sub5
         Kind$ = "3"
      CASE IS = 6
          GOSUB sub6
         Kind$ = "4"
      CASE ELSE
      ' not assigned
   END SELECT
RETURN

END SUB

SUB Min.Ecc.Control
      Table.Layout
      LOCATE 5, 5: PRINT "what is the minimum eccentricity ?"
      Descriptions 2
      tbg% = 0
      tfg% = 7
      InpEccMin# = EccMin
      DO
         LOCATE 8, 8: PRINT "Emin = ";
         Numin InpEccMin#, 2, 4, Exit.Code%, tbg%, tfg%
         IF EccMin < 0 THEN
            LOCATE 8, 15: PRINT SPACE$(10)
         END IF
```

```
      LOOP UNTIL EccMin >= 0
      EccMin = InpEccMin#
END SUB

SUB Norm.Value
      Table.Layout
      Descriptions 12
      tbg% = 0
      tfg% = 7
      LOCATE 5, 5: PRINT "Input the norm value to terminate"
      LOCATE 6, 5: PRINT "iteration :"
      IF Epsilon = 0 THEN Epsilon = .001
      InpEpsilon# = Epsilon
      LOCATE 9, 5: PRINT "E = ";
      Numin InpEpsilon#, 1, 6, Exit.Code%, tbg%, tfg%
      Epsilon = InpEpsilon#
END SUB

SUB Num.Der.Val
      tbg% = 0
      tfg% = 7
      Table.Layout
      Descriptions 11
      IF dYbar = 0 THEN dYbar = .1
      IF dAlfa = 0 THEN dAlfa = .1
      IF dAst = 0 THEN dAst = .1
      LOCATE 5, 5: PRINT "Input the delta values for numerical "
      LOCATE 6, 5: PRINT "derivatives :"
      LOCATE 7, 5: PRINT "    _"
      LOCATE 8, 5: PRINT "d Y  = ";
      PRINT USING "##.##"; dYbar
      LOCATE 10, 5: PRINT "d ane= ";
      PRINT USING "##.##"; dAlfa
      LOCATE 12, 5: PRINT "d Ast= ";
      PRINT USING "##.##"; dAst

Get.InpdYbar:
      InpdYbar# = dYbar
      LOCATE 8, 12
      Numin InpdYbar#, 2, 2, Exit.Code%, tbg%, tfg%
      dYbar = InpdYbar#
      IF Exit.Code% = -72 THEN GOTO Get.dAst:
      IF Exit.Code% = -80 OR Exit.Code% = 0 THEN GOTO Get.dAlfa
      GOTO End.Num.Der.Val

Get.dAlfa:
      InpdAlfa# = dAlfa
      LOCATE 10, 12
      Numin InpdAlfa#, 2, 2, Exit.Code%, tbg%, tfg%
      dAlfa = InpdAlfa#
      IF Exit.Code% = -72 THEN GOTO Get.InpdYbar
      IF Exit.Code% = -80 OR Exit.Code% = 0 THEN GOTO Get.dAst
      GOTO End.Num.Der.Val
```

```
Get.dAst:
      InpdAst# = dAst
      LOCATE 12, 12
      Numin InpdAst#, 2, 2, Exit.Code%, tbg%, tfg%
      dAst = InpdAst#
      IF Exit.Code% = -72 THEN GOTO Get.dAlfa:
      IF Exit.Code% = -80 OR Exit.Code% = 0 THEN GOTO Get.InpdYbar

End.Num.Der.Val:

END SUB

SUB Peak.Stress.Inp
      Table.Layout
      Descriptions 6
      LOCATE 5, 5: PRINT "What is the value for k3 ? "
      LOCATE 8, 5: PRINT "   rmax =  k3  x fcd"
      LOCATE 10, 5: PRINT "   k3 <Press Enter for 0.85>=";
      IF Peak = 0 THEN Peak = .85
      InpPeak# = Peak
      tbg% = 0
      tfg% = 7
      Numin InpPeak#, 1, 5, Exit.Code%, tbg%, tfg%
      Peak = InpPeak#
END SUB

SUB ReStart.Iter.Val
   LOCATE 25, 1: PRINT "| F1:select bar| F2:Save Output| F3:Solve
                       | F4:Restart| F5:End| F6:Solver Table |";
   GOTO First.Box

Section.box:
   GOSUB Box.section
   IF Exit.Code% = -73 THEN GOTO Main.Sub.End
   IF Exit.Code% = -81 THEN GOTO First.Box

File.box:
   GOSUB Box.File
   IF Exit.Code% = -81 THEN GOTO First.Box
   IF Exit.Code% = -73 THEN GOTO Main.Sub.End
   GOTO Section.box

First.Box:
   LOCATE 3, 7: PRINT USING "#######.##"; Nd
   LOCATE 4, 7: PRINT USING "#######.##"; Mxd
   LOCATE 5, 7: PRINT USING "#######.##"; Myd

   IF NOCSN = 0 THEN
      LOCATE 6, 4: PRINT "Md:"
      LOCATE 6, 7
  PRINT USING "#######.##"; (MxdOrg * MxdOrg + MydOrg * MydOrg) ^ .5
   ELSE
      LOCATE 6, 4: PRINT "                "
   END IF
```

```
   LOCATE 7, 8: PRINT USING "####.#####"; Ybar
   LOCATE 8, 8: PRINT USING "####.#####"; AlfaNE
   LOCATE 9, 8: PRINT USING "####.#####"; Ast
   GOSUB Box.load.iter0
   IF Exit.Code% = -73 THEN GOTO Section.box
   IF Exit.Code% = -81 OR Exit.Code% = 0 THEN GOTO Min.Eccen.Box
   GOTO Main.Sub.End

Min.Eccen.Box:
   GOSUB box.Min.Eccen
   IF Exit.Code% = -73 THEN GOTO First.Box
   IF Exit.Code% = -81 OR Exit.Code% = 0 THEN GOTO Jacob.box
   GOTO Main.Sub.End

Jacob.box:
   GOSUB box.Jacob
   IF Exit.Code% = -73 THEN GOTO Min.Eccen.Box
   IF Exit.Code% = -81 OR Exit.Code% = 0 THEN GOTO Kind.Solv.box
   GOTO Main.Sub.End

Kind.Solv.box:
   GOSUB box.Kind.Solv
   IF Exit.Code% = -73 THEN GOTO Jacob.box
   IF Exit.Code% = -81 OR Exit.Code% = 0 THEN GOTO Material.box
   GOTO Main.Sub.End

Material.box:
   GOSUB box.Material
   IF Exit.Code% = -73 THEN GOTO Kind.Solv.box
   IF Exit.Code% = -81 OR Exit.Code% = 0 THEN GOTO SolvPar.box
   GOTO Main.Sub.End

SolvPar.box:
   GOSUB box.SolvPar
   IF Exit.Code% = -73 THEN GOTO Material.box
   IF Exit.Code% = -81 OR Exit.Code% = 0 THEN GOTO First.Box
   GOTO Main.Sub.End

Main.Sub.End:
   LOCATE 25, 1: PRINT SPACE$(80);
   IF UnBi$ = "U" THEN AlfaNE = 0
   IF Exit.Code% = -61 AND Kind <> 0 AND dint = 0 THEN
   LOCATE 25, 1: PRINT "│ F1:select bar│ F2:Save Output│ F3:Solve
                       │ F4:Restart│ F5:End│ F6:Solver Table │";
   GOTO SolvPar.box
   END IF
   FOR i = 3 TO 9
       LOCATE i, 20: PRINT SPACE$(13);
   NEXT i
   IF Kind = 3 AND Exit.Code% = -61 THEN
      col% = 43
      FOR ROW% = 3 TO 9
         LOCATE ROW%, col%: FOR i = 1 TO 37: PRINT "▒"; : NEXT i
      NEXT ROW%
      LOCATE 6, 51: PRINT ">▒Ecu =";
```

```
      InpEpsCu# = EpsCu
      Numin InpEpsCu#, 1, 6, Exit.Code%, -1, -1
      EpsCu = InpEpsCu#
      LOCATE 6, 51: PRINT "▒";
      LOCATE 7, 51: PRINT ">▒E50h=";
      InpEps50h# = Eps50h
      Numin InpEps50h#, 1, 6, Exit.Code%, -1, -1
      Eps50h = InpEps50h#
      LOCATE 7, 51: PRINT "▒";
   END IF
EXIT SUB

'********************** Subroutines ***********************

Box.load.iter0:
   STAST# = Ast
   STALFA# = AlfaNE
   STYBAR# = Ybar
   InpNdOrg# = NdOrg
   InpMxdOrg# = MxdOrg
   InpMydOrg# = MydOrg
   LOCATE 2, 1: PRINT "/-"
   FOR i = 3 TO 9
   LOCATE i, 1: PRINT "|"
   LOCATE i, 19: PRINT "|"
   NEXT i
   LOCATE 10, 1: PRINT "\-----------------/"
   LOCATE 2, 10: PRINT "---------\"

line.Get.Nd:
   LOCATE 3, 2: PRINT ">"
   LOCATE 3, 7
   Numin InpNdOrg#, 8, 2, Exit.Code%, -1, -1
   LOCATE 3, 2: PRINT " "
   NdOrg = InpNdOrg#
   IF Exit.Code% = -72 THEN GOTO get.Ast0
   IF Exit.Code% = -80 OR Exit.Code% = 0 THEN GOTO line.Get.Mxd
   GOTO l.End.Get.Load

line.Get.Mxd:
   LOCATE 4, 2: PRINT ">"
   LOCATE 4, 7
   Numin InpMxdOrg#, 8, 2, Exit.Code%, -1, -1
   LOCATE 4, 2: PRINT " "
   MxdOrg = InpMxdOrg#
   IF NOCSN = 0 THEN
      LOCATE 6, 4: PRINT "Md:"
      LOCATE 6, 7
 PRINT USING "#######.##"; (MxdOrg * MxdOrg + MydOrg * MydOrg) ^ .5
   END IF
   IF Exit.Code% = -72 THEN GOTO line.Get.Nd
   IF Exit.Code% = -80 OR Exit.Code% = 0 THEN GOTO line.Get.Myd
   GOTO l.End.Get.Load
```

```
line.Get.Myd:
   LOCATE 5, 2: PRINT ">"
   LOCATE 5, 7
   Numin InpMydOrg#, 8, 2, Exit.Code%, -1, -1
   LOCATE 5, 2: PRINT " "
   MydOrg = InpMydOrg#
   IF NOCSN = 0 THEN
      LOCATE 6, 4: PRINT "Md:"
      LOCATE 6, 7
 PRINT USING "########.##"; (MxdOrg * MxdOrg + MydOrg * MydOrg) ^ .5
   END IF
   IF Exit.Code% = -72 THEN GOTO line.Get.Mxd
   IF Exit.Code% = -80 OR Exit.Code% = 0 THEN GOTO get.Ybar0
   GOTO l.End.Get.Load

get.Ybar0:
   LOCATE 7, 2: PRINT ">"
   LOCATE 7, 8
   Numin STYBAR#, 4, 5, Exit.Code%, -1, -1
   LOCATE 7, 2: PRINT " "
   IF Exit.Code% = -72 OR Exit.Code% = 1 THEN GOTO line.Get.Myd
   IF Exit.Code% = -80 OR Exit.Code% = 0 THEN GOTO get.AlfaNe0
   GOTO End.Get.Iter.start

get.AlfaNe0:
   IF UnBi$ = "B" THEN
   LOCATE 8, 2: PRINT ">"
   LOCATE 8, 8
   Numin STALFA#, 4, 5, Exit.Code%, -1, -1
   LOCATE 8, 2: PRINT " "
   IF Exit.Code% = -72 OR Exit.Code% = 1 THEN GOTO get.Ybar0
   IF Exit.Code% = -80 OR Exit.Code% = 0 THEN GOTO get.Ast0
   GOTO End.Get.Iter.start
   END IF

get.Ast0:
   LOCATE 9, 2: PRINT ">"
   LOCATE 9, 8
   Numin STAST#, 4, 5, Exit.Code%, -1, -1
   LOCATE 9, 2: PRINT " "
   IF Exit.Code% = -72 OR Exit.Code% = 1 THEN
      IF UnBi$ = "B" THEN GOTO get.AlfaNe0 ELSE GOTO get.Ybar0
   END IF
   IF Exit.Code% = -80 OR Exit.Code% = 0 THEN GOTO line.Get.Nd

End.Get.Iter.start:

l.End.Get.Load:
   Nd = NdOrg
   Mxd = MxdOrg
   Myd = MydOrg
   Md = (MxdOrg * MxdOrg + MydOrg * MydOrg) ^ .5
   IF STYBAR# <> Ybar THEN Ybar = STYBAR#
   IF STALFA# <> AlfaNE THEN AlfaNE = STALFA#
   IF STAST# <> Ast THEN Ast = STAST#
```

```
    LOCATE 2, 1: PRINT "/-"
    FOR i = 3 TO 9
    LOCATE i, 1: PRINT "|"
    LOCATE i, 19: PRINT "|"
    NEXT i
    LOCATE 10, 1: PRINT "\-----------------"
    LOCATE 2, 10: PRINT "---------"

RETURN

Box.section:
LOCATE 1, 53: PRINT USING "\        \"; Section.Fname$
LOCATE 1, 42: PRINT "| >"
LOCATE 1, 80: PRINT "|";
LOCATE 2, 42: PRINT "\-------------------------------------/"

Text.In.Head:
      LOCATE 1, 53
      Textin Section.Fname$, 8, 1, Exit.Code%, -1, -1
      IF Section.Fname$ = "" THEN BEEP: GOTO Text.In.Head
      ltext = LEN(Section.Fname$)
      textkind$ = "Text"
      IF ASC(MID$(Section.Fname$, 1, 1)) < 65 THEN
         FOR i = 1 TO ltext
 IF ASC(MID$(Section.Fname$, i, 1)) < 65 THEN textkind$ = "Number"
         NEXT i
         IF textkind$ = "Number" THEN
            secnum = VAL(Section.Fname$)
            IF secnum > -1 AND secnum < 1000 THEN
               SELECT CASE ltext
                  CASE IS = 1
                     Sec$ = "SECTION.00" + Section.Fname$
                  CASE IS = 2
                     Sec$ = "SECTION.0" + Section.Fname$
                  CASE IS = 3
                     Sec$ = "SECTION." + Section.Fname$
                  CASE ELSE
                     GOTO Text.In.Head
                  END SELECT
               GOTO start.Read.head
            ELSE
               GOTO Text.In.Head
            END IF
         END IF
      ELSE
         Sec$ = Section.Fname$ + ".SEC"
      END IF

start.Read.head:
   Funit% = FREEFILE
   OPEN "I", #Funit%, Sec$
   INPUT #Funit%, NOCSN, NOSBAR
   ERASE XYC, XYCN, XYS
   DIM XYC(NOCSN + 1, 3), XYCN(NOCSN * 2, 2), XYS(NOSBAR, 3)
   IF NOCSN = 0 THEN
```

```
      INPUT #Funit%, DE, DI, Rbar
      FOR i = 1 TO NOSBAR
         INPUT #Funit%, XYS(i, 3)
      NEXT i
   ELSE
      FOR i = 1 TO NOCSN
         INPUT #Funit%, XYC(i, 1), XYC(i, 2)
      NEXT i
      XYC(NOCSN + 1, 1) = XYC(1, 1)
      XYC(NOCSN + 1, 2) = XYC(1, 2)
      FOR i = 1 TO NOSBAR
         INPUT #Funit%, XYS(i, 1), XYS(i, 2), XYS(i, 3)
      NEXT i
   END IF
   CLOSE #Funit%
   SUMRAT = 0
   FOR i = 1 TO NOSBAR
      SUMRAT = SUMRAT + XYS(i, 3)
   NEXT i
   FOR i = 1 TO NOSBAR
      XYS(i, 3) = XYS(i, 3) / SUMRAT
   NEXT i
LOCATE 2, 41: PRINT "\/- Stress-Strain ------- Ecu=        -\"
LOCATE 1, 42: PRINT "   "
LOCATE 1, 80: PRINT " ";
   IF NOCSN = 0 THEN
   UnBi$ = "U"
   LOCATE 16, 32: PRINT "Uniaxial"
   END IF
   IF NOCSN <> 0 THEN Arbit.Sec.Proper
   Arbit.Sec.Window
   WINDOW (Winx1, Winy1)-(Winx2, Winy2)
   Graph.Draw
RETURN

Box.File:
LOCATE 1, 71: PRINT USING "\        \"; DataFile$
LOCATE 1, 42: PRINT "|"
LOCATE 1, 63: PRINT ">"
LOCATE 1, 80: PRINT "|";
LOCATE 2, 42: PRINT "\-------------------------------------/"

t.inp:
      LOCATE 1, 71
      Textin DataFile$, 8, 1, Exit.Code%, -1, -1
      Strip.FName DataFile$, Extension$
      IF DataFile$ = "" THEN BEEP: GOTO t.inp
      MatFile$ = DataFile$ + ".MAT"

      Funit% = FREEFILE
      OPEN "I", #Funit%, MatFile$
      INPUT #Funit%, K1, Ec, Es, FcdOrg, Fck, Fctd, FydOrg, Fyk
      CLOSE #Funit%
   Fcd = FcdOrg
   Fyd = FydOrg
```

```
   LOCATE 14, 8: PRINT USING "######.###"; Fyd
   LOCATE 15, 8: PRINT USING "######.###"; Fyk
   LOCATE 16, 8: PRINT USING "##########"; Es
   LOCATE 18, 8: PRINT USING "######.###"; Fcd
   LOCATE 19, 8: PRINT USING "######.###"; Fck
   LOCATE 20, 8: PRINT USING "##########"; Ec
   LOCATE 21, 11: PRINT USING "#.#####"; K1
   LOCATE 22, 11: PRINT USING "#.#####"; Peak
   LOCATE 23, 9: PRINT USING "####.##"; Fctd
   LOCATE 2, 41: PRINT "\/- Stress-Strain ------- Ecu=        -\"
   LOCATE 1, 42: PRINT " "
   LOCATE 1, 63: PRINT " "
   LOCATE 1, 80: PRINT " ";
RETURN

box.Min.Eccen:
   LOCATE 2, 33: PRINT "/-------\"
   LOCATE 3, 33: PRINT "|Min.Ec.|"
   LOCATE 4, 33: PRINT "|       |"
   LOCATE 5, 33: PRINT "\-------/"
      InpEccMin# = EccMin

      DO
         LOCATE 4, 34
         Numin InpEccMin#, 2, 4, Exit.Code%, -1, -1
      LOOP UNTIL EccMin >= 0

      EccMin = InpEccMin#
   LOCATE 2, 33: PRINT "--------\"
   LOCATE 3, 33: PRINT "|"
   LOCATE 3, 41: PRINT "|"
   LOCATE 4, 33: PRINT "|"
   LOCATE 4, 41: PRINT "|"
   LOCATE 5, 33: PRINT "|-------|"
   IF EccMin > 0 AND Nd > 0 THEN
      Control.Min.Ecc
      LOCATE 25, 1
      PRINT "| F1:select bar| F2:Save Output| F3:Solve| F4:Restart
            | F5:End| F6:Solver Table |";
   END IF

RETURN

box.Jacob:
   LOCATE 5, 33: PRINT "/-------\"
   LOCATE 6, 33: PRINT "|Secant:|"
   LOCATE 7, 33: PRINT "|"
   LOCATE 7, 41: PRINT "|"
   LOCATE 8, 33: PRINT "|"
   LOCATE 8, 41: PRINT "|"
   LOCATE 9, 33: PRINT "|"
   LOCATE 9, 41: PRINT "|"
   LOCATE 10, 33: PRINT "\-------/"
```

```
Get.dYbar:
      InpdYbar# = dYbar
      LOCATE 7, 34: PRINT ">"
      LOCATE 7, 35
      Numin InpdYbar#, 2, 2, Exit.Code%, -1, -1
      LOCATE 7, 34: PRINT " "
      dYbar = InpdYbar#
      IF Exit.Code% = -72 THEN GOTO Get.dAs:
      IF Exit.Code% = -80 OR Exit.Code% = 0 THEN GOTO Get.dAl
      GOTO End.N.D.V

Get.dAl:
      InpdAlfa# = dAlfa
      LOCATE 8, 34: PRINT ">"
      LOCATE 8, 35
      Numin InpdAlfa#, 2, 2, Exit.Code%, -1, -1
      LOCATE 8, 34: PRINT " "
      dAlfa = InpdAlfa#
      IF Exit.Code% = -72 THEN GOTO Get.dYbar
      IF Exit.Code% = -80 OR Exit.Code% = 0 THEN GOTO Get.dAs
      GOTO End.N.D.V

Get.dAs:
      InpdAst# = dAst
      LOCATE 9, 34: PRINT ">"
      LOCATE 9, 35
      Numin InpdAst#, 2, 2, Exit.Code%, -1, -1
      LOCATE 9, 34: PRINT " "
      dAst = InpdAst#
      IF Exit.Code% = -72 THEN GOTO Get.dAl:
      IF Exit.Code% = -80 OR Exit.Code% = 0 THEN GOTO Get.dYbar

End.N.D.V:
   LOCATE 5, 33: PRINT "|-------|"
   LOCATE 6, 33: PRINT "|Secant:|"
   LOCATE 7, 33: PRINT "|"
   LOCATE 7, 41: PRINT "|"
   LOCATE 8, 33: PRINT "|"
   LOCATE 8, 41: PRINT "|"
   LOCATE 9, 33: PRINT "|"
   LOCATE 9, 41: PRINT "|"
   LOCATE 10, 33: PRINT "--------/"

RETURN

box.Kind.Solv:
   LOCATE 11, 1: PRINT "/- Method :  ---------------------------\"
   LOCATE 12, 1: PRINT "|                                        |"
   LOCATE 13, 1: PRINT "\----------------------------------------/"
   IF ArMethCode$ = "TS-500" AND Kind = 0 THEN line.no% = 1
   IF ArMethCode$ = "ACI" AND Kind = 0 THEN line.no% = 2
   IF Kind = 1 THEN line.no% = 3
   IF Kind = 2 THEN line.no% = 4
   IF Kind = 3 THEN line.no% = 5
   IF Kind = 4 THEN line.no% = 6
```

```
   IF line.no% = 0 THEN line.no% = 1
   GOSUB prn.kind

T.in.H:
   LOCATE 1, 37
   Textin T$, 1, 37, Exit.Code%, -1, -1
   IF Exit.Code% = -73 OR Exit.Code% = -81 THEN GOTO sel.end
   IF Exit.Code% < -58 AND Exit.Code% > -65 THEN GOTO sel.end
   IF Exit.Code% = -72 THEN
      line.no% = line.no% - 1
   ELSE
      line.no% = line.no% + 1
   END IF
   IF line.no% < 1 THEN line.no% = 6
   IF line.no% > 6 THEN line.no% = 1
   GOSUB prn.kind
   GOTO T.in.H

sel.end:
   LOCATE 11, 1: PRINT "/- Method: ------------------------------\"
   LOCATE 12, 1: PRINT "|                                        |"
   LOCATE 13, 1: PRINT "|- Reinforcement --- Solver ------------|"
   GOSUB prn.kind
RETURN

prn.kind:
   LOCATE 12, 3
   SELECT CASE line.no%
      CASE IS = 1
         ArMethCode$ = "TS-500": Kind = 0: KindT$ = "N"
         PRINT "Area Method & TS-500 Code.            "
         EpsCu = .003
         Rec.Stress.Draw
      CASE IS = 2
         ArMethCode$ = "ACI": Kind = 0: KindT$ = "N"
         PRINT "Area Method & ACI Code.               "
         EpsCu = .003
         Rec.Stress.Draw
      CASE IS = 3
         Kind = 1
         PRINT "Int. of Area & Rec. Stress-Block.     "
         EpsCu = .003
         Rec.Stress.Draw
      CASE IS = 4
         Kind = 2
         PRINT "Int. of Area & CEB Stress-Block.      "
         EpsCu = .0035
         Stress.Draw
      CASE IS = 5
         Kind = 3
         PRINT "Int.of Area & Kent&Park Stress-Block. "
         EpsCu = .003
         Z = 325
         Stress.Draw
      CASE IS = 6
```

```
         Kind = 4
         PRINT "Int. of Area & Hognestad Stress-Block."
         EpsCu = .0038
         eps0 = .002
         Stress.Draw
      CASE ELSE
      ' not assigned
   END SELECT

RETURN

box.Material:

   LOCATE 13, 1: PRINT "/- Reinforcement -\"
   FOR LL% = 14 TO 23
      LOCATE LL%, 1: PRINT "|"
      LOCATE LL%, 19: PRINT "|"
   NEXT LL%
   LOCATE 17, 1: PRINT "|- Concrete  -----+"
   LOCATE 24, 1: PRINT "\-----------------/";

in.Fyd:
   Numin.Fyd# = FydOrg
   LOCATE 14, 2: PRINT ">"
   LOCATE 14, 8
   Numin Numin.Fyd#, 6, 3, Exit.Code%, -1, -1
   LOCATE 14, 2: PRINT " "
   FydOrg = Numin.Fyd#
   IF Exit.Code% = -72 THEN GOTO in.KindT
   IF Exit.Code% = -73 OR Exit.Code% = -81 THEN GOTO end.in.mat
   IF Exit.Code% < -58 AND Exit.Code% > -65 THEN GOTO end.in.mat
   Fyd = FydOrg

in.Fyk:
   Numin.Fyk# = Fyk
   LOCATE 15, 2: PRINT ">"
   LOCATE 15, 8
   Numin Numin.Fyk#, 6, 3, Exit.Code%, -1, -1
   LOCATE 15, 2: PRINT " "
   Fyk = Numin.Fyk#
   IF Exit.Code% = -72 THEN GOTO in.Fyd
   IF Exit.Code% = -73 OR Exit.Code% = -81 THEN GOTO end.in.mat
   IF Exit.Code% < -58 AND Exit.Code% > -65 THEN GOTO end.in.mat

in.Es:
   Numin.Es# = Es
   LOCATE 16, 2: PRINT ">"
   LOCATE 16, 8
   Numin Numin.Es#, 10, 0, Exit.Code%, -1, -1
   LOCATE 16, 2: PRINT " "
   Es = Numin.Es#
   IF Exit.Code% = -72 THEN GOTO in.Fyk
   IF Exit.Code% = -73 OR Exit.Code% = -81 THEN GOTO end.in.mat
   IF Exit.Code% < -58 AND Exit.Code% > -65 THEN GOTO end.in.mat
```

```
in.Fcd:
   Numin.Fcd# = FcdOrg
   LOCATE 18, 2: PRINT ">"
   LOCATE 18, 8
   Numin Numin.Fcd#, 6, 3, Exit.Code%, -1, -1
   LOCATE 18, 2: PRINT " "
   FcdOrg = Numin.Fcd#
   IF Exit.Code% = -72 THEN GOTO in.Es
   IF Exit.Code% = -73 OR Exit.Code% = -81 THEN GOTO end.in.mat
   IF Exit.Code% < -58 AND Exit.Code% > -65 THEN GOTO end.in.mat
   Fcd = FcdOrg

in.Fck:
   Numin.Fck# = Fck
   LOCATE 19, 2: PRINT ">"
   LOCATE 19, 8
   Numin Numin.Fck#, 6, 3, Exit.Code%, -1, -1
   LOCATE 19, 2: PRINT " "
   Fck = Numin.Fck#
   IF Exit.Code% = -72 THEN GOTO in.Fcd
   IF Exit.Code% = -73 OR Exit.Code% = -81 THEN GOTO end.in.mat
   IF Exit.Code% < -58 AND Exit.Code% > -65 THEN GOTO end.in.mat

in.Ec:
   Numin.Ec# = Ec
   LOCATE 20, 2: PRINT ">"
   LOCATE 20, 8
   Numin Numin.Ec#, 10, 0, Exit.Code%, -1, -1
   LOCATE 20, 2: PRINT " "
   Ec = Numin.Ec#
   IF Exit.Code% = -72 THEN GOTO in.Fck
   IF Exit.Code% = -73 OR Exit.Code% = -81 THEN GOTO end.in.mat
   IF Exit.Code% < -58 AND Exit.Code% > -65 THEN GOTO end.in.mat

in.K1:
   Numin.K1# = K1
   LOCATE 21, 2: PRINT ">"
   LOCATE 21, 11
   Numin Numin.K1#, 1, 5, Exit.Code%, -1, -1
   LOCATE 21, 2: PRINT " "
   K1 = Numin.K1#
   IF Exit.Code% = -72 THEN GOTO in.Ec
   IF Exit.Code% = -73 OR Exit.Code% = -81 THEN GOTO end.in.mat
   IF Exit.Code% < -58 AND Exit.Code% > -65 THEN GOTO end.in.mat

in.Peak:
   Numin.Peak# = Peak
   LOCATE 22, 2: PRINT ">"
   LOCATE 22, 11
   Numin Numin.Peak#, 1, 5, Exit.Code%, -1, -1
   LOCATE 22, 2: PRINT " "
   IF Exit.Code% = -72 THEN GOTO in.K1
   IF Exit.Code% = -73 OR Exit.Code% = -81 THEN GOTO end.in.mat
   IF Exit.Code% < -58 AND Exit.Code% > -65 THEN GOTO end.in.mat
```

```
in.Fctd:
   Numin.Fctd# = Fctd
   LOCATE 23, 2: PRINT ">"
   LOCATE 23, 9
   Numin Numin.Fctd#, 4, 2, Exit.Code%, -1, -1
   LOCATE 23, 2: PRINT " "
   IF Exit.Code% = -72 THEN GOTO in.Peak
   IF Exit.Code% = -73 OR Exit.Code% = -81 THEN GOTO end.in.mat
   IF Exit.Code% < -58 AND Exit.Code% > -65 THEN GOTO end.in.mat

in.KindT:
   LOCATE 23, 19: PRINT "<";
   LOCATE 24, 17: PRINT "^";
   LOCATE 1, 37
   Text.in.Fctd$ = KindT$
   Textin T$, 1, 1, Exit.Code%, -1, -1
   LOCATE 23, 19: PRINT "|";
   LOCATE 24, 17: PRINT "-";
   IF Exit.Code% = -72 THEN GOTO in.Fctd
   IF Exit.Code% = -73 OR Exit.Code% = -81 THEN GOTO end.in.mat
   IF Exit.Code% < -58 AND Exit.Code% > -65 THEN GOTO end.in.mat
   IF Exit.Code% = -80 THEN GOTO in.Fyd
   IF KindT$ = "Y" THEN KindT$ = "N" ELSE KindT$ = "Y"
   LOCATE 23, 17: PRINT KindT$;
   GOTO in.KindT

end.in.mat:
   LOCATE 13, 1: PRINT "|- Reinforcement -┤"
   FOR LL% = 14 TO 23
      LOCATE LL%, 1: PRINT "|"
      LOCATE LL%, 19: PRINT "|"
   NEXT LL%
   LOCATE 17, 1: PRINT "|- Concrete ------┤"
   LOCATE 24, 1: PRINT "\-----------------";
RETURN
'***********************

box.SolvPar:
   LOCATE 13, 19: PRINT "/- Solver -----------\"
   FOR LL% = 14 TO 16
   LOCATE LL%, 19: PRINT "|";
   LOCATE LL%, 41: PRINT "|";
   NEXT LL%
   LOCATE 17, 19: PRINT "\--------------------/"

In.Norm:
   LOCATE 14, 20: PRINT ">"
   LOCATE 14, 32
   InpEpsilon# = Epsilon
   Numin InpEpsilon#, 1, 6, Exit.Code%, -1, -1
   LOCATE 14, 20: PRINT " "
   Epsilon = InpEpsilon#
   IF Exit.Code% = -72 THEN GOTO in.Un.Bi
   IF Exit.Code% = -73 OR Exit.Code% = -81 THEN GOTO end.in.solv
   IF Exit.Code% < -58 AND Exit.Code% > -65 THEN GOTO end.in.solv
```

```
In.dint:
   LOCATE 15, 20: PRINT ">"
   LOCATE 15, 33
   Inpdint# = dint
   Numin Inpdint#, 3, 2, Exit.Code%, -1, -1
   IF Inpdint# < 0 THEN BEEP: GOTO In.dint
   LOCATE 15, 20: PRINT " "
   dint = Inpdint#
   IF Exit.Code% = -72 THEN GOTO In.Norm
   IF Exit.Code% = -73 OR Exit.Code% = -81 THEN GOTO end.in.solv
   IF Exit.Code% < -58 AND Exit.Code% > -65 THEN GOTO end.in.solv

in.Un.Bi:
   IF NOCSN = 0 THEN
      UnBi$ = "U"
      LOCATE 16, 32: PRINT "Uniaxial"
      GOTO In.dint
   ELSE
      LOCATE 16, 20: PRINT ">"
      LOCATE 1, 37
      Textin T$, 1, 1, Exit.Code%, -1, -1
      LOCATE 16, 20: PRINT " "
      IF Exit.Code% = -72 THEN GOTO In.dint
      IF Exit.Code% = -73 OR Exit.Code% = -81 THEN GOTO end.in.solv
      IF Exit.Code% < -58 AND Exit.Code% > -65 THEN GOTO end.in.solv
      IF Exit.Code% = -80 THEN GOTO In.Norm
      IF UnBi$ = "U" THEN UnBi$ = "B" ELSE UnBi$ = "U"
   END IF

   IF UnBi$ = "U" THEN
      LOCATE 16, 32: PRINT "Uniaxial"
   ELSE
      LOCATE 16, 32: PRINT " Biaxial"
   END IF
   GOTO in.Un.Bi

end.in.solv:
   LOCATE 13, 19: PRINT "-- Solver -----------|"
   FOR LL% = 14 TO 16
   LOCATE LL%, 19: PRINT "|";
   LOCATE LL%, 41: PRINT "|";
   NEXT LL%
   LOCATE 17, 19: PRINT "┼- Steel Bars --------|"
RETURN

END SUB

SUB Strip.Thickness
      Table.Layout
      LOCATE 5, 5: PRINT "Enter the integration thickness :"
      Descriptions 4
      tbg% = 0
      tfg% = 7
      IF dint = 0 THEN dint = 1
      Inpdint# = dint
```

```
get.dint:
      LOCATE 8, 10: PRINT "dint =";
      Numin Inpdint#, 2, 2, Exit.Code%, tbg%, tfg%
      IF Inpdint# < 0 THEN BEEP: GOTO get.dint
      dint = Inpdint#
END SUB

SUB Tensile.Strength
   Table.Layout
   LOCATE 5, 3: PRINT "Do you want to consider tensile strength of"
   LOCATE 6, 3: PRINT "concrete ?"
   Descriptions 5
   IF KindT$ = "" THEN KindT$ = "N"
   IF KindT$ = "Y" THEN
      COLOR 0, 7
      GOSUB subc1
      COLOR 7, 0
      GOSUB subc2
   ELSE
      COLOR 0, 7
      GOSUB subc2
      COLOR 7, 0
      GOSUB subc1
   END IF

   DO
   LOCATE 5, 2
   Textin T$, 1, Caps.On%, Exit.Code%, -1, -1
IF Exit.Code%=-73 OR Exit.Code%=-81 OR Exit.Code%=0 THEN GOTO l.op.end
   IF KindT$ = "Y" THEN KindT$ = "N" ELSE KindT$ = "Y"
   IF KindT$ = "Y" THEN
      COLOR 0, 7
      GOSUB subc1
      COLOR 7, 0
      GOSUB subc2
   ELSE
      COLOR 0, 7
      GOSUB subc2
      COLOR 7, 0
      GOSUB subc1
   END IF
l.op.end:

   LOOP UNTIL Exit.Code% = -73 OR Exit.Code% = -81 OR Exit.Code% = 0

EXIT SUB

subc1:
   LOCATE 9, 16: PRINT "Consider."
RETURN
subc2:
   LOCATE 11, 16: PRINT "Ignore.  "
RETURN

END SUB
```

```
SUB Uni.Bi.Bending
   IF UnBi$ = "" THEN UnBi$ = "B"
   Table.Layout
   Descriptions 13
   LOCATE 5, 5: PRINT "Solve as :"
   IF UnBi$ = "U" THEN
      COLOR 0, 7
      GOSUB subb1
      COLOR 7, 0
      GOSUB subb2
   ELSE
      COLOR 0, 7
      GOSUB subb2
      COLOR 7, 0
      GOSUB subb1
   END IF
   DO
   LOCATE 5, 4
   Textin T$, 1, Caps.On%, Exit.Code%, -1, -1
IF Exit.Code%=-73 OR Exit.Code%=-81 OR Exit.Code%=0 THEN GOTO loop.end
   IF UnBi$ = "U" THEN UnBi$ = "B" ELSE UnBi$ = "U"
   IF UnBi$ = "U" THEN
      COLOR 0, 7
      GOSUB subb1
      COLOR 7, 0
      GOSUB subb2
   ELSE
      COLOR 0, 7
      GOSUB subb2
      COLOR 7, 0
      GOSUB subb1
   END IF
loop.end:
   LOOP UNTIL Exit.Code% = -73 OR Exit.Code% = -81 OR Exit.Code% = 0

EXIT SUB

subb1:
   LOCATE 8, 9: PRINT "Uniaxial bending. (Ignore Myd)"
RETURN

subb2:
   LOCATE 10, 9: PRINT "Biaxial bending."
RETURN

END SUB
```

B.6. RCSDCOM

```
'       ****************************************************************
'       *       This include file describes the common variables       *
'       ****************************************************************

COMMON SHARED /Load..../ Nd, Myd, Mxd, Md, N, Mx, My, M
COMMON SHARED /Load..../ NdOrg, MydOrg, MxdOrg, EccMin
COMMON SHARED /Unknown./ Ybar, AlfaNE, Ast,EpsCu,Eps50h,Peak, eps0, Z
COMMON SHARED /Strip.../ YPint, Xp, dint, XL, YL, SumL, Strain, Stress
COMMON SHARED /Strip.../ HNEMAX, HDINT, MNE,Mi,MIJ, Hdind, Adim, YINTEND
COMMON SHARED /Material/ Ec,Es,K1,Fctd,Fcd,Fck,Fyd, Fyk, FcdOrg, FydOrg
COMMON SHARED /SecData./ DE, DI, Rbar, NOSBAR, NOCSN, NodNew
COMMON SHARED /SecDim../ XYC(), XYS(), XYCN()
COMMON SHARED /NRDim.../ x1(), x2(), F1(), d(), B(), a(), d
COMMON SHARED /SecGeo../ CSXmax, CSYmax, CSXmin, CSYmin
COMMON SHARED /SecGeo../ AreaGr, XgGros, YgGros, AreaCC, CapReduc
COMMON SHARED /SolPar../ Kind, Iter, Epsilon, Criter,dYbar,dAlfa,dAst
COMMON SHARED /Graphic./ Graphcard%, AspRatio
COMMON SHARED /Graphic./ Winx1!, Winx2!, Winy1!, Winy2!
COMMON SHARED /Graphic./ Referx1!, ReferY1!, ReferX2!, ReferY2!
COMMON SHARED /Constant/ PI#, AstOK
COMMON SHARED /Control./ Cont$,EccControl$,REFL$,KindT$,UnBi$,Rot$,N0max
COMMON SHARED /Control./ NdIgnor$, AST1MSG$, AstMax$, ArMethCode$
COMMON SHARED /FlNames./ Section.Fname$, DataFile$, Exit.Code%
```

APPENDIX: C

SAMPLE INPUT AND OUTPUT

A P P E N D I X C

SAMPLE INPUT AND OUTPUT

C.1. General.

In this Appendix Ex.10 is solved and related data and output are presented in a more detailed form. The data includes two files. The one called "DUEY18A.MAT" contains the material properties and the other file i.e. "SECTION.003" contains the sectional geometry. The other data are input directly in an interactive form by the computer program.

C.2. Matrerial Data File.

The map of the file "DUEY18A.MAT" is presented as following.

LINE	Column 1.........2.........3.........4.........
1.	0.82 318000 2E6 200 300 12.5 3650 4200

C.3. Sectional Geometry Data File.

The map of the file "SECTION.003" is presented as following.

LINE	Column 1.........2.........3.........4.........
	4 8
	0 0
	0 40
	40 40
	40 0
	4 4 .125
	20 4 .125
	36 4 .125
	36 20 .125
	36 36 .125
	20 36 .125
	4 36 .125
	4 20 .125

C.4. Main Program Demonstration.

RCSD

Ver. 5.0

Reinforced Concrete Section Design

FOR DESIGN OF REINFORCED CONCRETE SECTIONS
UNDER AXIAL LOAD AND BIAXIAL BENDING.

<STRIKE ANY KEY TO CONTINUE>

PROGRAMMED BY:
MARJANI F.
Dec. 1989 ,Ver. 5.0

AS PART OF AN MS THESIS SUPERVISED BY:

U. ERSOY

PROF. DR. OF CIVIL ENGINEERING
MIDDLE EAST TECHNICAL UNIVERSITY

<STRIKE ANY KEY TO CONTINUE

INPUT MENU

ENTER THE TYPE OF SOLUTION.

A) TS-500 code & Area Method
B) ACI Code & Area Method.

1) Rectangular Stress Block &
Integration of Area Method.

2) CEB Stress Block &
Integration of Area Method.

3) Kent & Park Stress-Strain Relationship
& Integration of Area Method.

4) Hognestad Stress-Strain Relationship &
Integration of Area Method.

Format

Select by arrow keys

& Press ‹Enter›

Description

Area Method :
Developed for rectangular stress block only.
Used by TS-500 & ACI codes.

Integration of Area Method:
Any stress-strain relation is Available . Concrete internal forces are calculated by dis-cretizing of section into small strips parallel to neutral axis.
Where strip thickness is specified by : [dint]

INPUT MENU

Enter the integration thickness :

dint = 1.00

Format

Print the value,

& Press Enter.

Description

dint is the thickness of integration strips.

INPUT MENU

Do you want to consider tensile strength of concrete ?

Consider.

Ignore.

Format

Select by arrow keys

& Press <Enter>

Description

Consider:
The tensile strength of concrete will be considered in calculation of concrete internal forces.

Ignor :
Will not.

INPUT MENU

What is the value for k3 ?

σmax = k3 x fcd

k3 ‹Press Enter for 0.85›= .85000

Format

Print the value.

& Press Enter.

Description

The peak value of stress will be k3 times the concrete compressive strength.

In most of codes k3 is takes as 0.85 (ACI,CEB & TS500)

INPUT MENU

what is the minimum eccentricity ?

Emin = 2.5000

Format

Print the value
and press ‹Enter›

Description

Print zero for No control
i.e. Minimum eccentricity will not be checked.
Used for the cases when axial load N ≤ 0

For the given eccentricity:
The Program will control whether the eccentricity is less than the minimum value or not.
(in both X and Y directions).
(TS-500 : e ≥ .1 d , e ≥ 2.5cm
If e ‹ emin.
The program uses emin.

INPUT MENU

Enter Material file name :

Material data file name =DUEY18A

Format

Print the value.

& Press Enter.

Description

Material File :
It is the file where the material properties of the R/C section are saved.
These are:
K1 Ec Es fcd fck fctd fyd fyk

File name has an extention of '.MAT' . Note that file name should be written without any extention.

INPUT MENU

Input section file :

Section: 3

Format

Print section no
Or
Print Section file name
& press Enter.

Description

Section file is the file where sectional geometry of R/C cross-section are saved.
If you give the section number, program will search for the file named 'section.fff' where the extention is the section number.For example SECTION.027

If you give the section file name, program will search for the file name given with an extension of '.SEC '

INPUT MENU

Input the design loads :

Nd = 200000.00

Mxd= 2100000.00

Myd= 1600000.00

Format

Print the value

& press Enter.

Description

Nd :
It is the design axial load.

Mxd :
It is the design moment about X-axis.

Myd :
It is the design moment about Y-axis.

INPUT MENU

Input the delta values for numerical derivatives :

d $\bar{Y}$ = .10

d ane= 0.10

d Ast= 0.10

Format

Print the value

& press Enter.

Description

To obtain the jacobian matrix the value of the delta to be used in calculation of derivatives (in secant method) is needed.

$$\frac{\delta f}{\delta X} = \frac{F(X+dX) - F(X)}{dX}$$

INPUT MENU

Input the norm value to terminate iteration :

ε = .001000

Format

Print the value

& press Enter.

Description

ε describe the resolution of the results.

$\epsilon = \sqrt{(\delta Ybar)^2 + (\delta\alpha)^2 + (\delta Ast)^2}$

INPUT MENU

Solve as :

Uniaxial bending. (Ignore Myd)

Biaxial bending.

Format

Select by arrow keys

& Press <Enter>

Description

In uniaxial bending Myd is ignored and neutral axis is assumed to be perpendicular to the plane of loading.

In biaxial bending there is not any restriction to the position of neutral axis and it may or may not be perpendicular to the plane of loading.

RCSD: Solver Table

Section:3 File: DUEY18A

Input

>Nd : 200000.00
Mxd: 2100000.00
Myd: 1600000.00

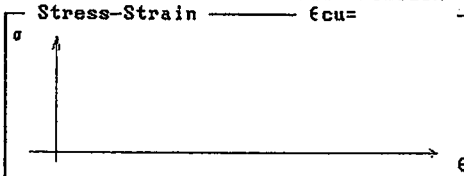
$\bar{Y}$: 35.23809
αne: 37.30395
Ast: 16.00000

Iter=

Min.Ec.
2.5000

Secant:
0.10
0.10
0.10

Stress-Strain εcu=



Method :
Int. of Area & Rec. Stress-Block

Reinforcement
fyd = 3650.000
fyk = 4200.000
Es = 2000000

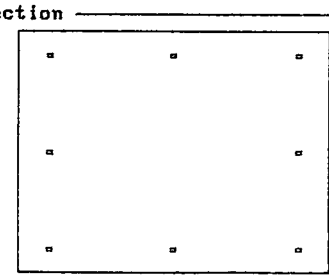
Solver
Norm : ε = 0.001000
Strip: dint= 1.00
Solve as : Biaxial

Concrete
fcd = 200.000
fck = 300.000
Ec = 318000
k1 = 0.82000
K3 = 0.85000
fctd= 12.50<N>

Steel Bars

No	As	ø	As +

Section



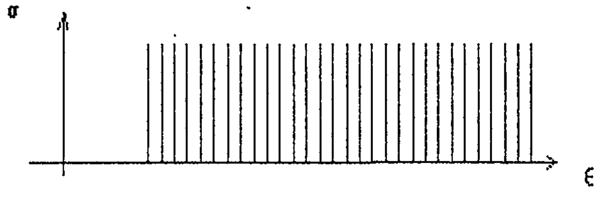
| F1:select bar | F2:Save Output | F3:Solve | F4:Restart | F5:End | F6:Solver Table |

RCSD: Solver Table

Section:3 File: DUEY18A

Input		Iter= 4	Min.Ec.
Nd :	200000.00	200000.11	2.5000
Mxd:	2100000.00	2100008.25	
Myd:	1600000.00	1600032.75	Secant:
$\bar{Y}$:	35.23809	46.12953	0.10
αne:	37.30395	39.53342	0.10
Ast:	16.00000	47.82656	0.10

Stress-Strain — €cu=0.00300



Method :
Int. of Area & Rec. Stress-Block

Reinforcement
fyd = 3650.000
fyk = 4200.000
Es = 2000000

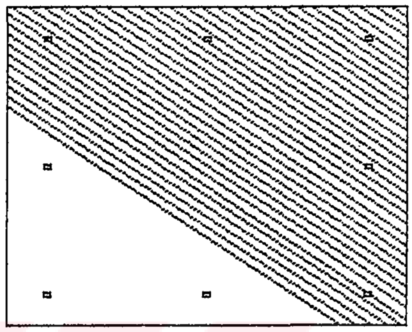
Solver
Norm : € = 0.001000
Strip: dint= 1.00
Solve as : Biaxial

Concrete
fcd = 200.000
fck = 300.000
Ec = 318000
k1 = 0.82000
K3 = 0.85000
fctd= 12.50<N>

Steel Bars

No	As	ø	As +
1	5.978	28	0.179
2	5.978	28	0.179
3	5.978	28	0.179
4	5.978	28	0.179
5	5.978	28	0.179

Section



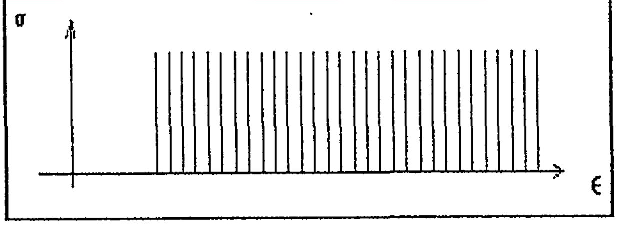
| F1:select bar| F2:Save Output| F3:Solve| F4:Restart| F5:End| F6:Solver Table |

RCSD: Solver Table

Section:3 File: DUEY18A

Input		Iter= 4	Min.Ec.
Nd :	200000.00	200000.11	2.5000
Mxd:	2100000.00	2100008.25	
Myd:	1600000.00	1600032.75	Secant:
$\bar{Y}$:	35.23809	46.12953	0.10
αne:	37.30395	39.53342	0.10
Ast:	16.00000	47.82656	0.10

Stress-Strain — €cu=0.00300



Method :
Int. of Area & Rec. Stress-Block

Reinforcement
fyd = 3650.000
fyk = 4200.000
Es = 2000000

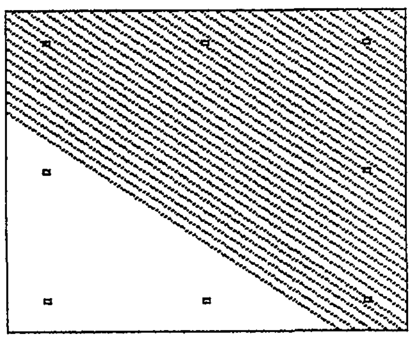
Solver
Norm : € = 0.001000
Strip: dint= 1.00
Solve as : Biaxial

Concrete
fcd = 200.000
fck = 300.000
Ec = 318000
k1 = 0.82000
K3 = 0.85000
fctd= 12.50<N>

Steel Bars

No	As	ø	As +
6	5.978	28	0.179
7	5.978	28	0.179
8	5.978	28	0.179

Section



| F1:select bar| F2:Save Output| F3:Solve| F4:Restart| F5:End| F6:Solver Table |

C.5. Output File.

```
MATERIAL DATA FILE NAME = DUEY18A
TYPE OF SOLUTION= RECTANGULAR STRESS BLOCK AND INTEGRATION OF AREA METHOD
dint=  1
Peak Stress :K3= .85  fc
K1= .82
ES= 2000000
Fcd= 200
Fyd= 3650
Nd= 200000
Mxd= 2100000
Myd= 1600000
SECTION FILE NAME = 3

# OF CONCRETE SECTION NODES= 4
# OF SECTION STEEL BARS 8
SECTION ROTATED 0 DEG.
CONCRETE SECTION COORDINATES.
------------------------------
      -X-      -Y-
       0        0
       0       40
      40       40
      40        0

COORDINATE OF STEEL BARS.
--------------------------
      -X-      -Y-               -RATIO-
       4        4               .125
      20        4               .125
      36        4               .125
      36       20               .125
      36       36               .125
      20       36               .125
       4       36               .125
       4       20               .125
** R E S U L T S **

ϵcu= .003

YBAR= 46.12953
ALFANE= 39.53342
AST= 47.82656
```